



Benutzerhandbuch

AWS Database Migration Service



AWS Database Migration Service: Benutzerhandbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS Database Migration Service?	1
Von AWS DMS durchgeführte Migrationsaufgaben	2
Wie AWS DMS funktioniert	4
Überblick über AWS DMS	4
Komponenten	6
Quellen	14
Quellen für die Datenmigration	14
Quellen für DMS Fleet Advisor	17
Quellen für DMS Schema Conversion	18
Quellen für homogene DMS-Datenmigrationen	18
Targets (Ziele)	19
Ziele für die Datenmigration	19
Ziele für DMS Fleet Advisor	22
Ziele für DMS Schema Conversion	22
Ziele für homogene DMS-Datenmigrationen	23
Amazon-Ressourcennamen	23
Mit anderen AWS Diensten	26
Support für AWS CloudFormation	26
Erste Schritte	28
Einrichtung	28
So melden Sie sich für ein AWS-Konto an	28
Einen Administratorbenutzer erstellen	29
Voraussetzungen	30
Erstellen einer VPC	31
Erstellen von Amazon-RDS-Parametergruppen	33
Erstellen Ihrer Amazon-RDS-Quelldatenbank	34
Erstellen Ihrer Amazon-RDS-Zieldatenbank	36
Erstellen eines Amazon-EC2-Clients	37
Füllen Ihrer Quelldatenbank	38
Schema migrieren	40
Replikation	42
Schritt 1: Erstellen einer Replikations-Instance	42
Schritt 2: Angeben von Quell- und Zielendpunkten	45
Schritt 3: Erstellen einer Aufgabe und Migrieren der Daten	46

Schritt 4: Testen der Replikation	48
Schritt 5: Bereinigen von AWS DMS-Ressourcen	50
Weitere Ressourcen	52
Ermitteln von Datenbanken für die Migration	53
Unterstützte AWS-Regionen	54
Erste Schritte	56
Einrichtung	57
Erstellen der erforderlichen Ressourcen	57
Erstellen von Datenbankbenutzern	67
Datenkollektoren	74
Berechtigungen	75
Erstellen eines Datenkollektors	76
Installieren eines Datenkollektors	78
Ermitteln von Betriebssystem- und Datenbankservern	82
Verwalten von überwachten Objekten	85
Verwenden von SSL	88
Erfassen von Daten	89
Fehlerbehebung	95
-Bestand	98
Verwenden eines Datenbankinventars	99
Verwenden des Schemainventars	100
Zielempfehlungen	102
Ziel-Instances	103
Wie bestimmt DMS Fleet Advisor Ziel-Spezifikationen?	103
Generieren von Zielempfehlungen	104
Einzelheiten der Empfehlung	106
Exportieren von Zielempfehlungen	108
Migrationsbeschränkungen	109
Fehlerbehebung	128
Einschränkungen	129
Konvertieren von Datenbankschemata	131
Unterstützt AWS-Regionen	132
Features	133
Einschränkungen	134
Erste Schritte	135
Voraussetzungen	136

Schritt 1: Erstellen eines Instance-Profils	142
Schritt 2: Konfigurieren von Datenanbietern	142
Schritt 3: Erstellen eines Migrationsprojekts	143
Schritt 4: Erstellen eines Bewertungsberichts	144
Schritt 5: Konvertieren Ihres Quellcodes	145
Schritt 6: Anwenden des konvertierten Codes	146
Schritt 7: Bereinigen und Fehlersuche	146
Einrichten eines Netzwerks	147
Konfiguration mit einer VPC	148
Konfiguration mit mehreren VPCs	148
Verwenden von AWS Direct Connect oder einer VPN	149
Verwenden einer Internetverbindung	149
Verwenden einer Umgebung ohne Internet-Gateway	150
Erstellen von Quelldatenanbietern	150
Verwenden von SQL Server als Quelle	151
Verwenden von Oracle als Quelle	153
Verwenden von Oracle Data Warehouse als Quelle	154
Verwenden von PostgreSQL als Quelle	157
Verwenden von MySQL als Quelle	157
Erstellen von Zieldatenanbietern	158
Verwenden von MySQL als Ziel	158
Verwenden von PostgreSQL als Ziel	160
Verwenden von Amazon Redshift als Ziel	161
Verwaltung von Migrationsprojekten	162
Angaben von Migrationsprojekt-Einstellungen	163
Bewertungsberichte zur Datenbankmigration	164
Erstellen eines Bewertungsberichts	165
Anzeigen Ihres Bewertungsberichts	165
Speichern von Bewertungsberichten	166
Schemakonvertierung	169
Einrichten von Transformationsregeln	170
Konvertieren Ihres Datenbankschemas	172
Angaben von Einstellungen für die Schemakonvertierung	176
Aktualisieren Ihrer Datenbankschemas	181
Speichern und Anwenden des Schemas	182
Verwenden von Erweiterungspaketen	184

Homogene Datenbankmigrationen	186
Unterstützte AWS-Regionen	187
Features	188
Einschränkungen	188
Übersicht	189
Einrichtung	190
Erstellen von IAM-Ressourcen	190
Einrichten eines Netzwerks	195
Erstellen von Quelldatenanbietern	199
Verwenden von MySQL oder MariaDB als Quelle	200
Verwenden von PostgreSQL als Quelle	204
Verwenden von MongoDB oder Amazon DocumentDB als Quelle	207
Erstellen von Zieldatenanbietern	211
Verwenden von MySQL oder MariaDB als Ziel	212
Verwenden von PostgreSQL als Ziel	213
Verwenden von Amazon DocumentDB als Ziel	215
Migrieren von Daten	216
Erstellen einer Datenmigration	216
Auswahlregeln	219
Verwaltung von Datenmigrationen	221
Überwachung von Datenmigrationen	223
Migrationsstatus	225
Migrieren von Daten aus MySQL	226
Migrieren von Daten aus PostgreSQL	228
Migrieren von Daten aus MongoDB	230
Fehlerbehebung	231
Erstellen einer Datenmigration	232
Starten einer Datenmigration	232
Verbindungsprobleme	233
Ansichten werden in PostgreSQL als Tabellen migriert	233
Arbeiten mit Migrationsprojekten	234
Erstellen einer Subnetzgruppe	235
Erstellen von Instance-Profilen	235
Erstellen von Datenanbietern	237
Erstellen von Migrationsprojekten	239
Verwaltung von Migrationsprojekten	241

Bewährte Methoden	243
Planen der Migration für AWS Database Migration Service	243
Schemakonvertierung	245
Überprüfung der AWS DMS-Dokumentation	245
Ausführen eines Machbarkeitsnachweises	246
Verbessern der Leistung	246
Verwenden Ihres eigenen Vor-Ort-Nameservers	252
Verwenden von Amazon Route 53 Resolver mit AWS DMS	253
Migrieren von Binary Large Objects (LOBs)	254
Verwenden des begrenzten LOB-Modus	254
Verbesserte LOB-Leistung	255
Verbessern der Leistung beim Migrieren umfangreicher Tabellen mit Zeilenfilterung	258
Fortlaufende Replikation	259
Reduzieren des Workloads Ihrer Quelldatenbank	260
Reduzierung von Engpässen in Ihrer Zieldatenbank	260
Verwendung der Datenvalidierung	261
Überwachen von Metriken	261
Ereignisse	262
Verwenden des Aufgabenprotokolls	262
Problembehandlung bei der Replikation mit Time Travel	263
Ändern des Benutzers und des Schemas für ein Oracle-Ziel	263
Ändern von Tabellen- und Index-Tabellenräumen für ein Oracle-Ziel	264
Aktualisieren einer Replikations-Instance	265
Grundlegendes zu Migrationskosten	266
Arbeiten mit Serverless AWS DMS	267
Komponenten von DMS Serverless	268
Unterstützte Engine-Versionen	271
Erstellen einer Serverless-Anwendung	272
AWS DMS Serverlose Replikationen ändern	274
Compute-Konfiguration	278
Grundlegendes zur automatischen Skalierung im serverlosen Modus AWS DMS	280
Überwachung serverloser Replikationen AWS DMS	281
Einschränkungen von Serverless	286
Arbeiten mit Replikations-Instances	288
Auswahl der Typen von Replikations-Instances	293
Entscheidung, welche Instance-Klasse verwendet werden soll	298

Burstable-Instances im unbegrenzten Modus	299
Bestimmen der Größe einer Replikations-Instance	300
Zu berücksichtigende Faktoren	301
Häufige Probleme	302
Bewährte Methoden	303
Versionen der Replikations-Engine	304
Aktualisieren der Engine-Version über die Konsole	304
Aktualisierung der Engine-Version mit dem AWS CLI	305
Öffentliche und private Replikations-Instances	306
IP-Adressierung und Netzwerktypen	306
Einrichten eines Netzwerks für eine Replikations-Instance	308
Netzwerkkonfigurationen für die Datenbankmigration	309
Erstellen einer Replikationssubnetzgruppe	318
Auflösen von Domain-Endpunkten mithilfe von DNS	320
Festlegen eines Verschlüsselungsschlüssels	320
Erstellen einer Replikations-Instance	321
Ändern einer Replikations-Instance	328
Neustarten einer Replikations-Instance	334
Löschen einer Replikations-Instance	337
DMS-Wartungszeitfenster	339
Auswirkungen der Wartung auf vorhandene Migrationsaufgaben	339
Ändern der Einstellung des Wartungszeitfensters	340
Endpunkte	342
Erstellen der Quell- und Zielendpunkte	342
Quellen für die Datenmigration	348
Verwenden von Oracle als Quelle	349
Verwenden von SQL Server als Quelle	422
Verwenden einer Azure SQL-Datenbank als Quelle	454
Verwenden einer Azure-SQL-Managed-Instance als Quelle	454
Verwenden von Azure Database für PostgreSQL als Quelle	455
Verwendung einer Azure-Datenbank für MySQL als Quelle	456
Verwendung von OCI MySQL Heatwave als Quelle	457
Verwenden von Google Cloud for MySQL als Quelle	458
Verwenden von Google Cloud für PostgreSQL als Quelle	458
Verwenden von PostgreSQL als Quelle	460
Verwenden von MySQL als Quelle	499

Verwenden von SAP ASE als Quelle	513
Verwenden von MongoDB als Quelle	522
Verwenden von Amazon DocumentDB als Quelle	541
Verwenden von Amazon S3 als Quelle	559
Verwenden von IBM Db2 LUW als Quelle	572
Verwendung von IBM Db2 für z/OS als Quelle	580
Ziele für die Datenmigration	621
Verwenden von Oracle als Ziel	623
Verwenden von SQL Server als Ziel	635
Verwenden von PostgreSQL als Ziel	641
Verwenden von MySQL als Ziel	655
Verwenden von Amazon Redshift als Ziel	664
Verwenden von SAP ASE als Ziel	691
Verwenden von Amazon S3 als Ziel	694
Verwenden von Amazon DynamoDB als Ziel	747
Verwenden von Amazon Kinesis Data Streams als Ziel	769
Verwenden von Apache Kafka als Ziel	789
Verwenden von OpenSearch als Ziel	817
Verwenden von Amazon DocumentDB als Ziel	824
Verwenden von Amazon Neptune als Ziel	838
Verwenden von Redis als Ziel	855
Verwenden von Babelfish als Ziel	863
Verwenden von Amazon Timestream als Ziel	872
Verwenden von Db2 als Ziel	883
VPC-Endpunkte für die Datenmigration	885
Wer ist bei einer Migration auf AWS-DMS-Version 3.4.7 und höher betroffen?	886
Wer ist bei einer Migration auf AWS-DMS-Version 3.4.7 und höher nicht betroffen?	886
Vorbereitung einer Migration auf AWS-DMS-Version 3.4.7 oder höher	886
Unterstützte DDL-Anweisungen	888
Aufgaben	889
Erstellen einer Aufgabe	894
Task settings (Aufgabeneinstellungen)	904
Einrichten der LOB-Unterstützung	960
Erstellen mehrerer Aufgaben	963
Fortlaufende Replikationsaufgaben	963
Starten der Replikation von einem CDC-Startpunkt aus	965

Durchführen der bidirektionalen Replikation	971
Ändern einer Aufgabe	975
Verschieben einer Aufgabe	976
Erneutes Laden von Tabellen während einer Aufgabe	977
AWS Management Console	978
Table mapping (Tabellenzuweisung)	979
Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole	980
Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON	985
Auswahlregeln und Aktionen	986
Platzhalter bei der Tabellenzuordnung	994
Transformationsregeln und Aktionen	995
Verwenden von Transformationsregelausdrücken zum Definieren von Spalteninhalten	1020
Regeln und Operationen für Tabellen- und Sammlungseinstellungen	1035
Verwenden von Quellfiltern	1069
Anwenden von Filtern	1070
Filtern nach Zeit und Datum	1076
Aktivieren und Verwenden von Vormigrationsbewertungen	1077
Voraussetzungen	1078
Angaben, Starten und Anzeigen von Bewertungsläufen	1081
Individuelle Bewertungen	1086
Starten und Anzeigen von Datentypbewertungen	1109
Problembehandlung bei Bewertungsläufen	1113
Angaben zusätzlicher Daten	1114
Überwachen von Aufgaben	1115
Aufgabenstatus	1117
Tabellenstatus während der Aufgaben	1120
Überwachen von Replikationsaufgaben mit Amazon CloudWatch	1121
AWS Database Migration Service-Metriken	1123
Metriken zur Replikations-Instance	1126
Metriken für die Replikationsaufgabe	1129
Anzeigen und Verwalten von AWS-DMS-Protokollen	1133
Protokollierung von AWS DMS-API-Aufrufen mit AWS CloudTrail	1135
AWS DMS-Informationen in CloudTrail	1135
Grundlagen zu AWS DMS-Protokolldateieinträgen	1136
Kontext-Protokollierung	1140
Objekttypen	1140

Beispiele für die Protokollierung	1142
Einschränkungen	1143
Arbeiten mit EventBridge-Ereignissen	1145
Verwenden von Amazon-EventBridge-Ereignisregeln für AWS DMS	1146
AWS DMS-Ereigniskategorien und -Ereignismeldungen	1147
ReplicationInstance-Ereignismeldungen	1147
ReplicationTask-Ereignismeldungen	1152
Replikations-Ereignismeldungen	1154
Arbeiten mit Amazon-SNS-Ereignissen	1157
verschieben von Event-Abonnements zu Amazon EventBridge	1157
Arbeiten mit Amazon-SNS-Ereignissen und Benachrichtigungen	1158
AWS-DMS-Ereigniskategorien und -Ereignismeldungen für SNS-Benachrichtigungen	1160
Abonnieren von AWS-DMS-Ereignisbenachrichtigungen mithilfe von SNS	1164
Verwendung von AWS Management Console	1164
Überprüfen der Zugriffsrichtlinie für Ihr SNS-Thema	1167
Datenvalidierung	1169
Replikationsaufgaben-Statistiken	1170
Statistiken zu Replikationsaufgaben mit Amazon CloudWatch	1173
Erneutes Validieren von Tabellen während einer Aufgabe	1174
AWS Management Console	1174
Ändern von Validierungsregeln mithilfe des JSON-Editors	1175
Reine Validierungsaufgaben	1176
Reine Volllastvalidierung	1176
Reine CDC-Validierung	1177
Anwendungsfälle für die reine Validierung	1177
Fehlerbehebung	1178
Redshift-Validierungsleistung	1180
Einschränkungen	1181
S3-Validierung	1182
Voraussetzungen	1183
Berechtigungen	1184
Einschränkungen	1185
Reine Validierungsaufgaben	1186
Markieren von Ressourcen	1188
API	1190
Sicherheit	1192

Datenschutz	1195
Datenverschlüsselung	1195
Richtlinie für den Datenverkehr zwischen Netzwerken	1196
Datenschutz in DMS Fleet Advisor	1197
Identity and Access Management	1198
Zielgruppe	1198
Authentifizierung mit Identitäten	1199
Verwalten des Zugriffs mit Richtlinien	1203
Funktionsweise AWS Database Migration Service von mit IAM	1205
Beispiele für identitätsbasierte Richtlinien	1213
Beispiele für eine ressourcenbasierte Richtlinie	1222
Verwenden von Secrets für den Zugriff auf Ressourcen	1227
Verwenden von serviceverknüpften Rollen	1237
Fehlerbehebung	1245
Erforderliche IAM-Berechtigungen	1248
IAM-Rollen für die CLI und API	1253
Serviceübergreifende Confused-Deputy-Prävention	1259
Von AWS verwaltete Richtlinien	1262
Compliance-Validierung	1269
Ausfallsicherheit	1271
Sicherheit der Infrastruktur	1272
Differenzierte Zugriffskontrolle	1276
Verwenden von Ressourcennamen für die Zugriffskontrolle	1276
Verwendung von Tags zur Zugriffssteuerung	1279
Festlegen eines Verschlüsselungsschlüssels	1287
Netzwerksicherheit	1290
Verwenden von SSL	1292
Einschränkungen bei der Verwendung von SSL mit AWS DMS	1294
Verwalten von Zertifikaten	1295
Aktivieren von SSL für einen MySQL-kompatiblen, PostgreSQL- oder SQL Server- Endpunkt	1295
Ändern des Datenbankpassworts	1298
Einschränkungen	1299
Ressourcenkontingente für AWS Database Migration Service	1299
Informationen zur API-Anforderungsdrosselung	1301
Unterstützung bei der Fehlersuche und Diagnose	1302

Migrationsaufgaben werden langsam ausgeführt	1303
Die Statusleiste der Aufgabe bewegt sich nicht	1304
Die Aufgabe ist abgeschlossen, aber es wurde nichts migriert	1304
Fremdschlüssel und sekundäre Indizes fehlen	1304
AWS DMS erstellt keine CloudWatch Protokolle	1305
Bei der Verbindung mit Amazon RDS treten Probleme auf	1306
Fehlermeldung: Falsche Thread-Verbindungszeichenfolge: Falscher Thread-Wert 0	1306
Es treten Netzwerkprobleme auf	1306
CDC ist nach Volllast hängen geblieben	1307
Fehler durch Primärschlüsselverletzung beim Neustart einer Aufgabe	1308
Erstes Laden des Schemas fehlgeschlagen	1308
Aufgaben mit unbekanntem Fehler fehlgeschlagen	1308
Bei erneutem Laden der Aufgabe werden Tabellen von Beginn an geladen	1308
Die Anzahl der Tabellen pro Aufgabe verursacht Probleme	1309
Aufgaben schlagen fehl, wenn ein Primärschlüssel in der LOB-Spalte erstellt wird	1309
Doppelte Datensätze in einer Zieltabelle ohne Primärschlüssel	1309
Quellendpunkte fallen in den reservierten IP-Bereich	1309
Zeitstempel sind in Amazon-Athena-Abfragen unleserlich	1310
Fehlersuche bei Verwendung von Oracle	1310
Abrufen von Daten aus Ansichten	1311
Migrieren von LOBs aus Oracle 12c	1311
Wechseln zwischen Oracle LogMiner und Binary Reader	1311
Fehler: Oracle CDC angehalten 122301 Oracle CDC maximale Wiederholversuche überschritten.	1312
Automatisches Hinzufügen der zusätzlichen Protokollierung zu einem Oracle- Quellendpunkt	1312
LOB-Änderungen werden nicht erfasst	1313
Fehler: ORA-12899: Wert zu groß für Spalte <i>column-name</i>	1313
Datentyp NUMBER wird nicht richtig interpretiert	1314
Datensätze fehlen bei Volllast	1314
Table Error	1314
Fehler: Cannot retrieve Oracle archived Redo log destination ids	1315
Bewertung der Leseleistung von Oracle-Redo- oder -Archivprotokollen	1315
Fehlersuche bei Verwendung von MySQL	1317
CDC-Aufgabe schlägt für Amazon RDS-DB-Instance-Endpoint aufgrund deaktivierter binärer Protokollierung fehl	1318

Verbindungen mit einer MySQL-Ziel-Instance werden während einer Aufgabe getrennt	1318
Hinzufügen von Autocommit zu einem MySQL-kompatiblen Endpunkt	1319
Deaktivieren von Fremdschlüsseln auf einem MySQL-kompatiblen Zielendpunkt	1320
Durch Fragezeichen ersetzte Zeichen	1320
„Bad event“-Protokolleinträge	1320
Change Data Capture (CDC) mit MySQL 5.5	1321
Erhöhen der Aufbewahrungszeit für binäre Protokolle für Amazon RDS-DB-Instances	1321
Protokollmeldung: Einige Änderungen von der Quelldatenbank hatten bei Anwendung auf die Zieldatenbank keine Auswirkungen.	1321
Fehler: Bezeichner zu lang	1321
Fehler: Felddatenumwandlung schlägt aufgrund nicht unterstützten Zeichensatzes fehl	1322
Fehler: Codeseite 1252 zu UTF8 [120112] eine Felddatenkonvertierung ist fehlgeschlagen	1322
Indizes, Fremdschlüssel oder kaskadierende Aktualisierungen oder Löschungen wurden nicht migriert	1323
Fehlersuche bei Verwendung von PostgreSQL	1325
Verkürzte JSON-Datentypen	1326
Spalten eines benutzerdefinierten Datentyps werden nicht korrekt migriert	1326
Fehler: Kein Schema zum Erstellen ausgewählt	1327
Lösch- und Aktualisierungsvorgänge für eine Tabelle werden bei Verwendung von CDC nicht repliziert	1327
Truncate-Anweisungen werden nicht ordnungsgemäß propagiert	1327
Verhindern, dass PostgreSQL DDL erfasst	1327
Auswahl des Schemas, in dem Datenbankobjekte für die DDL-Erfassung erstellt werden ..	1327
Oracle-Tabellen fehlen nach Migration zu PostgreSQL	1328
ReplicationSlotDiskUsage erhöht und restart_Isn stoppt bei langen Transaktionen wie ETL-Workloads	1328
Für Aufgabe, die Ansicht als Quelle verwendet, wurden keine Zeilen kopiert	1328
Fehlersuche bei Verwendung von Microsoft SQL Server	1329
Fehler bei Erfassung von Änderungen für SQL Server-Datenbank	1329
Fehlende Identitätsspalten	1329
Fehler: SQL Server unterstützt keine Publikationen	1330
Änderungen werden im Ziel nicht angezeigt	1330
Uneinheitliche Tabelle, die über Partitionen hinweg zugeordnet ist	1330
Fehlersuche bei Verwendung von Amazon Redshift	1331
Laden in einen Amazon-Redshift-Cluster in einer anderen AWS-Region	1332

Fehler: Beziehung "attrep_apply_exceptions" bereits vorhanden	1332
Fehler mit Tabellen, deren Name mit „awsdms_changes“ beginnt	1332
Anzeige von Tabellen in Clustern mit Namen wie dms.awsdms_changes000000000XXXX	1332
Berechtigungen für die Verwendung mit Amazon Redshift erforderlich	1332
Fehlersuche bei Verwendung von Amazon Aurora MySQL	1333
Fehler: CHARACTER SET UTF8-Felder beendet durch ',' umschlossen von "" Zeilen beendet durch '\n'	1333
Fehlersuche bei Verwendung von SAP ASE	1333
Fehler: LOB-Spalten haben NULL-Werte, wenn die Quelle einen zusammengesetzten eindeutigen Index mit NULL-Werten hat	1333
Fehlersuche bei Verwendung von IBM Db2	1334
Fehler: Fortsetzen ab Zeitstempel wird nicht unterstützt	1334
Fehlersuche im Zusammenhang mit der Latenz	1334
CDC-Latenztypen	1335
Häufige Ursachen für CDC-Latenz	1336
Fehlersuche bei Latenzproblemen	1340
Arbeiten mit Diagnoseunterstützungsskripts	1356
Oracle-Unterstützungsskripts	1357
SQL-Server-Unterstützungsskripts	1361
MySQL-kompatible Unterstützungsskripts	1387
PostgreSQL-Unterstützungsskripts	1389
Arbeiten mit dem Diagnoseunterstützungs-AMI	1392
Starten Sie eine neue diagnostische AWS DMS-Amazon-EC2-Instance	1393
Erstellen einer IAM-Rolle	1393
Ausführen von Diagnosetests	1394
Nächste Schritte	1399
AMI-IDs nach Region	1400
Referenz	1402
AWS-DMS-Datentypen	1402
Versionshinweise	1405
AWS DMS Versionshinweise zu 3.5.2	1406
AWS DMS Versionshinweise zu 3.5.1	1406
AWS DMS 3.5.0 Beta-Versionshinweise	1418
AWS DMS Versionshinweise zu 3.4.7	1424
AWS DMS Versionshinweise zu 3.4.6	1434
AWS DMS Versionshinweise zu 3.4.5	1441

AWS DMS Versionshinweise zu 3.4.4	1444
AWS DMS Versionshinweise zu 3.4.3	1447
AWS DMS Versionshinweise zu 3.4.2	1449
AWS DMS Versionshinweise zu 3.4.1	1451
AWS DMS Versionshinweise zu 3.4.0	1452
AWS DMS Versionshinweise zu 3.3.4	1454
AWS DMS Versionshinweise zu 3.3.3	1455
Dokumentverlauf	1457
AWS-Glossar	1462
.....	mcdlxiii

Was ist AWS Database Migration Service?

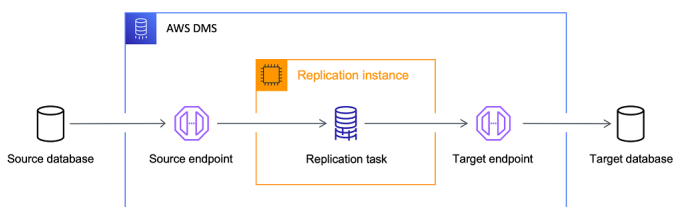
AWS Database Migration Service (AWS DMS) ist ein Cloud-Service, der die Migration von relationalen Datenbanken, Data Warehouses, NoSQL-Datenbanken und anderen Arten von Datenspeichern ermöglicht. Sie können AWS DMS verwenden, um Ihre Daten in die AWS Cloud oder zwischen Kombinationen aus Cloud und On-Premises-Einrichtungen zu migrieren.

Mit AWS DMS können Sie Ihre Quelldatenspeicher ermitteln, Ihre Quellschemas konvertieren und Ihre Daten migrieren.

- Zum Ermitteln Ihrer Quelldateninfrastruktur können Sie DMS Fleet Advisor verwenden. Dieser Service erfasst Daten von Ihren On-Premises-Datenbank- und Analyseservern und erstellt ein Inventar von Servern, Datenbanken und Schemas, die Sie in die AWS Cloud migrieren können.
- Für Migrationen zu einer anderen Datenbank-Engine können Sie die DMS-Schemakonvertierung verwenden. Dieser Service bewertet Ihre Quellschemas automatisch und konvertiert sie in eine neue Ziel-Engine. Alternativ können Sie AWS Schema Conversion Tool (AWS SCT) auf Ihren lokalen PC herunterladen, um Ihre Quellschemas zu konvertieren.
- Nachdem Sie Ihre Quellschemas konvertiert und den konvertierten Code auf Ihre Zieldatenbank angewendet haben, können Sie Ihre Daten mithilfe von AWS DMS migrieren. Sie können einmalige Migrationen durchführen oder laufende Änderungen replizieren, um Quellen und Ziele synchron zu halten. Da AWS DMS Teil der AWS Cloud ist, profitieren Sie von der Kosteneffizienz, der schnellen Markteinführung, der Sicherheit und der Flexibilität, die AWS-Services bieten.

Grundsätzlich ist AWS DMS ein Server in der AWS Cloud, auf dem Replikationssoftware ausgeführt wird. Sie legen eine Quell- und eine Zielverbindung an, um AWS DMS mitzuteilen, wo Daten extrahiert und geladen werden sollen. Dann planen Sie eine Aufgabe, die auf diesem Server ausgeführt wird, um Ihre Daten zu verschieben. AWS DMS erstellt die Tabellen und zugehörigen Primärschlüssel, falls diese nicht bereits im Ziel vorhanden sind. Sie können die Zieltabellen selbst erstellen, wenn Sie dies vorziehen. Sie können auch AWS Schema Conversion Tool (AWS SCT) verwenden, um einige oder alle Zieltabellen, Indizes, Ansichten, Auslöser usw. zu erstellen.

Das folgende Diagramm veranschaulicht den Replikationsprozess in AWS DMS.



Referenzen

- AWS-Regionen, die AWS DMS unterstützen – Informationen dazu, welche AWS-Regionen AWS DMS unterstützen, finden Sie unter [Mit einer AWS DMS Replikationsinstanz arbeiten](#).
- Kosten der Datenbankmigration – Informationen über die Kosten der Datenbankmigration finden Sie auf der [Seite mit Preisen für AWS Database Migration Service](#).
- Features und Vorteile von AWS DMS – Informationen zu den Features und Vorteilen von AWS DMS finden Sie unter [Funktionen des AWS Database Migration Service](#).
- Verfügbare Datenbankoptionen – Weitere Informationen zu den verschiedenen Datenbankoptionen, die in Amazon Web Services verfügbar sind, finden Sie unter [Choosing the right database for your organization](#).

Von AWS DMS durchgeführte Migrationsaufgaben

AWS DMS übernimmt viele der schwierigen oder langwierigen Aufgaben, die mit einem Migrationsprojekt verbunden sind:

- Bei einer herkömmlichen Lösung müssen Sie Kapazitätsanalysen durchführen, Hardware und Software beschaffen, Systeme installieren und verwalten sowie die Installation testen und debuggen. AWS DMS verwaltet die Bereitstellung, Verwaltung und Überwachung aller für die Migration benötigten Hard- und Softwarekomponenten automatisch. Ihre Migration kann innerhalb weniger Minuten nach dem Start des AWS DMS-Konfigurationsprozesses abgeschlossen werden.
- Mit AWS DMS können Sie Ihre Migrationsressourcen nach Bedarf nach oben (oder unten) skalieren, um sie an Ihren tatsächlichen Workload anzupassen. Wenn Sie beispielsweise feststellen, dass Sie zusätzlichen Speicherplatz benötigen, können Sie den zugewiesenen Speicherplatz problemlos erhöhen und die Migration in der Regel innerhalb von Minuten erneut starten.
- AWS DMS verwendet ein Servicemodell mit nutzungsabhängiger Abrechnung. Im Gegensatz zu herkömmlichen Lizenzmodellen mit Vorabkosten für die Anschaffung und laufenden Unterhaltungskosten zahlen Sie nur für AWS DMS-Ressourcen, die Sie nutzen.
- AWS DMS verwaltet automatisch die gesamte Infrastruktur, die Ihren Migrationsserver unterstützt, einschließlich Hardware und Software, Software-Patching und Fehlerberichten.
- AWS DMS bietet ein automatisches Failover. Wenn Ihr primärer Replikationsserver aus irgendeinem Grund ausfällt, übernimmt ein Backup-Replikationsserver, ohne dass der Dienst unterbrochen wird.

- AWS DMS Fleet Advisor inventarisiert Ihre Dateninfrastruktur automatisch. Der Service erstellt Berichte, die Ihnen helfen, Kandidaten für eine Migration zu identifizieren und die Migration zu planen.
- Die AWS-DMS-Schemakonvertierung bewertet automatisch die Komplexität Ihrer Migration für Ihren Quelldatenanbieter. Darüber hinaus konvertiert dieses Tool Datenbankschemas und Codeobjekte in ein mit der Zieldatenbank kompatibles Format und wendet dann den konvertierten Code an.
- AWS DMS kann Sie dabei unterstützen, auf eine moderne, vielleicht kostengünstigere Datenbank-Engine umzusteigen, als die, die Sie gerade betreiben. Beispielsweise kann AWS DMS Ihnen helfen, die Vorteile der verwalteten Datenbankservices zu nutzen, die von Amazon Relational Database Service (Amazon RDS) oder Amazon Aurora bereitgestellt werden. Der Service kann Sie auch dabei unterstützen, auf den verwalteten Data-Warehouse-Service von Amazon Redshift, NoSQL-Plattformen wie Amazon DynamoDB oder kostengünstige Speicherplattformen wie Amazon Simple Storage Service (Amazon S3) umzusteigen. Umgekehrt unterstützt AWS DMS auch den Prozess, wenn Sie sich von der alten Infrastruktur verabschieden, aber weiterhin die gleiche Datenbank-Engine verwenden wollen.
- AWS DMS unterstützt fast alle der heute beliebtesten DBMS-Engines als Quellendpunkte. Weitere Informationen finden Sie unter [Quellen für die Datenmigration](#).
- AWS DMS bietet eine breite Palette an verfügbaren Ziel-Engines. Weitere Informationen finden Sie unter [Ziele für die Datenmigration](#).
- Sie können von jeder der unterstützten Datenquellen auf jedes der unterstützten Datenziele migrieren. AWS DMS unterstützt vollständig heterogene Datenmigrationen zwischen den unterstützten Engines.
- AWS DMS sorgt für eine sichere Datenmigration. Daten im Ruhezustand werden mit der AWS Key Management Service (AWS KMS)-Verschlüsselung verschlüsselt. Während der Migration können Sie Secure Socket Layers (SSL) verwenden, um Ihre übertragenen Daten auf dem Weg von der Quelle zum Ziel zu verschlüsseln.

So funktioniert AWS der Database Migration Service

AWS Database Migration Service (AWS DMS) ist ein Webdienst, mit dem Sie Daten von einem Quelldatenspeicher zu einem Zieldatenspeicher migrieren können. Diese beiden Datenspeicher werden als Endpunkte bezeichnet. Sie können zwischen den Quell- und Zielendpunkten migrieren, die die gleiche Datenbank-Engine verwenden, wie z. B. von einer Oracle-Datenbank in eine Oracle-Datenbank. Außerdem können Sie zwischen Quell- und Zielendpunkten migrieren, die verschiedene Datenbank-Engines verwenden, wie z. B. von einer Oracle-Datenbank in eine PostgreSQL-Datenbank. Die einzige Voraussetzung für die Nutzung AWS DMS ist, dass sich einer Ihrer Endpunkte in einem AWS Dienst befindet. Sie können es nicht verwenden AWS DMS , um von einer lokalen Datenbank zu einer anderen lokalen Datenbank zu migrieren.

Weitere Informationen über die Kosten der Datenbankmigration finden Sie auf der [Seite mit den Preisen für AWS Database Migration Service](#).

Verwenden Sie die folgenden Themen, um das besser zu verstehen. AWS DMS

Themen

- [Überblick über AWS DMS](#)
- [Komponenten von AWS DMS](#)
- [Quellen für AWS DMS](#)
- [Ziele für AWS DMS](#)
- [Erstellung eines Amazon-Ressourcennamens \(ARN\) für AWS DMS](#)
- [Verwendung AWS DMS mit anderen Diensten AWS](#)

Überblick über AWS DMS

Um eine Datenbankmigration durchzuführen, AWS DMS stellt man eine Verbindung zum Quelldatenspeicher her, liest die Quelldaten und formatiert die Daten für die Nutzung durch den Zieldatenspeicher. Anschließend werden die Daten in der Zieldatenspeicher geladen. Die meisten dieser Vorgänge werden im Arbeitsspeicher ausgeführt, wenn auch umfangreiche Transaktionen eventuell auf Festplatte gepuffert werden. Zwischengespeicherte Transaktionen und Protokolldateien werden ebenfalls auf Festplatte geschrieben.

Auf einer höheren Ebene gehen AWS DMS Sie bei der Verwendung wie folgt vor:

- Ermitteln Sie Datenbanken in Ihrer Netzwerkumgebung, die sich gut für eine Migration eignen.
- Konvertieren Sie Ihre Quelldatenbankschemata und die meisten Datenbankcodeobjekte automatisch in ein mit der Zieldatenbank kompatibles Format.
- Erstellen eines Replikationsservers.
- Erstellen der Quell- und Zielendpunkte, die über Verbindungsinformationen zu Ihren Datenspeichern verfügen.
- Erstellen Sie eine oder mehrere Migrationsaufgaben zum Migrieren von Daten zwischen den Quell- und Zieldatenspeichern.

Eine Aufgabe kann aus drei Hauptphasen bestehen:

- Migration vorhandener Daten (Volllast)
- Die Anwendung der zwischengespeicherten Änderungen
- Laufende Replikation (Erfassung von Datenänderungen)

Bei einer Volllastmigration, bei der vorhandene Daten aus der Quelle in das Ziel verschoben werden, werden Daten aus Tabellen im Quelldatenspeicher in Tabellen im Zieldatenspeicher AWS DMS geladen. Während des vollständigen Ladevorgangs werden alle Änderungen an den Tabellen, die gerade geladen werden, auf dem Replikationsserver im Cache gespeichert. Dies sind die zwischengespeicherten Änderungen. Es ist wichtig zu beachten, dass AWS DMS Änderungen für eine bestimmte Tabelle erst erfasst werden, wenn der vollständige Ladevorgang für diese Tabelle gestartet wurde. Mit anderen Worten: Die Änderungserfassung beginnt für jede einzelne Tabelle zu einem jeweils anderen Zeitpunkt.

Wenn der vollständige Ladevorgang für eine bestimmte Tabelle abgeschlossen ist, AWS DMS werden sofort die zwischengespeicherten Änderungen für diese Tabelle übernommen. Sobald die Tabelle geladen und die zwischengespeicherten Änderungen übernommen wurden, AWS DMS beginnt die Erfassung der Änderungen als Transaktionen für die laufende Replikationsphase. Wenn bei einer Transaktion Tabellen noch nicht vollständig geladen sind, werden die Änderungen lokal auf der Replikations-Instance gespeichert. Nachdem alle AWS DMS zwischengespeicherten Änderungen übernommen wurden, sind die Tabellen transaktionskonsistent. An diesem Punkt AWS DMS geht es zur Phase der laufenden Replikation über, in der die Änderungen als Transaktionen angewendet werden.

Zu Beginn der laufenden Replikationsphase führt ein Rückstand bei den Transaktionen im Allgemeinen zu einer Verzögerung zwischen der Quell- und Zieldatenbank. Schließlich erreicht

die Migration aber einen stabilen Zustand, nachdem dieser Transaktionsrückstand abgearbeitet wurde. An diesem Punkt können Sie Ihre Anwendungen herunterfahren und zulassen, dass alle verbleibenden Transaktionen auf die Zieldatenbank angewendet werden. Wenn Sie Ihre Anwendungen dann starten, verweisen sie auf die Zieldatenbank.

AWS DMS erstellt die Zielschemaobjekte, die für die Durchführung einer Datenmigration erforderlich sind. Sie können einen minimalistischen Ansatz verwenden AWS DMS und nur die Objekte erstellen, die für eine effiziente Migration der Daten erforderlich sind. Mit diesem Ansatz werden Tabellen, Primärschlüssel und in einigen Fällen eindeutige Indizes AWS DMS erstellt, es werden jedoch keine anderen Objekte erstellt, die für eine effiziente Migration der Daten aus der Quelle nicht erforderlich sind.

Alternativ können Sie die interne DMS-Schemakonvertierung verwenden, AWS DMS um Ihre Quelldatenbankschemas und die meisten Datenbankcode-Objekte automatisch in ein Format zu konvertieren, das mit der Zieldatenbank kompatibel ist. Diese Konvertierung umfasst Tabellen, Ansichten, gespeicherte Prozeduren, Funktionen, Datentypen, Synonyme usw. Alle Objekte, die DMS Schema Conversion nicht automatisch konvertieren kann, sind deutlich gekennzeichnet. Um die Migration abzuschließen, können Sie diese Objekte manuell konvertieren.

Komponenten von AWS DMS

In diesem Abschnitt werden die internen Komponenten von AWS DMS und deren Zusammenspiel beschrieben, um Ihre Datenmigration durchzuführen. Das Verständnis der zugrunde liegenden Komponenten von AWS DMS kann Ihnen helfen, Daten effizienter zu migrieren und einen besseren Einblick bei der Fehlersuche und -behebung zu erhalten.

Eine AWS DMS Migration besteht aus fünf Komponenten: Erkennung der zu migrierenden Datenbanken, automatische Schemakonvertierung, eine Replikationsinstanz, Quell- und Zielpunkte sowie eine Replikationsaufgabe. Sie erstellen eine AWS DMS Migration, indem Sie die erforderliche Replikationsinstanz, Endpunkte und Aufgaben in einem erstellen. AWS-Region

Database Discovery

DMS Fleet Advisor erfasst Daten aus mehreren Datenbankumgebungen, um Einblicke in Ihre Dateninfrastruktur zu erhalten. DMS Fleet Advisor erfasst Daten von Ihren On-Premises-Datenbank- und -Analyseservern von einem oder mehreren zentralen Standorten aus, ohne dass sie auf jedem Computer installiert werden müssen. Derzeit unterstützt DMS Fleet Advisor Microsoft-SQL-Server-, MySQL-, Oracle- und PostgreSQL-Datenbankserver.

Auf der Grundlage der in Ihrem Netzwerk gefundenen Daten erstellt DMS Fleet Advisor ein Inventar, das Sie überprüfen können, um festzustellen, welche Datenbankserver und Objekte überwacht werden sollen. Sobald Details zu diesen Servern, Datenbanken und Schemata erfasst werden, können Sie die Durchführbarkeit Ihrer geplanten Datenbankmigrationen analysieren.

Schema- und Codemigration

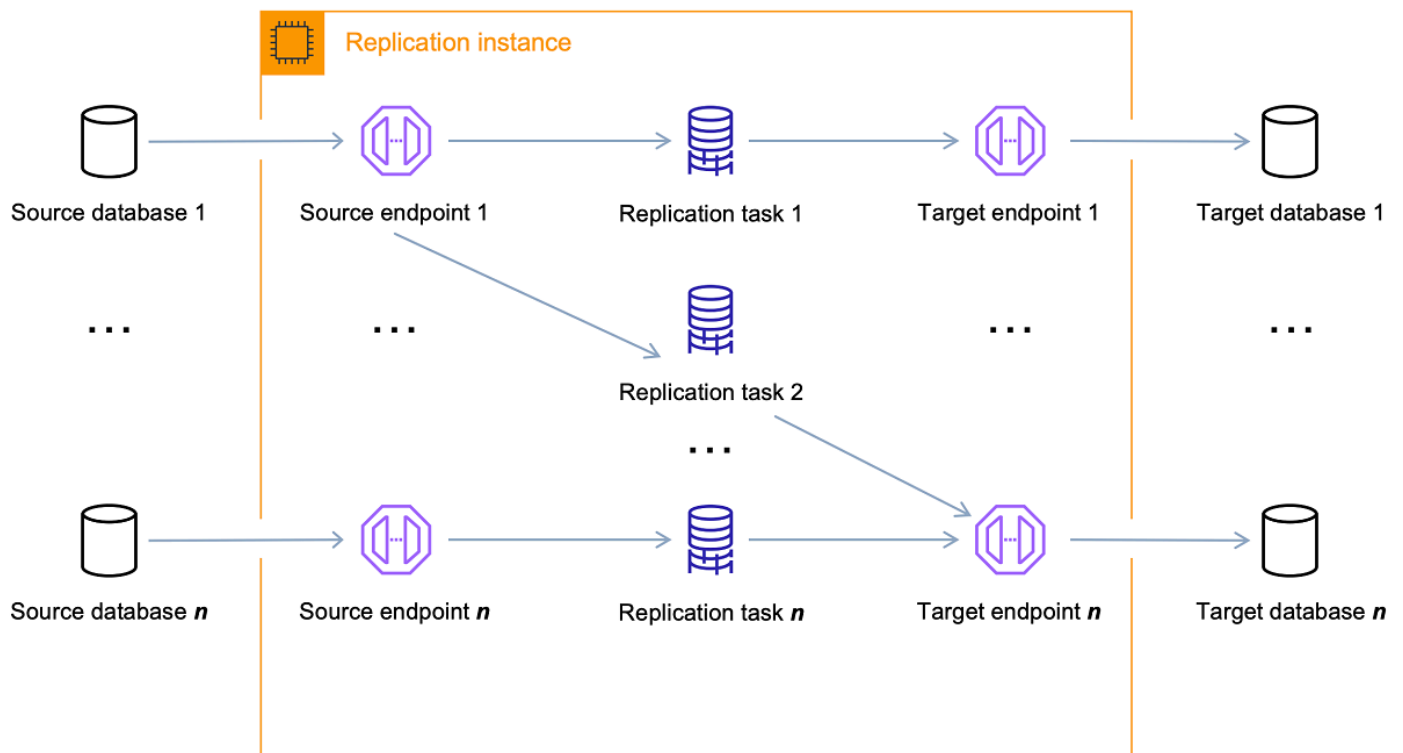
Die DMS-Schemakonvertierung in AWS DMS macht Datenbankmigrationen zwischen verschiedenen Datenbanktypen vorhersehbarer. Mithilfe von DMS Schema Conversion können Sie die Komplexität Ihrer Migration für Ihren Quelldatenanbieter bewerten und anschließend Datenbankschemata und Codeobjekte konvertieren. Sie können dann den konvertierten Code auf Ihre Zieldatenbank anwenden.

Allgemein arbeitet DMS Schema Conversion mit den folgenden drei Komponenten: Instance-Profile, Datenanbieter und Migrationsprojekte. Ein Instance-Profil spezifiziert Netzwerk- und Sicherheitseinstellungen. Ein Datenanbieter speichert Anmeldeinformationen für Datenbankverbindungen. Ein Migrationsprojekt enthält Datenanbieter, ein Instanzprofil und Migrationsregeln. AWS DMS verwendet Datenanbieter und ein Instanzprofil, um einen Prozess zu entwerfen, der Datenbankschemas und Codeobjekte konvertiert.

Replikations-Instance

Auf einer höheren Ebene ist eine AWS DMS Replikationsinstanz einfach eine verwaltete Amazon Elastic Compute Cloud (Amazon EC2) -Instance, die eine oder mehrere Replikationsaufgaben hostet.

Die folgende Abbildung zeigt ein Beispiel für eine Replikations-Instance mit mehreren zugehörigen Replikationsaufgaben.



Eine einzelne Replikationsinstanz kann je nach den Merkmalen Ihrer Migration und der Kapazität des Replikationsservers eine oder mehrere Replikationsaufgaben hosten. AWS DMS bietet eine Vielzahl von Replikationsinstanzen, sodass Sie die optimale Konfiguration für Ihren Anwendungsfall auswählen können. Weitere detaillierte Informationen über die verschiedenen Klassen von Replikations-Instances finden Sie unter [Auswahl der richtigen AWS DMS-Replikationsinstanz für Ihre Migration](#).

AWS DMS erstellt die Replikationsinstanz auf einer Amazon EC2 EC2-Instance. Einige der kleineren Instance-Klassen reichen für das Testen des Service oder für kleine Migrationen aus. Wenn Ihre Migration eine große Anzahl von Tabellen umfasst oder Sie mehrere gleichzeitige Replikationsaufgaben ausführen möchten, sollten Sie in Betracht ziehen, eine der größeren Instances zu verwenden. Wir empfehlen diese Vorgehensweise, da AWS DMS eine erhebliche Menge an Arbeitsspeicher und CPU-Leistung verbraucht.

Abhängig von der Klasse der Amazon-EC2-Instance, die Sie auswählen, umfasst Ihre Replikations-Instance entweder 50 GB oder 100 GB Datenspeicher. Dieser Umfang ist normalerweise ausreichend für die meisten Kunden. Wenn Ihre Migration jedoch große Transaktionen oder eine große Menge an Datenänderungen beinhaltet, sollten Sie die Basisspeicherzuordnung erhöhen. Change Data Capture (CDC) kann dazu führen, dass Daten auf die Festplatte geschrieben werden, je nachdem, wie schnell das Ziel die Änderungen

schreiben kann. Da Protokolldateien auch auf die Festplatte geschrieben werden, führt eine Erhöhung des Schweregrads für die Protokollierung auch zu einem höheren Speicherverbrauch.

AWS DMS kann mithilfe einer Multi-AZ-Bereitstellung Hochverfügbarkeit und Failover-Unterstützung bieten. In einer Multi-AZ-Bereitstellung AWS DMS wird automatisch ein Standby-Replikat der Replikationsinstanz in einer anderen Availability Zone bereitgestellt und verwaltet. Die primäre Replikations-Instance wird auf das Standby-Replikat repliziert. Wenn die primäre Replikations-Instance ausfällt oder nicht mehr reagiert, nimmt der Standby-Modus alle laufenden Tasks mit minimaler Unterbrechung wieder auf. Da die primäre Replikations-Instance ihren Status ständig in den Standby-Modus repliziert, verursacht die Multi-AZ-Bereitstellung einen gewissen Leistungs-Overhead.

Ausführlichere Informationen zur AWS DMS Replikationsinstanz finden Sie unter [Mit einer AWS DMS Replikationsinstanz arbeiten](#)

Anstatt eine Replikationsinstanz zu erstellen und zu verwalten, können Sie Ihre Replikation mithilfe von AWS DMS Serverless automatisch AWS DMS bereitstellen lassen. Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Serverless](#).

Endpunkt

AWS DMS verwendet einen Endpunkt, um auf Ihren Quell- oder Zieldatenspeicher zuzugreifen. Die spezifischen Verbindungsinformationen sind je nach Datenspeicher unterschiedlich, aber in der Regel geben Sie beim Anlegen eines Endpunkts die folgenden Informationen an.

- Endpunkttyp – Quelle oder Ziel.
- Engine-Typ – Typ der Datenbank-Engine, z. B. Oracle oder PostgreSQL.
- Servername — Servername oder IP-Adresse, die erreicht AWS DMS werden kann.
- Port – Port-Nummer für die Datenbankserver-Verbindungen.
- Verschlüsselung – Secure Socket Layer (SSL)-Modus, wenn SSL zum Verschlüsseln der Verbindung verwendet wird.
- Anmeldeinformationen – Benutzername und Passwort für ein Konto mit den erforderlichen Zugriffsrechten.

Wenn Sie mit der AWS DMS Konsole einen Endpunkt erstellen, verlangt die Konsole, dass Sie die Endpunktverbindung testen. Der Test muss erfolgreich sein, bevor Sie den Endpunkt in einer AWS DMS Aufgabe verwenden können. Wie die Verbindungsinformationen sind auch die spezifischen Prüfkriterien für verschiedene Engine-Typen unterschiedlich. Im Allgemeinen überprüft AWS DMS, ob die Datenbank unter dem angegebenen Servernamen und Port

vorhanden ist, und ob die mitgelieferten Anmeldeinformationen verwendet werden können, um eine Verbindung zur Datenbank mit den für eine Migration erforderlichen Berechtigungen herzustellen. Wenn der Verbindungstest erfolgreich ist, werden Schemainformationen AWS DMS heruntergeladen und gespeichert, um sie später bei der Aufgabenkonfiguration zu verwenden. Schemainformationen können beispielsweise Tabellendefinitionen, Primärschlüsseldefinitionen und eindeutige Schlüsseldefinitionen beinhalten.

Mehrere Replikationsaufgaben können einen einzelnen Endpunkt verwenden. Beispielsweise könnten Sie zwei logisch getrennte Anwendungen auf derselben Quelldatenbank hosten, die Sie separat migrieren möchten. In diesem Fall erstellen Sie zwei Replikationsaufgaben, eine für jede Gruppe von Anwendungstabellen. Sie können in beiden Aufgaben denselben AWS DMS Endpunkt verwenden.

Sie können das Verhalten eines Endpunkts mithilfe der Endpunkteinstellungen anpassen. Endpunkteinstellungen können verschiedene Verhaltensweisen kontrollieren, wie z. B. Protokollierungsdetail, Dateigröße und andere Parameter. Für jeden Datenspeicher-Engine-Typ sind unterschiedliche Endpunkteinstellungen verfügbar. Sie finden die spezifischen, Endpunkteinstellungen für jeden Datenspeicher im Quell- oder Ziel-Abschnitt für den jeweiligen Datenspeicher. Eine Liste unterstützter Quell- und Zieldatenspeicher finden Sie unter [Quellen für AWS DMS](#) und [Ziele für AWS DMS](#).

Ausführlichere Informationen zu AWS DMS Endpunkten finden Sie unter [Arbeiten mit AWS-DMS-Endpunkten](#).

Replikationsaufgaben

Sie verwenden eine AWS DMS Replikationsaufgabe, um einen Datensatz vom Quellendpunkt zum Zielendpunkt zu verschieben. Das Erstellen einer Replikations-Aufgabe ist der letzte Schritt, die Sie durchführen müssen, bevor Sie eine Migration beginnen.

Beim Erstellen einer Replikationsaufgabe geben Sie die folgenden Aufgabeneinstellungen an:

- Replikations-Instance – die Instance, die die Aufgabe hostet und ausführt
- Quellendpunkt
- Zielendpunkt
- Die Optionen für den Migrationstyp sind nachfolgend aufgeführt. Eine vollständige Erläuterung der Migrationsoptionen finden Sie unter [Erstellen einer Aufgabe](#).
 - Volllast (Migrieren vorhandener Daten) – Wenn Sie sich einen Ausfall leisten können, der ausreichend lange dauert, um Ihre vorhandenen Daten zu kopieren, eignet sich diese Option

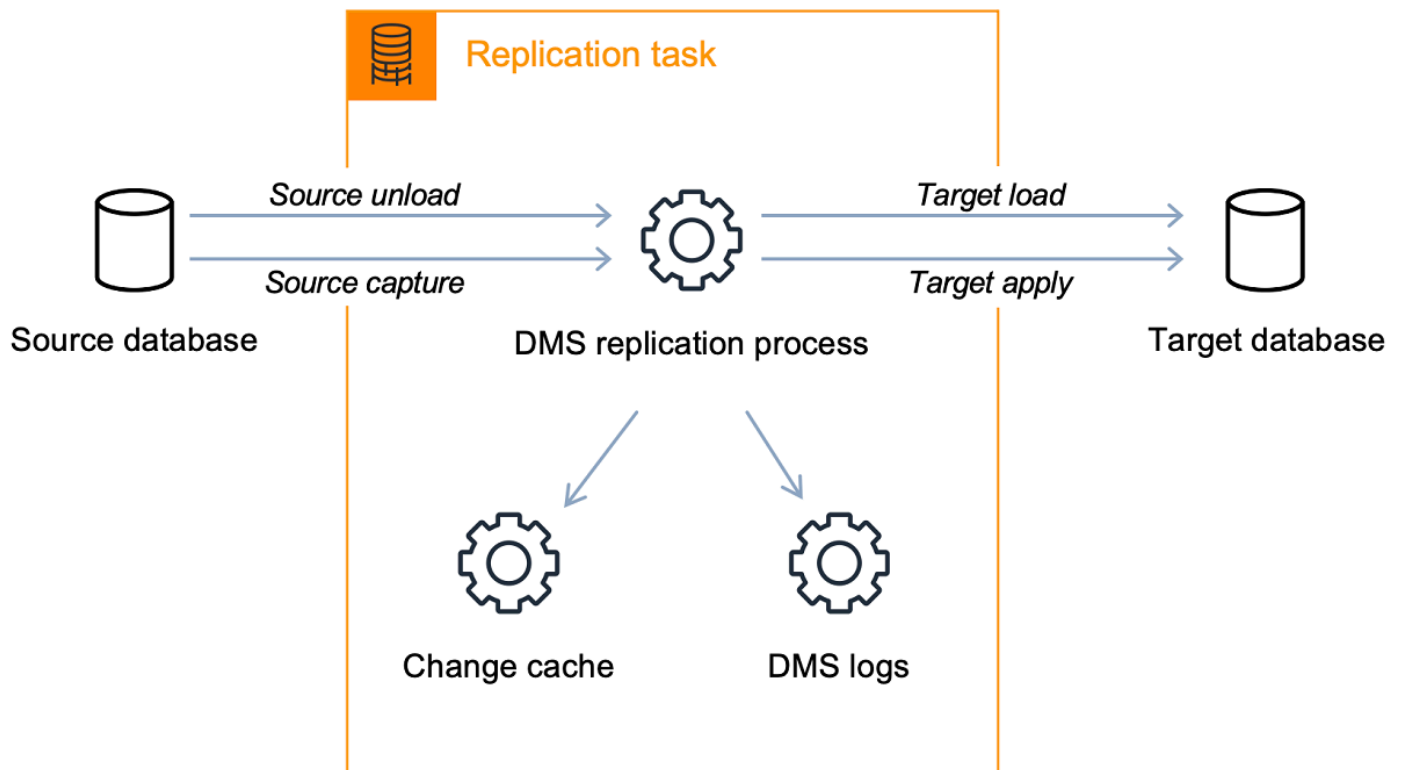
sehr gut. Bei dieser Option werden die Daten einfach aus Ihrer Quelldatenbank in Ihre Zieldatenbank migriert. Tabellen werden erstellt, wenn dies erforderlich ist.

- Volllast + CDC (Migrieren vorhandener Daten und Replizieren fortlaufender Änderungen) – Diese Option führt einen vollständigen Datenladevorgang durch und erfasst dabei Änderungen an der Quelle. Nachdem der vollständige Ladevorgang abgeschlossen ist, werden die erfassten Änderungen auf die Zieldatenbank angewendet. Am Ende erreicht die Anwendung der Änderungen einen stabilen Zustand. An diesem Punkt können Sie Ihre Anwendungen herunterfahren. Die verbleibenden Änderungen werden an das Ziel weitergeleitet. Starten Sie dann Ihre Anwendungen erneut, die nun auf das Ziel verweisen.
- Nur CDC (Replizieren nur von Datenänderungen) – Manchmal ist es effizienter, vorhandene Daten mithilfe einer anderen Methode als AWS DMS zu kopieren. Bei einer homogenen Migration kann es z. B. eventuell effizienter sein, beim Laden der Massendaten native Export-/Import-Tools zu verwenden. In diesem Fall können Sie Änderungen ab AWS DMS dem Start des Bulk-Loads replizieren, um Ihre Quell- und Zieldatenbanken synchron zu halten.
- Die Optionen des Vorbereitungsmodus für die Zieltabelle werden nachfolgend aufgeführt. Eine vollständige Erläuterung der Zieltabellenmodi finden Sie unter [Erstellen einer Aufgabe](#).
 - Nichts tun — AWS DMS geht davon aus, dass die Zieltabellen auf dem Ziel vorab erstellt wurden.
 - Tabellen auf dem Ziel löschen — AWS DMS löscht die Zieltabellen und erstellt sie neu.
 - Verkürzen – Wenn Sie Tabellen in der Zieldatenbank erstellt haben, kürzt AWS DMS diese, bevor die Migration gestartet wird. Wenn keine Tabellen vorhanden sind und Sie diese Option auswählen, werden alle fehlenden Tabellen AWS DMS erstellt.
- Die Optionen des LOB-Modus werden nachfolgend aufgeführt. Eine vollständige Erläuterung der LOB-Modi finden Sie unter [Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS](#).
 - Keine LOB-Spalten einschließen – Die LOB-Spalten sind von der Migration ausgeschlossen.
 - Vollständiger LOB-Modus — Migriert komplette LOBs unabhängig von ihrer Größe. AWS DMS migriert LOBs stückweise in Blöcken, die durch den Parameter Max LOB Size gesteuert werden. Dieser Modus ist langsamer als der Modus "Limited LOB".
 - Eingeschränkter LOB-Modus – Die LOBs werden auf den Wert des Parameters Maximale LOB-Größe gekürzt. Dieser Modus ist schneller als der Modus "Full LOB".

- Tabellenzuweisungen – gibt an, welche Tabellen migriert werden und auf welche Weise. Weitere Informationen finden Sie unter [Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen](#).
- Die Datentransformationen werden nachfolgend aufgeführt. Weitere Informationen zu Datentransformationen finden Sie unter [Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON](#).
 - Ändern von Schema-, Tabellen- und Spaltennamen
 - Ändern von Tabellenraumnamen (für Oracle-Zielendpunkte)
 - Definieren von primären Schlüsseln und eindeutigen Indizes auf dem Ziel.
- Datenvalidierung
- CloudWatch Amazon-Protokollierung

Unter Verwendung der Aufgabe migrieren Sie Daten vom Quell- zum Zielendpunkt. Die Aufgabenverarbeitung erfolgt auf der Replikations-Instance. Sie geben an, welche Tabellen und Schemas für die Migration verwendet werden sollen. Zudem legen Sie eine spezielle Verarbeitung, z. B. die Protokollierung von Anforderungen, die Steuerung von Tabellendaten und die Fehlerbehandlung, fest.

Konzeptionell erfüllt eine AWS DMS Replikationsaufgabe zwei unterschiedliche Funktionen, wie in der folgenden Abbildung dargestellt.



Der Volllastprozess ist einfach zu verstehen. Die Daten werden komplett aus der Quelle extrahiert und direkt in das Ziel geladen. Sie können die Anzahl der Tabellen, die parallel extrahiert und geladen werden sollen, auf der AWS DMS Konsole unter Erweiterte Einstellungen angeben.

Weitere Informationen zu AWS DMS Aufgaben finden Sie unter [Arbeiten mit AWS DMS-Aufgaben](#).

Fortlaufende Replikation oder CDC (Change Data Capture)

Sie können eine AWS DMS Aufgabe auch verwenden, um laufende Änderungen am Quelldatenspeicher zu erfassen, während Sie Ihre Daten auf ein Ziel migrieren. Der Prozess zur Erfassung von Änderungen, der bei der Replikation laufender Änderungen von einem Quellendpunkt AWS DMS verwendet wird, sammelt Änderungen an den Datenbankprotokollen mithilfe der systemeigenen API der Datenbank-Engine.

Im CDC-Prozess ist die Replikationsaufgabe darauf ausgelegt, Änderungen von der Quelle zum Ziel zu streamen, wobei In-Memory-Puffer verwendet werden, um Daten während des Transports zu halten. Wenn die In-Memory-Puffer aus irgendeinem Grund erschöpft sind, lässt die Replikationsaufgabe anstehende Änderungen in den Change Cache auf der Festplatte überlaufen. Dies kann beispielsweise der Fall sein, wenn AWS DMS Änderungen an der Quelle schneller erfasst werden, als sie auf das Ziel angewendet werden können. In diesem Fall überschreitet die Ziellatenz der Aufgabe die Quelllatenz der Aufgabe.

Sie können dies überprüfen, indem Sie in der AWS DMS Konsole zu Ihrer Aufgabe navigieren und die Registerkarte Aufgabenüberwachung öffnen. Die CDC LatencyTarget - und LatencySource CDC-Diagramme werden unten auf der Seite angezeigt. Wenn Sie eine Aufgabe haben, die eine Ziellatenz anzeigt, dann ist es wahrscheinlich, dass eine Abstimmung auf dem Ziel-Endpunkt erforderlich ist, um die Anwendungsrate zu erhöhen.

Die Replikationsaufgabe verwendet auch Speicher für Aufgabenprotokolle, wie oben beschrieben. Der für Ihre Replikations-Instance vorkonfigurierte Speicherplatz reicht in der Regel für die Protokollierung und übergelaufene Änderungen aus. Wenn Sie zusätzlichen Speicherplatz benötigen, z. B. wenn Sie bei der Untersuchung eines Migrationsproblems ein detailliertes Debugging verwenden, können Sie die Replikations-Instance modifizieren, um mehr Speicherplatz zuzuordnen.

Quellen für AWS DMS

Sie können verschiedene Quelldatenspeicher in verschiedenen AWS DMS Funktionen verwenden. Die folgenden Abschnitte enthalten die Listen der unterstützten Quelldatenspeicher für jedes AWS DMS Feature.

Themen

- [Quellendpunkte für die Datenmigration](#)
- [Quelldatenbanken für DMS Fleet Advisor](#)
- [Quelldatenanbieter für DMS Schema Conversion](#)
- [Quelldatenanbieter für homogene DMS-Datenmigrationen](#)


Quellendpunkte für die Datenmigration

Sie können die folgenden Datenspeicher als Quellendpunkte für die Datenmigration mit AWS DMS verwenden.

Lokale und EC2-Instance-Datenbanken


- Oracle-Versionen 10.2 und höher (für Versionen 10.x), 11g und bis zu 12.2, 18c und 19c für die Editionen Enterprise, Standard, Standard One und Standard Two
- Microsoft-SQL-Server-Versionen 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019 und 2022.

- Die Enterprise, Standard, Workgroup, Developer und Web Editionen unterstützen die Vollastreplikation.
- Die Enterprise, Standard (Version 2016 und höher) und Developer Editionen unterstützen zusätzlich zur Vollast-Replikation auch CDC-Replikation (fortlaufend).
- Die Express-Edition wird nicht unterstützt.
- MySQL-Versionen 5.5, 5.6, 5.7 und 8.0.

 Note


Support für MySQL 8.0 als Quelle ist in den AWS DMS Versionen 3.4.0 und höher verfügbar, es sei denn, die Transaktions-Payload ist komprimiert. Support für Google Cloud for MySQL 8.0 als Quelle ist in den AWS DMS Versionen 3.4.6 und höher verfügbar.

- MariaDB (unterstützt als MySQL-kompatible Datenquelle), Versionen 10.0 (nur 10.0.24 und höher), 10.2, 10.3, 10.4, 10.5 und 10.6.

 Note


Support für MariaDB als Quelle ist in allen AWS DMS Versionen verfügbar, in denen MySQL unterstützt wird.

- PostgreSQL Version 9.4 und höher (für 9.x-Versionen), 10.x, 11.x, 12.x., 13.x, 14.x und 15.x.

 Note

AWS DMS unterstützt nur PostgreSQL Version 15.x in den Versionen 3.5.1 und höher.

- MongoDB-Versionen 3.x, 4.0, 4.2, 4.4, 5.0 und 6.0

 Note

AWS DMS Versionen 3.5.0 und höher unterstützen keine MongoDB-Versionen vor 3.6.

- SAP Adaptive Server Enterprise (ASE) Versionen 12.5, 15, 15.5, 15.7, 16 und höher
- IBM Db2 für Linux-, UNIX- und Windows (Db2 LUW)-Versionen:
 - Version 9.7, alle Fix Packs
 - Version 10.1, alle Fix Packs

- Version 10.5, alle Fix Packs außer Fix Pack 5
- Version 11.1, alle Fix Packs
- Version 11.5, Mods (0–8) mit nur Fix Pack Zero
- IBM Db2 für z/OS Version 12

Verwaltete Datenbankservices von Drittanbietern:

- Microsoft Azure SQL Database
- Microsoft Azure PostgreSQL Flexible Server Versionen 11.2, 12.15, 13.11, 14.8 und 15.3.
- Microsoft Azure MySQL Flexible Server Versionen 5.7 und 8.
- Google Cloud for MySQL Versionen 5.6, 5.7 und 8.0.
- Google Cloud for PostgreSQL Versionen 9.6, 10, 11, 12, 13, 14 und 15.
- OCI MySQL Heatwave Version 8.0.34.

Amazon-RDS-Instance-Datenbanken und Amazon Simple Storage Service (Amazon S3)

- Oracle-Versionen 11g (Versionen 11.2.0.4 und höher) und bis 12.2, 18c und 19c für die Editionen Enterprise, Standard, Standard One und Standard Two
- Microsoft-SQL-Server-Versionen 2012, 2014, 2016, 2017, 2019 und 2022 für die Editionen Enterprise, Standard, Workgroup und Developer

Note

AWS DMS unterstützt SQL Server Express nicht. Die Web-Edition wird nur für die reine Volllastreplikation unterstützt.

- MySQL-Versionen 5.5, 5.6, 5.7 und 8.0.

Note

Support für MySQL 8.0 als Quelle ist in den AWS DMS Versionen 3.4.0 und höher verfügbar, es sei denn, die Transaktions-Payload ist komprimiert.

- MariaDB (unterstützt als MySQL-kompatible Datenquelle), Versionen 10.0.24 bis 10.0.28, 10.2, 10.3, 10.4, 10.5 und 10.6.

Note

Support für MariaDB als Quelle ist in allen AWS DMS Versionen verfügbar, in denen MySQL unterstützt wird.

- PostgreSQL Version 10.x, 11.x, 12.x, 13.x, 14.x und 15.x.

Note

AWS DMS unterstützt nur PostgreSQL 15.x in den Versionen 3.5.1 und höher.

- Amazon Aurora mit MySQL-Kompatibilität (unterstützt als MySQL-kompatible Datenquelle)
- Amazon Aurora mit PostgreSQL-Kompatibilität (unterstützt als PostgreSQL-kompatible Datenquelle)
- Amazon S3
- Amazon DocumentDB (mit MongoDB-Kompatibilität) Versionen 3.6, 4.0 und 5.0.
- Amazon RDS für IBM Db2 LUW.

Informationen zum Arbeiten mit einer bestimmten Quelle finden Sie unter [Arbeiten mit AWS DMS Endpunkten](#).

Informationen zu den unterstützten Zielendpunkten finden Sie unter [Zielendpunkte für die Datenmigration](#).

Quelldatenbanken für DMS Fleet Advisor

DMS Fleet Advisor unterstützt die folgenden Quelldatenbanken.

- Microsoft SQL Server Version 2012 und bis 2019
- MySQL Version 5.6 und bis 8
- Oracle Version 11g, Release 2 und bis 12c, 19c und 21c
- PostgreSQL Version 9.6 und bis 13

Informationen zum Arbeiten mit einer bestimmten Quelle finden Sie unter [Erstellen von Datenbankbenutzern für AWS DMS Fleet Advisor](#).

Eine Liste der Datenbanken, die DMS Fleet Advisor zur Generierung von Zielempfehlungen verwendet, finden Sie unter [Ziele für DMS Fleet Advisor](#).

Quelldatenanbieter für DMS Schema Conversion

DMS Schema Conversion unterstützt die folgenden Datenanbieter als Quellen für Ihre Migrationsprojekte.

- Microsoft SQL Server Version 2008R2, 2012, 2014, 2016, 2017 und 2019
- Oracle Version 10.2 und höher, 11g und bis 12.2, 18c und 19c sowie Oracle Data Warehouse
- PostgreSQL Version 9.2 und höher
- MySQL Version 5.5 und höher

Ihr Quelldatenanbieter kann eine selbstverwaltete Engine sein, die On-Premises oder auf einer Amazon Elastic Compute Cloud (Amazon EC2)-Instance ausgeführt wird.

Informationen zum Arbeiten mit einer bestimmten Quelle finden Sie unter [Erstellen von Quelldatenanbietern in DMS Schema Conversion](#).

Informationen zu den unterstützten Zieldatenbanken finden Sie unter [Zieldatenanbieter für DMS Schema Conversion](#).

Die AWS Schema Conversion Tool (AWS SCT) unterstützt mehr Quell- und Zieldatenbanken als die DMS-Schemakonvertierung. Informationen zu Datenbanken, die das AWS SCT unterstützen, finden Sie unter [Was ist der AWS Schema Conversion Tool](#).

Quelldatenanbieter für homogene DMS-Datenmigrationen

Sie können die folgenden Datenanbieter als Quelle für homogene Datenmigrationen verwenden.

- MySQL Version 5.7 und höher
- MariaDB 10
- PostgreSQL Version 10.4 bis 14.x.
- MongoDB-Version 4.x, 5.x, 6.0
- Amazon DocumentDB Version 3.6, 4.0, 5.0

Ihr Quelldatenanbieter kann eine selbstverwaltete Engine sein, die On-Premises oder auf einer Amazon-EC2-Instance ausgeführt wird. Sie können auch eine Amazon-RDS-DB-Instance als Quelldatenanbieter verwenden.

Informationen zum Arbeiten mit einer bestimmten Quelle finden Sie unter [Erstellen von Quelldatenanbietern für homogene Datenmigrationen in AWS DMS](#).

Informationen zu den unterstützten Zieldatenbanken finden Sie unter [Zieldatenanbieter für homogene DMS-Datenmigrationen](#).

Ziele für AWS DMS

Sie können verschiedene Zieldatenspeicher in verschiedenen AWS DMS Funktionen verwenden. Die folgenden Abschnitte enthalten die Listen der unterstützten Zieldatenspeicher für jedes AWS DMS Feature.

Themen

- [Zielendpunkte für die Datenmigration](#)
- [Zieldatenbanken für DMS Fleet Advisor](#)
- [Zieldatenanbieter für DMS Schema Conversion](#)
- [Zieldatenanbieter für homogene DMS-Datenmigrationen](#)

Zielendpunkte für die Datenmigration

Sie können die folgenden Datenspeicher als Zielendpunkte für die Datenmigration mit AWS DMS verwenden.


Lokale und Amazon EC2-Instance-Datenbanken

- Oracle-Versionen 10g, 11g, 12c, 18c und 19c für die Editionen Enterprise, Standard, Standard One und Standard Two
- Microsoft SQL Server-Versionen 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019 und 2022 für die Editionen Enterprise, Standard, Workgroup und Developer.

Note


AWS DMS unterstützt die Web- und Express-Editionen von SQL Server nicht.

- MySQL-Versionen 5.5, 5.6, 5.7 und 8.0.
- MariaDB (unterstützt als MySQL-kompatibles Datenziel), Versionen 10.0.24 bis 10.0.28, 10.2, 10.3, 10.4, 10.5 und 10.6.

 Note

Support für MariaDB als Ziel ist in allen AWS DMS Versionen verfügbar, in denen MySQL unterstützt wird.

- PostgreSQL Version 9.4 und höher (für 9.x-Versionen), 10.x, 11.x, 12.x., 13.x, 14.x und 15.x.


 Note

AWS DMS unterstützt nur PostgreSQL 15.x in den Versionen 3.5.1 und höher.

- SAP Adaptive Server Enterprise (ASE) Versionen 15, 15.5, 15.7, 16 und höher
- Redis-Versionen 6.x

Amazon RDS-Instanzdatenbanken, Amazon Redshift, Amazon Redshift Serverless, Amazon DynamoDB, Amazon S3, Amazon OpenSearch Service, Amazon ElastiCache für Redis, Amazon Kinesis Data Streams, Amazon DocumentDB, Amazon Neptune und Apache Kafka

- Oracle-Versionen 11g (Versionen 11.2.0.3.v1 und höher), 12c, 18c und 19c für die Editionen Enterprise, Standard, Standard One und Standard Two
- Microsoft-SQL-Server-Versionen 2012, 2014, 2016, 2017, 2019 und 2022 für die Editionen Enterprise, Standard, Workgroup und Developer

 Note

AWS DMS unterstützt die SQL Server Web- und Express-Editionen nicht.

- MySQL-Versionen 5.5, 5.6, 5.7 und 8.0.
- MariaDB (unterstützt als MySQL-kompatibles Datenziel), Versionen 10.0.24 bis 10.0.28, 10.2, 10.3, 10.4, 10.5 und 10.6.

Note

Support für MariaDB als Ziel ist in allen AWS DMS Versionen verfügbar, in denen MySQL unterstützt wird.

- PostgreSQL Version 10.x, 11.x, 12.x, 13.x, 14.x und 15.x

Note

AWS DMS unterstützt nur PostgreSQL 15.x in den Versionen 3.5.1 und höher.

- IBM Db2 LUW Versionen 11.1 und 11.5
- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon Aurora Serverless v2
- Amazon Redshift
- Amazon Redshift Serverless
- Amazon S3
- Amazon DynamoDB
- OpenSearch Amazon-Dienst
- Amazon ElastiCache für Redis
- Amazon Kinesis Data Streams
- Amazon DocumentDB (mit MongoDB-Kompatibilität)
- Amazon Neptune
- Apache Kafka – [Amazon Managed Streaming für Apache Kafka \(Amazon MSK\)](#) und [selbstverwaltetes Apache Kafka](#)
- Babelfish (Version 3.2.0 und höher) für Aurora PostgreSQL (Versionen 15.3/14.8 und höher)

Informationen zur Arbeit mit einem bestimmten Ziel finden Sie unter [Arbeiten mit AWS DMS Endpunkten](#).

Weitere Informationen zu den unterstützten Endpunkten finden Sie unter [Quellendpunkte für die Datenmigration](#).

Zieldatenbanken für DMS Fleet Advisor

DMS Fleet Advisor generiert Zielempfehlungen unter Verwendung der neuesten Version der folgenden Zieldatenbanken.

- Amazon Aurora MySQL
- Amazon Aurora PostgreSQL
- Amazon RDS für MySQL
- Amazon RDS für Oracle
- Amazon RDS für PostgreSQL
- Amazon RDS für SQL Server

Informationen zu Zielempfehlungen in DMS Fleet Advisor finden Sie unter [Verwenden des AWS DMS-Fleet-Advisor-Features „Zielempfehlungen“](#).

Informationen zu unterstützten Quelldatenbanken finden Sie unter [Quelldatenbanken für DMS Fleet Advisor](#).

Zieldatenanbieter für DMS Schema Conversion

DMS Schema Conversion unterstützt die folgenden Datenanbieter als Ziele für Ihre Migrationsprojekte.

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

Informationen zum Arbeiten mit einem bestimmten Ziel finden Sie unter [Erstellen von Zieldatenanbietern in DMS Schema Conversion](#).

Informationen zu unterstützten Quelldatenbanken finden Sie unter [Quelldatenanbieter für DMS Schema Conversion](#).

Zielfdatenanbieter für homogene DMS-Datenmigrationen

Sie können die folgenden Datenanbieter als Ziele für homogene Datenmigrationen verwenden.

- Amazon Aurora MySQL Version 5.7 und höher
- Amazon Aurora PostgreSQL Version 10.4 bis 14.x
- Amazon Aurora Serverless v2
- Amazon RDS für MySQL Version 5.7 und höher
- Amazon RDS für MariaDB Version 10.2 und höher
- Amazon RDS für PostgreSQL Version 10.4 bis 14.x
- Amazon DocumentDB Version 4.0, 5.0 und DocumentDB Elastic Cluster

Informationen zum Arbeiten mit einem bestimmten Ziel finden Sie unter [Erstellung von Zielfdatenanbietern für homogene Datenmigrationen in AWS DMS](#).

Informationen zu unterstützten Quelldatenbanken finden Sie unter [Quelldatenanbieter für homogene DMS-Datenmigrationen](#).

Erstellung eines Amazon-Ressourcennamens (ARN) für AWS DMS

Wenn Sie die AWS DMS API AWS CLI oder verwenden, um Ihre Datenbankmigration zu automatisieren, arbeiten Sie mit Amazon Resource Name (ARNs). Jede in Amazon Web Services erstellte Ressource wird durch einen ARN identifiziert, bei dem es sich um eine eindeutige Kennung handelt. Wenn Sie die AWS DMS API AWS CLI oder verwenden, um Ihre Datenbankmigration einzurichten, geben Sie den ARN der Ressource an, mit der Sie arbeiten möchten.

Ein ARN für eine AWS DMS Ressource verwendet die folgende Syntax:

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

In dieser Syntax gilt Folgendes:

- *region* ist die ID des Ortes AWS-Region, an dem die AWS DMS Ressource erstellt wurde, z. us-west-2 B.

Die folgende Tabelle zeigt AWS-Region Namen und Werte, die Sie beim Erstellen eines ARN verwenden sollten.

Region	Name
Region Asien-Pazifik (Tokio)	ap-northeast-1
Region Asien-Pazifik (Seoul)	ap-northeast-2
Region Asien-Pazifik (Mumbai)	ap-south-1
Region Asien-Pazifik (Singapur)	ap-southeast-1
Region Asien-Pazifik (Sydney)	ap-southeast-2
Region Kanada (Zentral)	ca-central-1
Region China (Peking)	cn-north-1
Region „China (Ningxia)“	cn-northwest-1
Region Europa (Stockholm)	eu-north-1
Region Europa (Mailand)	eu-south-1
Region EU (Frankfurt)	eu-central-1
Region Europa (Irland)	eu-west-1
Region EU (London)	eu-west-2
Region Europa (Paris)	eu-west-3
Region Südamerika (São Paulo)	sa-east-1
Region USA Ost (Nord-Virginia)	us-east-1
Region USA Ost (Ohio)	us-east-2
Region US West (N. California)	us-west-1
Region USA West (Oregon)	us-west-2

- *account number* ist Ihre Kontonummer (ohne Bindestriche). Um Ihre Kontonummer zu finden, melden Sie sich unter <http://aws.amazon.com> bei Ihrem AWS Konto an, wählen Sie Mein Konto/ Konsole und dann Mein Konto aus.
- *resourcetype* ist der Ressourcentyp. AWS DMS

Die folgende Tabelle zeigt die Ressourcentypen, die beim Erstellen eines ARN für eine bestimmte AWS DMS Ressource verwendet werden müssen.

AWS DMS Ressourcentyp	ARN-Format
Replikations-Instance	<code>arn:aws:dms: <i>region</i>: <i>account</i>:rep: <i>resourcename</i></code>
Endpoint	<code>arn:aws:dms: <i>region</i>:<i>account</i>:endpoint: <i>resourcename</i></code>
Replikationsaufgabe	<code>arn:aws:dms: <i>region</i>:<i>account</i>:task:<i>resourcename</i></code>
Subnetzgruppe	<code>arn:aws:dms: <i>region</i>:<i>account</i>:subgrp:<i>resourcename</i></code>

- *resourcename* ist der der Ressource zugewiesene AWS DMS Ressourcenname. Dies ist eine beliebige generierte Zeichenfolge.

Die folgende Tabelle zeigt Beispiele für ARNs für AWS DMS Ressourcen. Hier gehen wir von einem AWS -Konto 123456789012 aus, das in der Region USA Ost (Nord-Virginia) erstellt wurde und über einen Ressourcenamen verfügt.

Ressourcentyp	Beispiel-ARN
Replikations-Instance	<code>arn:aws:dms:us-east-1:123456789012:rep:QLXQZ64MH7CXF4QCMGRVYVXAI</code>
Endpoint	<code>arn:aws:dms:us-east-1:123456789012:endpoint:D3HMZ2IGUCGFF3NTAXUXGF6S5A</code>
Replikationsaufgabe	<code>arn:aws:dms:us-east-1:123456789012:task:2PVRE MWNPGYJCVU2IBPTOYTIV4</code>

Ressourcentyp	Beispiel-ARN
Subnetzgruppe	<code>arn:aws:dms:us-east-1:123456789012:subgrp:test-tag-grp</code>

Verwendung AWS DMS mit anderen Diensten AWS

Sie können es AWS DMS mit mehreren anderen AWS Diensten verwenden:

- Sie können eine Amazon EC2 Instance oder eine Amazon RDS DB-Instance als Ziel für eine Datenmigration verwenden.
- Sie können das AWS Schema Conversion Tool (AWS SCT) verwenden, um Ihr Quellschema und Ihren SQL-Code in ein äquivalentes Zielschema und SQL-Code zu konvertieren.
- Sie können Amazon S3 als Speicherort für Ihre Daten oder als Zwischenschritt bei der Migration großer Datenmengen verwenden.
- Sie können AWS CloudFormation damit Ihre AWS Ressourcen für die Infrastrukturverwaltung oder -bereitstellung einrichten. Sie können beispielsweise AWS DMS Ressourcen wie Replikationsinstanzen, Aufgaben, Zertifikate und Endpunkte bereitstellen. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt und diese Ressourcen für Sie AWS CloudFormation bereitstellt und konfiguriert.

AWS DMS Unterstützung für AWS CloudFormation

Sie können AWS DMS Ressourcen bereitstellen mit AWS CloudFormation. AWS CloudFormation ist ein Service, der Sie bei der Modellierung und Einrichtung Ihrer AWS Ressourcen für die Infrastrukturverwaltung oder -bereitstellung unterstützt. Sie können beispielsweise AWS DMS Ressourcen wie Replikationsinstanzen, Aufgaben, Zertifikate und Endpunkte bereitstellen. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt und diese Ressourcen für Sie AWS CloudFormation bereitstellt und konfiguriert.

Als Entwickler oder Systemadministrator können Sie Sammlungen dieser Ressourcen erstellen und verwalten, die Sie dann für wiederkehrende Migrationsaufgaben oder die Bereitstellung von Ressourcen für Ihre Organisation verwenden können. Weitere Informationen zu AWS CloudFormation finden Sie unter [AWS CloudFormation Konzepte](#) im AWS CloudFormation Benutzerhandbuch.

AWS DMS unterstützt die Erstellung der folgenden AWS DMS Ressourcen mithilfe von AWS CloudFormation:

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

Erste Schritte mit AWS Database Migration Service

Im folgenden Tutorial erfahren Sie, wie Sie eine Datenbankmigration mit AWS Database Migration Service (AWS DMS) durchführen.

Gehen Sie folgendermaßen vor, um eine Datenbankmigration durchzuführen:

1. Richten Sie Ihr AWS-Konto mit den Schritten in [Einrichten für AWS Database Migration Service](#) ein.
2. Erstellen Sie Ihre Beispieldatenbanken und einen Amazon-EC2-Client, um Ihre Quelldatenbank zu füllen und die Replikation zu testen. Erstellen Sie außerdem eine Virtual Private Cloud (VPC) basierend auf dem Amazon Virtual Private Cloud (Amazon VPC)-Service, um Ihre Tutorial-Ressourcen zu speichern. Um diese Ressourcen zu erstellen, folgen Sie den Schritten unter [Voraussetzungen für AWS Database Migration Service](#).
3. Füllen Sie Ihre Quelldatenbank mit einem [Beispielskript zur Datenbankerstellung](#) auf.
4. Verwenden Sie DMS Schema Conversion oder AWS Schema Conversion Tool (AWS SCT), um das Schema von der Quelldatenbank in die Zieldatenbank zu konvertieren. Um DMS Schema Conversion zu verwenden, folgen Sie den Schritten unter [Erste Schritte mit DMS Schema Conversion](#). Um das Schema mit AWS SCT zu konvertieren, folgen Sie den Schritten unter [Schema migrieren](#).
5. Erstellen Sie eine Replikations-Instance, die alle Migrationsprozesse durchführt. Führen Sie die Schritte unter [Replikation](#) aus, um diese und die folgenden Aufgaben auszuführen.
6. Geben Sie die Quell- und Zielendpunkte an. Weitere Informationen zum Erstellen von Endpunkten finden Sie unter [Erstellen der Quell- und Zielendpunkte](#).
7. Erstellen Sie eine Aufgabe, um festzulegen, welche Tabellen und Replikationsprozesse verwendet werden sollen, und starten Sie die Replikation. Informationen zum Erstellen von Datenbankmigrationsaufgaben finden Sie unter [Erstellen einer Aufgabe](#).
8. Stellen Sie sicher, dass die Replikation funktioniert, indem Sie Abfragen in der Zieldatenbank ausführen.

Einrichten für AWS Database Migration Service

So melden Sie sich für ein AWS-Konto an

Wenn Sie kein AWS-Konto haben, führen Sie die folgenden Schritte zum Erstellen durch.

Anmeldung für ein AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für ein AWS-Konto anmelden, wird ein Root-Benutzer des AWS-Kontos erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem [Administratorbenutzer Administratorzugriff](#) zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff](#) erfordern.

AWS sendet Ihnen eine Bestätigungs-E-Mail, sobald die Anmeldung abgeschlossen ist. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Einen Administratorbenutzer erstellen

Wenn Sie sich für AWS-Konto registriert haben, sichern Sie Root-Benutzer des AWS-Kontos, aktivieren Sie AWS IAM Identity Center erstellen Sie einen Administratorbenutzer, damit Sie nicht den Root-Benutzer für alltägliche Aufgaben verwenden.

Schützen Ihres Root-Benutzer des AWS-Kontos

1. Melden Sie sich bei der [AWS Management Console](#) als Kontobesitzer an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-AnmeldungBenutzerhandbuch zu .

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen dazu finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für den Root-Benutzer Ihres AWS-Konto \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen eines Administratorbenutzers

1. Aktivieren von IAM Identity Center

Eine genaue Anleitung finden Sie unter [Aktivierung von AWS IAM Identity Center](#) im AWS IAM Identity Center-Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter [Konfigurieren des Benutzerzugriffs mit der Standard-IAM-Identity-Center-Verzeichnis](#) im AWS IAM Identity Center-Benutzerhandbuch.

Als Administratorbenutzer anmelden

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM-Identity-Center-Benutzer finden Sie unter [Anmelden beim AWS-Zugangsportale](#) im AWS-Anmeldung Benutzerhandbuch zu.

Voraussetzungen für AWS Database Migration Service

In diesem Abschnitt erfahren Sie, welche Aufgaben für AWS DMS erforderlich sind, z. B. die Einrichtung Ihrer Quell- und Zieldatenbanken. Im Rahmen dieser Aufgaben richten Sie auch eine Virtual Private Cloud (VPC) basierend auf dem Amazon-VPC-Service für Ihre Ressourcen ein. Darüber hinaus richten Sie eine Amazon-EC2-Instance ein, die Sie verwenden, um Ihre Quelldatenbank zu füllen und die Replikation in Ihrer Zieldatenbank zu überprüfen.

Note

Das Auffüllen der Quelldatenbank dauert bis zu 45 Minuten.

Für dieses Tutorial erstellen Sie eine MariaDB-Datenbank als Quelle und eine PostgreSQL-Datenbank als Ziel. In diesem Szenario werden häufig verwendete, kostengünstige Datenbank-Engines verwendet, um die Replikation zu demonstrieren. Die Verwendung verschiedener

Datenbank-Engines demonstriert AWS DMS-Features für die Migration von Daten zwischen heterogenen Plattformen.

Die Ressourcen in diesem Tutorial verwenden die Region USA West (Oregon). Wenn Sie eine andere AWS-Region verwenden möchten, geben Sie stattdessen die von Ihnen gewählte Region an, unabhängig davon, wo USA West (Oregon) angezeigt wird.

Note

Der Einfachheit halber verwenden die Datenbanken, die Sie für dieses Tutorial erstellen, keine Verschlüsselung oder andere erweiterte Sicherheits-Features. Sie müssen Sicherheits-Features verwenden, um die Sicherheit Ihrer Produktionsdatenbanken zu gewährleisten. Weitere Informationen finden Sie unter [Sicherheit in Amazon RDS](#).

Die erforderlichen Schritte finden Sie unter den folgenden Themen:

Themen

- [Erstellen einer VPC](#)
- [Erstellen von Amazon-RDS-Parametergruppen](#)
- [Erstellen Ihrer Amazon-RDS-Quelldatenbank](#)
- [Erstellen Ihrer Amazon-RDS-Zieldatenbank](#)
- [Erstellen eines Amazon-EC2-Clients](#)
- [Füllen Ihrer Quelldatenbank](#)

Erstellen einer VPC

In diesem Abschnitt erstellen Sie eine VPC, die Ihre AWS-Ressourcen enthält. Die Verwendung einer VPC ist eine bewährte Methode bei der Verwendung von AWS-Ressourcen, so dass Ihre Datenbanken, Amazon-EC2-Instances, Sicherheitsgruppen usw. logisch organisiert und sicher sind.

Wenn Sie eine VPC für Ihre Tutorial-Ressourcen verwenden, wird auch sichergestellt, dass Sie alle Ressourcen löschen, die Sie verwenden, wenn Sie mit dem Tutorial fertig sind. Sie müssen alle Ressourcen löschen, die eine VPC enthält, bevor Sie die VPC löschen können.

So erstellen Sie eine VPC für die Verwendung mit AWS DMS

1. Melden Sie sich an der AWS Management Console an und öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im Navigationsbereich die Option VPC-Dashboards und wählen Sie dann VPC erstellen.
3. Geben Sie auf der Seite VPC erstellen die folgenden Optionen ein:
 - Zu erstellende Ressourcen: VPC und mehr
 - Auto-Generierung von Namens-Tags: Wählen Sie Automatisch generieren und geben Sie **DMSVPC** ein.
 - IPv4-Block: **10.0.1.0/24**
 - IPv6-CIDR-Block: Kein IPv6-CIDR-Block
 - Tenancy: Standard
 - Anzahl der Availability Zones: 2
 - Anzahl der öffentlichen Subnetze: 2
 - Anzahl der privaten Subnetze: 2
 - NAT-Gateways (\$): Keine
 - VPC-Endpunkte: Keine

Wählen Sie VPC erstellen aus.

4. Wählen Sie im Navigationsbereich Ihre VPCs. Notieren Sie die VPC-ID für DMSVPC.
5. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
6. Wählen Sie die Gruppe mit dem Namen default aus mit einer VPC-ID, die der ID entspricht, die Sie für DMSVPC notiert haben.
7. Wählen Sie die Registerkarte Eingehende Regeln und anschließend Eingehende Regeln bearbeiten aus.
8. Wählen Sie Add rule. Fügen Sie eine Regel des Typs MySQL/Aurora hinzu und wählen Sie Anywhere-IPv4 als Quelle.
9. Wählen Sie erneut Regel hinzufügen. Fügen Sie eine Regel des Typs PostgreSQL hinzu und wählen Sie Anywhere-IPv4 als Quelle.
10. Wählen Sie Save rules (Regeln speichern) aus.

Erstellen von Amazon-RDS-Parametergruppen

Verwenden Sie Amazon-RDS-Parametergruppen, um Einstellungen für Ihre Quell- und Zieldatenbanken für AWS DMS festzulegen. Um die anfängliche und die fortlaufende Replikation zwischen Ihren Datenbanken zu ermöglichen, stellen Sie sicher, dass Sie Folgendes konfigurieren:

- Das Binärprotokoll Ihrer Quelldatenbank, so dass AWS DMS ermitteln kann, welche inkrementellen Aktualisierungen repliziert werden müssen.
- Die Replikationsrolle Ihrer Zieldatenbank, so dass Fremdschlüsseinschränkungen bei der ersten Datenübertragung von AWS DMS ignoriert werden. Mit dieser Einstellung kann AWS DMS Daten in einer anderen Reihenfolge migrieren.

So erstellen Sie Parametergruppen für die Verwendung mit AWS DMS

1. Öffnen Sie die Amazon-RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
2. Wählen Sie im Navigationsbereich Parametergruppen aus.
3. Wählen Sie auf der Seite Parametergruppen die Option Parametergruppe erstellen aus.
4. Geben Sie auf der Seite Parametergruppe erstellen die folgenden Parameter ein:
 - Parametergruppenfamilie: mariadb10.6
 - Gruppenname: **dms-mariadb-parameters**
 - Description (Beschreibung: **Group for specifying binary log settings for replication**)

Wählen Sie Erstellen.

5. Wählen Sie auf der Seite Parametergruppen dms-mariadb-parameters und auf der dms-mariadb-parameters Seite Bearbeiten aus.
6. Legen Sie die folgenden Parameter auf die folgenden Werte fest:
 - binlog_checksum: KEINE
 - binlog_format: ZEILE

Wählen Sie Änderungen speichern aus.

7. Wählen Sie auf der Seite Parametergruppen erneut die Option Parametergruppe erstellen aus.
8. Geben Sie auf der Seite Parametergruppe erstellen die folgenden Parameter ein:

- Parametergruppenfamilie: postgres13
- Gruppenname: **dms-postgresql-parameters**
- Description (Beschreibung: **Group for specifying role setting for replication**)

Wählen Sie Erstellen.

9. Wählen Sie auf der Seite Parametergruppen die Option `dms-postgresql-parameters`.
10. Wählen Sie auf der `dms-postgresql-parameters` Seite Bearbeiten und legen Sie den Parameter `session_replication_role` auf Replikat fest. Beachten Sie, dass sich der Parameter `session_replication_role` nicht auf der ersten Parameterseite befindet. Verwenden Sie die Paginierungssteuerelemente oder das Suchfeld, um den Parameter zu finden.
11. Wählen Sie Änderungen speichern aus.

Erstellen Ihrer Amazon-RDS-Quelldatenbank

Gehen Sie wie folgt vor, um Ihre Amazon-RDS-Quelldatenbank zu erstellen.

So erstellen Sie Ihre Amazon-RDS-für-MariaDB-Quelldatenbank

1. Öffnen Sie die Amazon-RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
2. Wählen Sie auf der Seite Dashboard die Option Datenbank erstellen im Abschnitt Datenbank aus. Wählen Sie nicht die Option Datenbank erstellen im Abschnitt Testen der neuen Amazon-RDS-Multi-AZ-Bereitstellungsoption für MySQL und PostgreSQL oben auf der Seite.
3. Stellen Sie auf der Seite Datenbank erstellen die folgenden Optionen ein:
 - Wählen Sie unter Auswahl eines Datenbank-Erstellungsverfahrens: Standard-Erstellung aus.
 - Engine-Optionen: Wählen Sie als Engine-Typ MariaDB aus. Lassen Sie für Version MariaDB 10.6.14 ausgewählt.
 - Vorlagen: Wählen Sie Dev/Test aus.
 - Einstellungen:
 - DB-Instance-Kennung: Geben Sie **dms-mariadb** ein.
 - Geben Sie im Abschnitt Einstellungen für Anmeldeinformationen Folgendes ein:
 - Master-Benutzername: Lassen Sie **admin** stehen.
 - Lassen Sie die Option Masteranmeldedaten in AWS Secrets Manager deaktiviert.

- Passwort automatisch generieren: Wählen Sie diese Option nicht aus.
- Master-Passwort: Geben Sie **changeit** ein.
- Passwort bestätigen: Geben Sie **changeit** erneut ein.
- Instance-Konfiguration:
 - DB-Instance-Klasse: Lassen Sie Standardklassen ausgewählt.
 - Wählen Sie als DB-Instance-Klasse db.m5.large aus.
- Speicherung:
 - Deaktivieren Sie das Feld Automatische Speicherskalierung aktivieren.
 - Behalten Sie die restlichen Einstellungen bei.
- Verfügbarkeit und Dauerhaftigkeit: Lassen Sie die Option Keine Standby-Instance erstellen ausgewählt.
- Connectivity (Konnektivität):
 - Computing-Ressource Lassen Sie Keine Verbindung zu einer EC2-Computing-Ressource unverändert
 - Netzwerktyp: Lassen Sie IPv4 ausgewählt.
 - Virtual Private Cloud: DMSVPC-vpc
 - Öffentlicher Zugang: Ja. Sie müssen den öffentlichen Zugriff aktivieren, um AWS Schema Conversion Tool verwenden zu können.
 - Availability Zone: us-west-2a
 - Behalten Sie die restlichen Einstellungen bei.
- Datenbankauthentifizierung: Lassen Sie die Option Passwortauthentifizierung ausgewählt.
- Deaktivieren Sie unter Überwachung das Feld Performance Insights aktivieren. Erweitern Sie den Abschnitt Zusätzliche Konfiguration und deaktivieren Sie das Kontrollkästchen Erweiterte Überwachung aktivieren.
- Erweitern Sie Zusätzliche Konfiguration.
 - Geben Sie auf der Seite Datenbankoptionen in das Feld Anfänglicher Datenbankname **dms_sample** ein.
 - Wählen Sie unter DB-Parametergruppe ausdms-mariadb-parameters.
 - Lassen Sie für Optionsgruppe die Option default:mariadb-10-6 ausgewählt.
 - Gehen Sie unter Backup wie folgt vor:

- Lassen Sie die Option Automatische Backups aktiviert ausgewählt. In Ihrer Quelldatenbank müssen automatische Backups aktiviert sein, um die fortlaufende Replikation zu unterstützen.
 - Wählen Sie unter Aufbewahrungszeitraum für Backups den Wert 1 Tag.
 - Lassen Sie für Backup-Fenster die Option Keine Präferenz ausgewählt.
 - Löschen Sie das Feld Tags in Schnappschüsse kopieren.
 - Lassen Sie das Kontrollkästchen Replikation in einer anderen AWS-Region aktivieren unmarkiert.
 - Deaktivieren Sie unter Verschlüsselung das Kontrollkästchen Verschlüsselung aktivieren.
 - Lassen Sie den Abschnitt Protokollexporte unverändert.
 - Deaktivieren Sie unter Wartung das Kontrollkästchen Automatische Nebenversions-Upgrades aktivieren und belassen Sie die Einstellung für das Wartungsfenster bei Keine Präferenz.
 - Lassen Sie Löschschutz aktivieren ohne Markierung.
4. Wählen Sie Datenbank erstellen aus.

Erstellen Ihrer Amazon-RDS-Zioldatenbank

Wiederholen Sie das vorherige Verfahren mit den folgenden Änderungen, um Ihre Amazon-RDS-Zioldatenbank zu erstellen.

So erstellen Sie Ihre RDS-für-PostgreSQL-Zioldatenbank

1. Wiederholen Sie die Schritte 1 und 2 des vorigen Verfahrens.
2. Legen Sie auf der Seite Datenbank erstellen dieselben Optionen fest, mit Ausnahme der folgenden:
 - a. Wählen Sie unter Engine-Optionen die Option PostgreSQL.
 - b. Wählen Sie für Version PostgreSQL 13.7-R1
 - c. Geben Sie als DB-Instance-ID **dms-postgresql** ein.
 - d. Lassen Sie für den Master-Benutzernamen die Option **postgres** ausgewählt.
 - e. Wählen Sie für DB-Parametergruppe dms-postgresql-parameters.
 - f. Löschen Sie Automatische Backups aktivieren.
3. Wählen Sie Datenbank erstellen aus.

Erstellen eines Amazon-EC2-Clients

In diesem Abschnitt erstellen Sie einen Amazon-EC2-Client. Sie verwenden diesen Client, um Ihre Quelldatenbank mit zu replizierenden Daten zu füllen. Sie verwenden diesen Client auch, um die Replikation zu überprüfen, indem Sie Abfragen in der Zieldatenbank ausführen.

Die Verwendung eines Amazon-EC2-Clients für den Zugriff auf Ihre Datenbanken bietet die folgenden Vorteile gegenüber dem Zugriff auf Ihre Datenbanken über das Internet:

- Sie können den Zugriff auf Ihre Datenbanken auf Clients beschränken, die sich in derselben VPC befinden.
- Wir haben geprüft, ob die Tools, die Sie in diesem Tutorial verwenden, auf Amazon Linux 2023 funktionieren und einfach zu installieren sind, was wir für dieses Tutorial empfehlen.
- Datenoperationen zwischen Komponenten in einer VPC funktionieren im Allgemeinen besser als Datenoperationen über das Internet.

So erstellen und konfigurieren Sie einen Amazon-EC2-Client zum Auffüllen Ihrer Quelldatenbank

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Dashboard die Option Instance starten.
3. Geben Sie auf der Seite Eine Instance starten die folgenden Werte ein:
 - a. Geben Sie im Abschnitt Name und Tags **DMSClient** für Name ein.
 - b. Behalten Sie die Einstellungen unter Anwendungs- und BS-Images (Amazon Machine Image) unverändert bei.
 - c. Wählen Sie im Abschnitt Instance-Typ die Option t2.xlarge aus.
 - d. Wählen Sie unter Schlüsselpaar (Anmeldung) die Option Neues Schlüsselpaar erstellen.
 - e. Geben Sie auf der Seite Schlüsselgruppe erstellen folgendes ein:
 - Schlüsselpaarname: **DMSKeyPair**
 - Schlüsselpaarartyp: Lassen Sie RSA unverändert.
 - Dateiformat für private Schlüssel: Wählen Sie pem für OpenSSH unter MacOS oder Linux oder ppk für PuTTY unter Windows.

Speichern Sie die Schlüsseldatei, wenn Sie dazu aufgefordert werden.

Note

Sie können auch ein vorhandenes Amazon-EC2-Schlüsselpaar verwenden, anstatt ein neues zu erstellen.

- f. Wählen Sie im Abschnitt Netzwerkeinstellungen die Option Bearbeiten aus. Wählen Sie die folgenden Einstellungen aus:
- VPC - erforderlich: Wählen Sie die VPC mit der ID aus, die Sie für die DMSVPC-vpc-VPC aufgezeichnet haben.
 - Subnetz: Wählen Sie das erste öffentliche Subnetz.
 - Öffentliche IP-Adresse automatisch zuweisen: Wählen Sie Aktivieren.

Behalten Sie die übrigen Einstellungen bei und wählen Sie Instance starten.

Füllen Ihrer Quelldatenbank

In diesem Abschnitt finden Sie Endpunkte für Ihre Quell- und Zieldatenbanken zur späteren Verwendung und verwenden die folgenden Tools, um die Quelldatenbank zu füllen:

- Git, um das Skript herunterzuladen, das Ihre Quelldatenbank füllt.
- MariaDB-Client, um dieses Skript auszuführen.

Endpunkte finden

Suchen und notieren Sie sich die Endpunkte Ihrer DB-Instances von RDS für MariaDB und RDS für PostgreSQL zur späteren Verwendung.

So ermitteln Sie Ihre DB-Instance-Endpunkte

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die Datenbank dms-mariadb aus und notieren Sie sich den Endpunkt-Wert für die Datenbank.
4. Wiederholen Sie die vorherigen Schritte für die dms-postgresql-Datenbank.

Füllen Ihrer Quelldatenbank

Stellen Sie als Nächstes eine Verbindung zu Ihrer Client-Instance her, installieren Sie die erforderliche Software, laden Sie AWS-Beispieldatenbankskripts von Git herunter und führen Sie die Skripts aus, um Ihre Quelldatenbank zu füllen.

So füllen Sie Ihre Quelldatenbank

1. Stellen Sie mithilfe des Host-Namens und des öffentlichen Schlüssels, die Sie in den vorherigen Schritten gespeichert haben, eine Verbindung zur Client-Instance her.

Weitere Informationen zum Herstellen einer Verbindung mit einer Amazon-EC2-Instance finden Sie unter [Zugriff auf Instances](#) im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

Note

Wenn Sie PuTTY verwenden, aktivieren Sie TCP-Keepalives auf der Seite mit den Verbindungseinstellungen, damit Ihre Verbindung nicht aufgrund von Inaktivität unterbrochen wird.

2. Installieren Sie Git, MariaDB und PostgreSQL. Bestätigen Sie die Installation nach Bedarf.

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

3. Führen Sie den folgenden Befehl aus, um die Datenbankerstellungsskripts von herunterzuladen GitHub.

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

4. Wechseln Sie in das `aws-database-migration-samples/mysql/sampledb/v1/`-Verzeichnis.
5. Führen Sie den folgenden Befehl aus. Geben Sie beispielsweise den Endpunkt für Ihre Quell-RDS-Instance an, den Sie zuvor notiert haben, zum Beispiel `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. Lassen Sie das Skript zur Datenbankerstellung ausführen. Das Skript benötigt bis zu 45 Minuten, um das Schema zu erstellen und die Daten aufzufüllen. Sie können Fehler und Warnungen, die das Skript anzeigt, ignorieren.

Migrieren Sie Ihr Quellschema in Ihre Zieldatenbank mit AWS SCT

In diesem Abschnitt verwenden Sie AWS Schema Conversion Tool, um Ihr Quellschema in Ihre Zieldatenbank zu migrieren. Alternativ können Sie DMS Schema Conversion verwenden, um Ihre Quelldatenbankschemata zu konvertieren. Weitere Informationen finden Sie unter [Erste Schritte mit DMS Schema Conversion](#).

So migrieren Sie Ihr Quellschema mit AWS SCT zu Ihrer Zieldatenbank

1. Installieren Sie AWS Schema Conversion Tool. Weitere Informationen finden Sie unter [Installieren, Prüfen und Aktualisieren von AWS SCT](#) im Benutzerhandbuch zum AWS Schema Conversion Tool.

Wenn Sie JDBC-Treiber für MySQL und PostgreSQL herunterladen, notieren Sie sich, wo Sie die Treiber speichern, falls das Tool Sie nach ihren Speicherorten fragt.

2. Öffnen Sie die AWS Schema Conversion Tool. Wählen Sie Datei und dann Neues Projekt aus.
3. Stellen Sie im Fenster Neues Projekt die folgenden Werte ein:
 - Setzen Sie Projektname auf **DMSProject**.
 - Behalten Sie den Speicherort unverändert bei, um Ihr AWS SCT-Projekt im Standardordner zu speichern.

Wählen Sie OK.

4. Wählen Sie Quelle hinzufügen, um Ihrem Projekt eine MySQL-Quelldatenbank hinzuzufügen, wählen Sie dann MySQL und schließlich Weiter.
5. Stellen Sie auf der Seite Quelle hinzufügen die folgenden Werte ein:
 - Name der Verbindung: **source**
 - Servername: Geben Sie den Endpunkt für die MySQL-Datenbank ein, den Sie zuvor notiert haben.
 - Server-Port: **3306**
 - Benutzername: **admin**

- Passwort: **changeit**
6. Wählen Sie Ziel hinzufügen, um Ihrem Projekt eine Amazon-RDS-für-PostgreSQL-Zioldatenbank hinzuzufügen, und wählen Sie dann Amazon RDS für PostgreSQL. Wählen Sie Next (Weiter).
 7. Stellen Sie auf der Seite Ziel hinzufügen die folgenden Werte ein:
 - Name der Verbindung: **target**
 - Servername: Geben Sie den Endpunkt für die PostgreSQL-Datenbank ein, den Sie zuvor notiert haben.
 - Server-Port: **5432**
 - Datenbank: Geben Sie den Namen Ihrer PostgreSQL-Datenbank ein.
 - Benutzername: **postgres**
 - Passwort: **changeit**
 8. Wählen Sie im linken Bereich dms_sample unter Schemata aus. Wählen Sie im rechten Bereich Ihre Amazon-RDS-für-PostgreSQL-Zioldatenbank aus. Wählen Sie Zuordnung erstellen. Sie können mehrere Zuordnungsregeln zu einem einzigen AWS SCT-Projekt hinzufügen. Weitere Informationen zu Zuordnungsregeln finden Sie unter [Erstellen von Zuordnungsregeln](#).
 9. Wählen Sie Hauptansicht.
 10. Wählen Sie im linken Bereich dms_sample unter Schemata aus. Öffnen Sie das Kontextmenü (rechte Maustaste) und wählen Sie Schema konvertieren aus. Bestätigen Sie die Aktion.

Nachdem das Tool das Schema konvertiert hat, wird das Schema dms_sample im rechten Bereich angezeigt.
 11. Öffnen Sie im rechten Bereich unter Schemata das Kontextmenü (Rechtsklick) für dms_sample und wählen Sie Auf Datenbank anwenden aus. Bestätigen Sie die Aktion.

Prüfen Sie, ob die Schemamigration abgeschlossen ist. Führen Sie die folgenden Schritte aus.

So prüfen Sie Ihre Schemamigration

1. Stellen Sie eine Verbindung zu Ihrem Amazon-EC2-Client her.
2. Verwenden Sie den folgenden Befehl, um den PSQL-Client zu starten. Geben Sie Ihren PostgreSQL-Datenbank-Endpunkt an und geben Sie das Datenbankpasswort ein, wenn Sie dazu aufgefordert werden.

```
psql \
```

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

3. Fragen Sie eine der (leeren) Tabellen ab, um zu überprüfen, ob AWS SCT das Schema korrekt angewendet hat.

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

Einrichten der Replikation für AWS Database Migration Service

In diesem Thema richten Sie die Replikation zwischen den Quell- und Zieldatenbanken ein.

Schritt 1: Erstellen einer Replikations-Instance über die AWS DMS-Konsole

Erstellen Sie zum Beginn mit AWS DMS eine Replikations-Instance.

Eine Replikations-Instance führt die tatsächliche Migration von Daten zwischen Quell- und Zielendpunkten durch. Ihre Instance benötigt ausreichend Speicher und Rechenleistung, um die Aufgaben auszuführen, mit denen Daten aus Ihrer Quelldatenbank in Ihre Zieldatenbank migriert werden. Wie groß diese Replikations-Instance sein sollte, hängt von der Menge der zu migrierenden Daten und den Aufgaben ab, die Ihre Instance ausführen muss. Weitere Informationen zu Replikations-Instances finden Sie unter [Mit einer AWS DMS Replikationsinstanz arbeiten](#).

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

So erstellen Sie eine Replikations-Instance über die Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Replikations-Instances und dann Replikations-Instance erstellen aus.
3. Geben Sie auf der Seite Replikations-Instance erstellen die Konfiguration Ihrer Replikations-Instance an:
 - a. Geben Sie unter Name **DMS-instance** ein.

- b. Geben Sie unter Beschreibung eine kurze Beschreibung für Ihre Replikations-Instance ein (optional).
- c. Lassen Sie für Instance-Klasse `dms.t3.medium` ausgewählt.

Die Instance benötigt ausreichend Speicher-, Netzwerk- und Rechenleistung für Ihre Migration. Weitere Informationen zur Auswahl einer Instance-Klasse finden Sie unter [Auswahl der richtigen AWS DMS-Replikationsinstanz für Ihre Migration](#).

- d. Behalten Sie für Engine-Version den Standardwert bei.
- e. Wählen Sie für Multi AZ Dev- oder Test-Workload (Single-AZ).
- f. Akzeptieren Sie für Allied Storage (GiB) den Standardwert von 50 GiB.

In AWS DMS wird der Speicher hauptsächlich für Protokolldateien und zwischengespeicherte Transaktionen verwendet. Für zwischengespeicherte Transaktionen wird Speicherplatz nur verwendet, wenn die zwischengespeicherten Transaktionen auf Festplatte geschrieben werden müssen. Daher verbraucht AWS DMS nicht viel Speicherplatz.

- g. Wählen Sie für Netzwerktyp IPv4.
 - h. Wählen Sie für VPC DMSVPC.
 - i. Behalten Sie für Replikationssubnetzgruppe die aktuell gewählte Replikationssubnetzgruppe bei.
 - j. Löschen Sie Öffentlich zugänglich.
4. Wählen Sie die Registerkarte Erweiterte Sicherheits- und Netzwerkkonfiguration aus, um Werte für Netzwerk- und Verschlüsselungseinstellungen festzulegen, wenn Sie diese benötigen:
- a. Wählen Sie für Availability Zone `us-west-2a`.
 - b. Wählen Sie für VPC-Sicherheitsgruppe(n) die Standard-Sicherheitsgruppe aus, falls sie nicht bereits ausgewählt ist.
 - c. Lassen Sie für AWS KMS key (Standard) `aws/dms` ausgewählt.
5. Lassen Sie die Einstellungen auf der Registerkarte Wartung unverändert. Der Standardwert ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitraum für jede AWS-Region an einem zufälligen Wochentag ausgewählt wird.
6. Wählen Sie Erstellen aus.

AWS DMS erstellt eine Replikations-Instance, um Ihre Migration durchzuführen.

Schritt 2: Angeben von Quell- und Zielendpunkten

Während Ihre Replikations-Instance erstellt wird, können Sie die Endpunkte der Quell- und Zieldatenbank für die bereits erstellten Amazon-RDS-Datenbanken angeben. Sie erstellen jeden Endpunkt separat.

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.

Descriptive Amazon Resource Name (ARN) - optional
A descriptive Amazon Resource Name (ARN) for the endpoint. This is optional.

So geben Sie die Endpunkte der Quell- und Zieldatenbank mit der AWS DMS-Konsole an

1. Wählen Sie im Navigationsbereich Endpunkte und anschließend Endpunkt erstellen aus.
2. Wählen Sie auf der Seite Endpunkt erstellen den Endpunkttyp der Quelle aus. Wählen Sie das Feld RDS-DB-Instance auswählen und wählen Sie die dms-mariadb-Instance aus.
3. Geben Sie im Abschnitt Endpunkt-Konfiguration **dms-mysql-source** für Endpunkt-Kennung ein.
4. Lassen Sie für Quell-Engine MySQL ausgewählt.

5. Wählen Sie für Zugriff auf Endpunktdatenbank die Option Zugriffsinformationen manuell bereitstellen aus. Prüfen Sie, ob Port, Secure-Socket-Layer (SSL)-Modus, Benutzername und Passwort korrekt sind.
6. Wählen Sie die Registerkarte Testen der Endpunktverbindung (optional). Wählen Sie für VPC DMSVPC.
7. Lassen Sie für Replikations-Instance dms-instance ausgewählt.
8. Wählen Sie Test ausführen aus.

Nachdem Sie Test ausführen ausgewählt haben, erstellt AWS DMS den Endpunkt mit den von Ihnen angegebenen Details und stellt eine Verbindung zu ihm her. Wenn die Verbindung fehlschlägt, bearbeiten Sie die Endpunktdefinition und testen Sie die Verbindung erneut. Sie können den Endpunkt auch manuell löschen.

9. Wählen Sie nach einem erfolgreichen Test Endpunkt erstellen aus.
10. Geben Sie anhand der AWS DMS-Konsole die Endpunkte der Zieldatenbank an. Wiederholen Sie dazu die vorherigen Schritte mit den folgenden Einstellungen:
 - Endpunktyp: Zielendpunkt
 - RDS-Instance: dms-postgresql
 - Endpunkt-Kennung: **dms-postgresql-target**
 - Ziel-Engine: **PostgreSQL** ausgewählt lassen.

Wenn Sie alle Informationen für Ihre Endpunkte bereitgestellt haben, erstellt AWS DMS Ihre Quell- und Zielendpunkte zur Verwendung bei der Datenbankmigration.

Schritt 3: Erstellen einer Aufgabe und Migrieren der Daten

In diesem Schritt erstellen Sie eine Aufgabe, um Daten zwischen den von Ihnen erstellten Datenbanken zu migrieren.

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](#)

So erstellen Sie eine Migrationsaufgabe und starten Ihre Datenbankmigration

1. Wählen Sie im Navigationsbereich der Konsole Datenbankmigrationsaufgaben und dann Aufgabe erstellen aus. Die Seite Datenbankmigrationsaufgabe erstellen wird geöffnet.
2. Geben Sie im Abschnitt Aufgabenkonfiguration die folgenden Aufgabenoptionen an:
 - Aufgaben-ID: Geben Sie **dms-task** ein.
 - Replikations-Instance: Wählen Sie Ihre Replikations-Instance (dms-instance-vpc-**<vpc id>**).
 - Quelldatenbank-Endpoint: Wählen Sie dms-mysql-source.
 - Zieldatenbank-Endpoint: Wählen Sie dms-postgresql-target.

- Migrationstyp: Wählen Sie Vorhandene Daten migrieren und fortlaufende Änderungen replizieren.
3. Wählen Sie die Registerkarte Aufgabeneinstellungen. Verwenden Sie die folgenden Einstellungen:
 - Zieltabellen-Vorbereitungsmodus: Keine Aktion
 - Aufgabe nach Abschluss der Volllast anhalten: Nicht anhalten
 4. Wählen Sie die Registerkarte Tabellenzuordnungen aus und erweitern Sie Auswahlregeln. Wählen Sie Neue Auswahlregel hinzufügen. Verwenden Sie die folgenden Einstellungen:
 - Schema: Geben Sie ein Schema ein
 - Schemaname: **dms_sample**
 5. Wählen Sie die Registerkarte Startup-Konfiguration der Migrationsaufgabe und dann Automatisch bei Erstellung aus.
 6. Wählen Sie Create task aus.

AWS DMS erstellt dann die Migrationsaufgabe und startet sie. Die anfängliche Datenbankreplikation dauert ca. 10 Minuten. Stellen Sie sicher, dass Sie den nächsten Schritt im Tutorial ausführen, bevor AWS DMS mit der Migration der Daten fertig ist.

Schritt 4: Testen der Replikation

In diesem Abschnitt fügen Sie während und nach der ersten Replikation Daten in die Quelldatenbank ein und fragen die Zieldatenbank nach den eingefügten Daten ab.

So testen Sie die Replikation

1. Stellen Sie sicher, dass Ihre Datenbankmigrationsaufgabe den Status Wird ausgeführt anzeigt, Ihre erste Datenbankreplikation, die im vorherigen Schritt gestartet wurde, jedoch noch nicht abgeschlossen ist.
2. Verbinden Sie sich mit Ihrem Amazon-EC2-Client und starten Sie den MySQL-Client mit dem folgenden Befehl. Geben Sie Ihren MySQL-Datenbank-Endpunkt an.

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

3. Führen Sie den folgenden Befehl aus, um einen Datensatz in die Quelldatenbank einzufügen.

```
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. Beenden Sie den MySQL-Client.

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

5. Bevor die Replikation abgeschlossen ist, fragen Sie die Zieldatenbank nach dem neuen Datensatz ab.

Stellen Sie von der Amazon-EC2-Instance aus mithilfe des folgenden Befehls eine Verbindung zur Zieldatenbank her und geben Sie dabei Ihren Zieldatenbank-Endpoint an.

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

Geben Sie das Passwort (**changeit**) ein, wenn Sie dazu aufgefordert werden.

6. Bevor die Replikation abgeschlossen ist, fragen Sie die Zieldatenbank nach dem neuen Datensatz ab.

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

7. Während Ihre Migrationsaufgabe ausgeführt wird, können Sie den Fortschritt Ihrer Datenbankmigration während der Ausführung überwachen:

- Wählen Sie im Navigationsbereich der DMS-Konsole die Option Datenbankmigrationsaufgaben aus.
- Wählen Sie dms-task.
- Wählen Sie Tabellenstatistiken.

Weitere Informationen zur Überwachung finden Sie unter [Überwachen von AWS-DMS-Aufgaben](#).

8. Fragen Sie nach Abschluss der Replikation die Zieldatenbank erneut nach dem neuen Datensatz ab. AWS DMS migriert den neuen Datensatz nach Abschluss der ersten Replikation.

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

9. Beenden Sie den psql-Client.

```
dms_sample=> quit
```

10. Wiederholen Sie Schritt 1, um erneut eine Verbindung zur Quelldatenbank herzustellen.

11. Fügen Sie einen weiteren Datensatz in die person-Tabelle ein.

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

12. Wiederholen Sie die Schritte 3 und 4, um die Verbindung mit der Quelldatenbank zu trennen und sich mit der Zieldatenbank zu verbinden.

13. Fragen Sie die Zieldatenbank erneut nach den replizierten Daten ab.

```
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)
```

Schritt 5: Bereinigen von AWS DMS-Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, empfehlen wir, dass Sie die erstellten Ressourcen löschen. Sie können den Befehl AWS verwenden, um sie zu entfernen. Stellen Sie sicher, dass Sie die Migrationsaufgaben löschen, bevor Sie die Replikations-Instance und die Endpunkte löschen.

So löschen Sie eine Migrationsaufgabe mithilfe der Konsole

1. Wählen Sie im Navigationsbereich der AWS DMS-Konsole Datenbankmigrationsaufgaben aus.
2. Wählen Sie dms-task.
3. Wählen Sie Actions (Aktionen), Delete (Löschen) aus.

Löschen einer Replikations-Instance unter Verwendung der -Konsole

1. Wählen Sie im Navigationsbereich der AWS DMS-Konsole die Option Replikations-Instances aus.
2. Wählen Sie DMS-Instance.
3. Wählen Sie Actions (Aktionen), Delete (Löschen) aus.

AWS DMS löscht die Replikations-Instance und entfernt sie von der Seite Replikations-Instances.

So entfernen Sie Endpunkte mithilfe der Konsole

1. Wählen Sie im Navigationsbereich der AWS DMS-Konsole die Option Endpunkte.
2. Wählen Sie dms-mysql-source.
3. Wählen Sie Actions (Aktionen), Delete (Löschen) aus.

Nachdem Sie Ihre AWS DMS-Ressourcen gelöscht haben, stellen Sie sicher, dass Sie auch die folgenden Ressourcen löschen. Hilfe zum Löschen von Ressourcen in anderen Services finden Sie in der Dokumentation der einzelnen Services.

- Ihre RDS-Datenbanken.
- Ihre RDS-Datenbank-Parametergruppen.
- Ihre RDS-Subnetzgruppen.
- Alle Amazon-CloudWatch-Protokolle, die zusammen mit Ihren Datenbanken und Ihrer Replikations-Instance erstellt wurden.
- Sicherheitsgruppen, die für Ihren Amazon-VPC- und Amazon-EC2-Client erstellt wurden. Stellen Sie sicher, dass Sie die Regel für eingehende Nachrichten aus der Standardeinstellung für die Sicherheitsgruppen launch-wizard-1 entfernen, da dies erforderlich ist, damit Sie sie löschen können.
- Ihr Amazon-EC2-Client.

- Ihre Amazon-VPC.
- Ihr Amazon-EC2-Schlüsselpaar für Ihren Amazon-EC2-Client.

Weitere Ressourcen für die Arbeit mit AWS Database Migration Service

Später in diesem Leitfaden erfahren Sie, wie Sie mit AWS DMS Ihre Daten zu und von den meisten gängigen kommerziellen und Open-Source-Datenbanken migrieren können.

Wir empfehlen Ihnen außerdem, sich bei der Vorbereitung und Durchführung eines Datenbankmigrationsprojekts die folgenden Ressourcen anzusehen:

- [Schrittweiser Migrationsleitfaden für AWS DMS](#) – Dieser Leitfaden enthält schrittweise Anleitungen, die den Prozess der Datenmigration zu AWS beschreiben.
- [AWS DMS-API-Referenz](#) – Diese Referenz beschreibt sämtliche API-Vorgänge für AWS Database Migration Service im Detail.
- [AWS CLI für AWS DMS](#) – Diese Referenz enthält Informationen zur Verwendung von AWS Command Line Interface (AWS CLI) mit AWS DMS.

Ermitteln und Evaluieren von Datenbanken für die Migration mit AWS DMS Fleet Advisor

Sie können DMS Fleet Advisor verwenden, um Metadaten und Leistungskennzahlen aus mehreren Datenbankumgebungen zu erfassen. Diese erfassten Metriken bieten Einblick in Ihre Dateninfrastruktur. [DMS Fleet Advisor](#) erfasst Metadaten und Metriken von Ihren On-Premises-Datenbank- und Analyseservern von einem oder mehreren zentralen Standorten aus, ohne dass es auf jedem Computer installiert werden muss. Derzeit unterstützt DMS Fleet Advisor die Erfassung von Erkennungs- und Metriken für Microsoft SQL Server-, MySQL-, Oracle- und PostgreSQL-Datenbankserver.

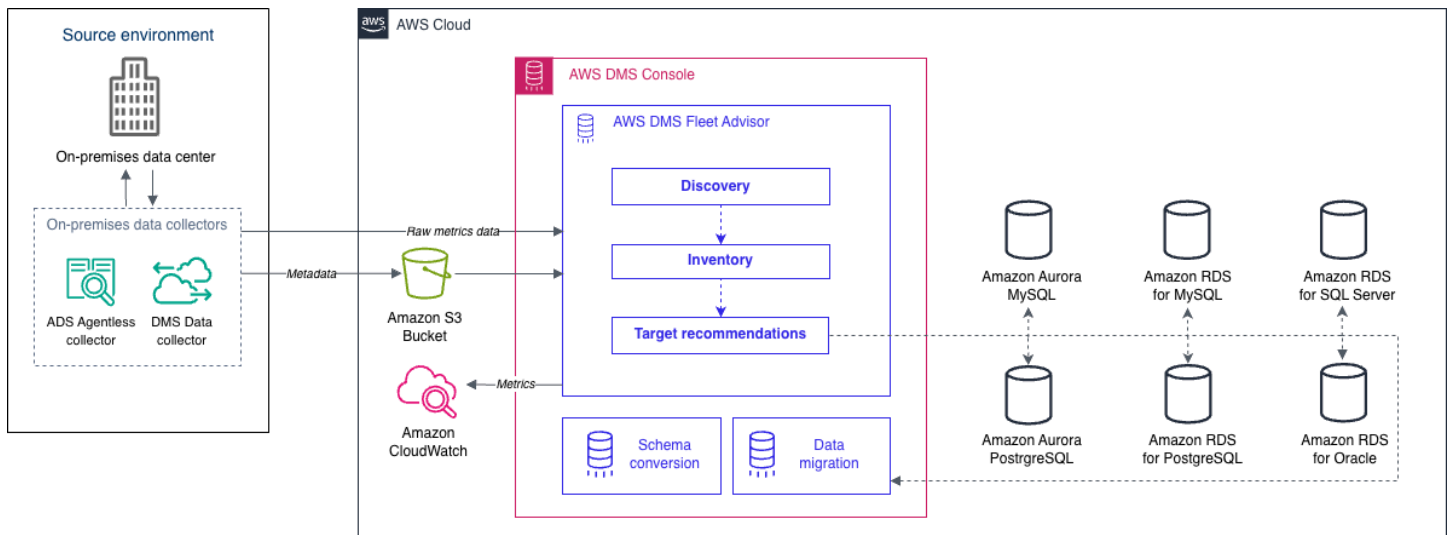
Auf der Grundlage der im Netzwerk gefundenen Daten können Sie ein Inventar erstellen, um die Liste der Datenbankserver für die weitere Datenerfassung zu definieren. Nachdem AWS DMS Informationen über Ihre Server, Datenbanken und Schemata gesammelt hat, können Sie die Durchführbarkeit Ihrer geplanten Datenbankmigrationen analysieren.

Für Datenbanken in Ihrem Inventar, die Sie zu AWS Cloud migrieren möchten, generiert DMS Fleet Advisor geeignete Zielempfehlungen. Bei der Generierung von Zielempfehlungen berücksichtigt DMS Fleet Advisor die Metriken Ihres Datenkollektors und die bevorzugten Einstellungen. Nachdem DMS Fleet Advisor Empfehlungen generiert hat, können Sie detaillierte Informationen für jede Zieldatenbankkonfiguration einsehen. Die Datenbanktechniker und Administratoren Ihres Unternehmens können DMS Fleet Advisor Target Recommendations verwenden, um die Migration ihrer On-Premises-Datenbanken zu AWS zu planen. Sie können verschiedene verfügbare Migrationsoptionen erkunden und diese Empfehlungen in die exportierenAWS Pricing Calculator, um die Kosten weiter zu optimieren.

Eine Liste der unterstützten Quelldatenbanken finden Sie unter [Quellen für DMS Fleet Advisor](#).

Eine Liste der Datenbanken, die DMS Fleet Advisor zur Generierung von Zielempfehlungen verwendet, finden Sie unter [Ziele für DMS Fleet Advisor](#). DMS Fleet Advisor generiert wie Empfehlungen, z. B. von der Oracle-Quell- zur Oracle-Zieldatenbank. DMS Fleet Advisor generiert auch heterogene Empfehlungen, z. B. die Migration von Oracle oder Microsoft SQL Server zur Zieldatenbank von RDS für PostgreSQL oder Aurora PostgreSQL.

Das folgende Diagramm veranschaulicht den Prozess der AWS DMS Fleet Advisor Target Recommendations.



Gehen Sie die folgenden Themen durch, um AWS DMS Fleet Advisor besser zu verstehen.

Themen

- [Unterstützte AWS-Regionen](#)
- [Erste Schritte mit DMS Fleet Advisor](#)
- [Einrichten von AWS DMS Fleet Advisor](#)
- [Ermitteln von Datenbanken für die Migration mithilfe von Datenkollektoren](#)
- [Verwenden von Inventaren für Analysen in AWS DMS Fleet Advisor](#)
- [Verwenden des AWS DMS-Fleet-Advisor-Features „Zielempfehlungen“](#)
- [Einschränkungen für DMS Fleet Advisor](#)

Unterstützte AWS-Regionen

Sie können DMS Fleet Advisor in den folgenden AWS-Regionen verwenden.

Name der Region	Region
USA Ost (Nord-Virginia)	us-east-1
USA Ost (Ohio)	us-east-2
USA West (Nordkalifornien)	us-west-1

Name der Region	Region
US West (Oregon)	us-west-2
Asien-Pazifik (Hongkong)	ap-east-1
Asien-Pazifik (Tokio)	ap-northeast-1
Asien-Pazifik (Seoul)	ap-northeast-2
Asien-Pazifik (Osaka)	ap-northeast-3
Asien-Pazifik (Mumbai)	ap-south-1
Asien-Pazifik (Singapur)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asien-Pazifik (Jakarta)	ap-southeast-3
Kanada (Zentral)	ca-central-1
Europa (Frankfurt)	eu-central-1
Europa (Stockholm)	eu-north-1
Europa (Irland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Milan)	eu-south-3
Kanada (Zentral)	ca-central-1
South America (São Paulo)	sa-east-1
Naher Osten (Bahrain)	me-south-1
Afrika (Kapstadt)	af-south-1

Erste Schritte mit DMS Fleet Advisor

Sie können DMS Fleet Advisor verwenden, um Ihre On-Premises-Quelldatenbanken für die Migration zu AWS Cloud zu ermitteln. Anschließend können Sie für jede Ihrer On-Premises-Datenbanken das korrekte Migrationsziel in der AWS Cloud bestimmen. Verwenden Sie den folgenden Workflow, um eine Bestandsaufnahme Ihrer Quelldatenbanken zu erstellen und Zielempfehlungen zu generieren.

1. Erstellen Sie einen Amazon-S3-Bucket sowie IAM-Richtlinien, -Rollen und Benutzer. Weitere Informationen finden Sie unter [Erstellen der erforderlichen Ressourcen](#).
2. Erstellen Sie Datenbankbenutzer mit den Mindestberechtigungen, die für den DMS-Datenkollektor erforderlich sind. Weitere Informationen finden Sie unter [Erstellen von Datenbankbenutzern](#).
3. Erstellen Sie einen Datenkollektor und laden Sie ihn herunter. Weitere Informationen finden Sie unter [Erstellen eines Datenkollektors](#).
4. Installieren Sie den Datenkollektor in Ihrer lokalen Umgebung. Als Nächstes konfigurieren Sie Ihren Datenkollektor so, dass er die erfassten Daten an DMS Fleet Advisor senden kann. Weitere Informationen finden Sie unter [Installieren eines Datenkollektors](#).
5. Ermitteln Sie die Betriebssystem- und Datenbankserver in Ihrer Datenumgebung. Weitere Informationen finden Sie unter [Ermitteln von Betriebssystem- und Datenbankservern](#).
6. Erfassen Sie Datenbank-Metadaten und Kennzahlen zur Ressourcennutzung. Weitere Informationen finden Sie unter [Erfassen von Daten](#).
7. Analysieren Sie Ihre Quelldatenbanken und Schemata. DMS Fleet Advisor führt eine umfangreiche Analyse Ihrer Datenbanken durch, um ähnliche Schemata zu identifizieren. Weitere Informationen finden Sie unter [Verwenden von Inventaren für Analysen in AWS DMS Fleet Advisor](#).
8. Generieren, betrachten und speichern Sie eine lokale Kopie der Zielempfehlungen für Ihre Quelldatenbanken. Weitere Informationen finden Sie unter [Zielempfehlungen](#).

Nachdem Sie das Migrationsziel für jede Quelldatenbank festgelegt haben, können Sie DMS Schema Conversion verwenden, um Ihre Datenbankschemata zu einer neuen Plattform zu konvertieren. Anschließend können Sie die Daten mit AWS DMS migrieren. Weitere Informationen finden Sie unter [Konvertieren von Datenbankschemata mithilfe von DMS Schema Conversion](#) und [Was ist AWS Database Migration Service?](#)

[Dieses Video](#) stellt die Benutzeroberfläche von DMS Schema Conversion vor und hilft Ihnen, sich mit den Kernkomponenten dieses Services vertraut zu machen.

Einrichten von AWS DMS Fleet Advisor

Führen Sie die folgenden erforderlichen Aufgaben aus, um AWS DMS Fleet Advisor einzurichten.

Themen

- [Erstellen der erforderlichen AWS-Ressourcen für AWS DMS Fleet Advisor](#)
- [Erstellen von Datenbankbenutzern für AWS DMS Fleet Advisor](#)

Erstellen der erforderlichen AWS-Ressourcen für AWS DMS Fleet Advisor

DMS Fleet Advisor benötigt eine Reihe von AWS-Ressourcen in Ihrem Konto, um Inventarinformationen weiterzuleiten und zu importieren und den Status des DMS-Datenkollektors zu aktualisieren.

Bevor Sie zum ersten Mal Daten erfassen und Inventare von Datenbanken und Schemata erstellen können, müssen Sie die folgenden Voraussetzungen erfüllen.

Um Ihren Amazon-S3-Bucket und Ihre IAM-Ressourcen zu konfigurieren, führen Sie einen der folgenden Schritte durch:

- [Konfigurieren von Amazon-S3- und IAM-Ressourcen mit AWS CloudFormation](#) (empfohlen).
- [Konfigurieren von Amazon-S3- und IAM-Ressourcen in der AWS Management Console](#)

Konfigurieren von Amazon-S3- und IAM-Ressourcen mit AWS CloudFormation

Bei einem CloudFormation Stack handelt es sich um eine Sammlung von AWS-Ressourcen, die Sie als einzelne Einheit verwalten können. Um das Erstellen der erforderlichen Ressourcen für DMS Fleet Advisor zu vereinfachen, können Sie die AWS CloudFormation Vorlagendateien verwenden, um CloudFormation Stacks zu erstellen. Weitere Informationen finden Sie unter [Erstellen eines Stacks in der AWS CloudFormation-Konsole](#) im Benutzerhandbuch zu AWS CloudFormation.

Note

Dieser Abschnitt gilt nur für die Verwendung des eigenständigen DMS-Fleet-Advisor-Kollektors. Informationen zur Verwendung eines einzigen On-Premises-Kollektors zum Erfassen von Informationen über Datenbanken und Server finden Sie unter

[Application Discovery Service Agentless Collector](#) im [AWS Application Discovery Service-Benutzerhandbuch](#).

Amazon S3- und IAM-Ressourcen, die von erstellt wurden CloudFormation

Wenn Sie die CloudFormation Vorlagen verwenden, erstellen sie Stacks, die die folgenden Ressourcen in Ihrem enthaltenAWS-Konto:

- Ein Amazon-S3-Bucket mit dem Namen `dms-fleetadvisor-data-accountId-region`
- Ein IAM-Benutzer mit dem Namen `FleetAdvisorCollectorUser-region`
- Eine IAM-Servicerolle mit dem Namen `FleetAdvisorS3Role-region`
- Eine Zugriffsrichtlinie mit dem Namen `FleetAdvisorS3Role-region-Policy`
- Eine Zugriffsrichtlinie mit dem Namen `FleetAdvisorCollectorUser-region-Policy`
- Eine IAM Service Linked Role (SLR) mit dem Namen `AWSServiceRoleForDMSFleetAdvisor`

Führen Sie die unten aufgeführten Schritte aus, um Ihre Ressourcen mit zu konfigurieren CloudFormation.

- [Schritt 1: Herunterladen der CloudFormation Vorlagendateien](#)
- [Schritt 2: Konfigurieren von Amazon S3 und IAM mit CloudFormation](#)

Schritt 1: Herunterladen der CloudFormation Vorlagendateien

Eine CloudFormation Vorlage ist eine Deklaration der AWS Ressourcen, aus denen ein Stack besteht. Die Vorlage wird als JSON-Datei gespeichert.

So laden Sie die CloudFormation Vorlagendateien herunter

1. Öffnen Sie das Kontextmenü (rechte Maustaste) für den bereitgestellten Link und wählen Sie Link speichern unter aus.
 - Wenn Sie DMS Fleet Advisor verwenden möchten, wählen Sie [dms-fleetadvisor-iam-slr-s3.zip](#) aus. Wenn Sie die SLR für DMS Fleet Advisor bereits erstellt haben, wählen Sie [dms-fleetadvisor-iam-s3.zip](#)
 - Wenn Sie planen, den AWS Application Discovery Service (ADS) Agentless Collector zu verwenden und die SLR dafür nicht erstellt haben, wählen Sie [dms-fleetadvisor-ads-iam-slr-](#)

[s3.zip aus](#). Wenn Sie die SLR für DMS Fleet Advisor bereits mit ADS erstellt haben, wählen Sie [dms-fleetadvisor-ads-iam-s3.zip aus](#).

2. Speichern Sie die Datei auf Ihrem Computer.

Schritt 2: Konfigurieren von Amazon S3 und IAM mit CloudFormation

Wenn Sie die CloudFormation Vorlage für IAM verwenden, werden die zuvor aufgeführten Amazon S3- und IAM-Ressourcen erstellt.

So konfigurieren Sie Amazon S3 und IAM mit CloudFormation

1. Öffnen Sie die - CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
2. Starten Sie den Assistenten zum Erstellen von Stacks, indem Sie in der Dropdown-Liste Stack erstellen und Mit neuen Ressourcen auswählen.
3. Gehen Sie auf der Seite Create stack (Stack erstellen) wie folgt vor:
 - a. Wählen Sie unter Prepare template (Vorlage vorbereiten) den Wert Template is ready (Vorlage ist bereit) aus.
 - b. Wählen Sie unter Template source (Vorlagenquelle) den Wert Upload a template file (Vorlagendatei hochladen) aus.
 - c. Navigieren Sie für Datei auswählen zu und wählen Sie dann `dms-fleetadvisor-iam-slr-S3.json`, `dms-fleetadvisor-iam-S3.json`, `dms-fleetadvisor-ads-iam-slr-s3.zip` oder `dms-fleetadvisor-ads-iam-s3.zip`.
 - d. Wählen Sie Weiter aus.
4. Führen Sie auf der Seite Specify DB Details (Festlegen von DB-Detail) die folgenden Schritte aus:
 - a. Geben Sie unter Stack-Name **`dms-fleetadvisor-iam-slr-s3`**, **`dms-fleetadvisor-iam-s3`**, **`dms-fleetadvisor-ads-iam-slr-s3`** oder **`dms-fleetadvisor-ads-iam-s3`** ein.
 - b. Wählen Sie Weiter aus.
5. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
6. Gehen Sie auf der Seite `dms-fleetadvisor-iam-slr-s3`, `dms-fleetadvisor-iam-s3`, `dms-fleetadvisor-ads-iam-slr-s3` oder `dms-fleetadvisor-ads-iam-s3` überprüfen wie folgt vor:

- a. Aktivieren Sie das Kontrollkästchen Ich bestätige, dass AWS CloudFormation ggf. IAM-Ressourcen mit benutzerdefinierten Namen erstellt.
- b. Wählen Sie Absenden aus.

CloudFormation erstellt den S3-Bucket und die IAM-Rollen und den Benutzer, die DMS Fleet Advisor benötigt. Wenn im linken Bereich dms-fleetadvisor-iam-slr-s3, dms-fleetadvisor-iam-s3, dms-fleetadvisor-ads-iam-slr-s3 oder dms-fleetadvisor-ads-iam-s3 CREATE_COMPLETE angezeigt, fahren Sie mit dem nächsten Schritt fort.

7. Wählen Sie im linken Bereich dms-fleetadvisor-iam-slr-s3, dms-fleetadvisor-iam-s3, dms-fleetadvisor-ads-iam-slr-s3 oder dms-fleetadvisor-ads-iam-s3 aus. Führen Sie im rechten Bereich die folgenden Schritte aus:
 - a. Wählen Sie Stack-Info aus. Ihr Stack hat eine ID im Format `arn:aws:cloudformation:region :account-no :stack/dms-fleetadvisor-iam-slr-s3/identifizier` , `arn:aws:cloudformation:region :account-no :stack/dms-fleetadvisor-iam-s3/identifizier` , `arn:aws:cloudformation:region :account-no :stack/dms-fleetadvisor-ads-iam-slr-s3/identifizier` oder `arn:aws:cloudformation:region :account-no :stack/dms-fleetadvisor-ads-iam-s3/identifizier` .
 - b. Wählen Sie Resources aus. Sie sollten Folgendes sehen:
 - Ein Amazon-S3-Bucket mit dem Namen `dms-fleetadvisor-data-accountId-region`
 - Eine Servicerolle mit dem Namen `FleetAdvisorS3Role-region`
 - Ein IAM-Benutzer mit dem Namen `FleetAdvisorCollectorUser-region`
 - Eine IAM-SLR mit dem Namen `AWSServiceRoleForDMSFleetAdvisor` (falls Sie `dms-fleet-advisor-iam-slr-s3.zip` oder `dms-fleet-advisor-ads-iam-slr-s3.zip` heruntergeladen haben).
 - Eine Zugriffsrichtlinie mit dem Namen `FleetAdvisorS3Role-region-Policy`
 - Eine Zugriffsrichtlinie mit dem Namen `FleetAdvisorCollectorUser-region-Policy`

Konfigurieren von Amazon-S3- und IAM-Ressourcen in der AWS Management Console

Erstellen eines Amazon-S3-Buckets

Erstellen Sie einen Amazon-S3-Bucket, in dem Inventarmetadaten gespeichert werden können. Wir empfehlen, diesen S3-Bucket vorab zu konfigurieren, bevor Sie DMS Fleet Advisor verwenden. AWS DMS speichert Ihre DMS-Fleet-Advisor-Inventarmetadaten in diesem S3-Bucket.

Weitere Informationen zum Erstellen eines Amazon-S3-Buckets finden Sie unter [Erstellen Ihres ersten S3-Buckets](#) im Amazon-S3-Benutzerhandbuch.

Note

DMS Fleet Advisor unterstützt nur SSE-S3-verschlüsselte Buckets.

So erstellen Sie einen Amazon-S3-Bucket zum Speichern von lokalen Datenumgebungsinformationen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Create Bucket (Bucket erstellen).
3. Geben Sie auf der Seite Bucket erstellen einen global eindeutigen Namen ein, der Ihren Anmeldenamen für den Bucket enthält, z. B. fa-bucket-*yoursignin*.
4. Wählen Sie die AWS-Region aus, in der Sie den DMS Fleet Advisor verwenden.
5. Behalten Sie die restlichen Einstellungen bei und wählen Sie Bucket erstellen.

Erstellen von IAM-Ressourcen

In diesem Abschnitt erstellen Sie IAM-Ressourcen für Ihren Datenkollektor, Ihren IAM-Benutzer und DMS Fleet Advisor.

Themen

- [Erstellen von IAM-Ressourcen für Ihren DMS-Datenkollektor](#)
- [Erstellen der serviceverknüpften DMS-Fleet-Advisor-Rolle](#)

Erstellen von IAM-Ressourcen für Ihren DMS-Datenkollektor

Um sicherzustellen, dass Ihr Datenkollektor ordnungsgemäß funktioniert und die erfassten Metadaten in Ihren Amazon-S3-Bucket hochlädt, erstellen Sie die folgenden Richtlinien. Erstellen Sie dann einen IAM-Benutzer mit den folgenden Mindestberechtigungen. Weitere Informationen zu, DMS-Datenkollektor finden Sie unter [Ermitteln von Datenbanken für die Migration mithilfe von Datenkollektoren](#).

So erstellen Sie eine IAM-Richtlinie für DMS Fleet Advisor und Ihren Datenkollektor zum Zugriff auf Amazon S3

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte JSON aus.
5. Fügen Sie den folgenden JSON-Code in den Editor ein und ersetzen Sie dabei den Beispielcode. Ersetzen Sie *fa_bucket* durch den Namen des Amazon-S3-Buckets, den Sie im vorherigen Abschnitt erstellt haben.

```
{
  "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. Wählen Sie Weiter: Tags und Weiter: Prüfen aus.
7. Geben Sie unter Name* **FleetAdvisorS3Policy** ein und wählen Sie dann Richtlinie erstellen aus.

So erstellen Sie eine IAM-Richtlinie für den DMS-Datenkollektor zum Zugriff auf DMS Fleet Advisor

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte JSON aus.
5. Fügen Sie den folgenden JSON-Code in den Editor ein und ersetzen Sie dabei den Beispielcode.

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

6. Wählen Sie Weiter: Tags und Weiter: Prüfen aus.
7. Geben Sie unter **DMSCollectorPolicyName** den Namen ein und wählen Sie dann Richtlinie erstellen aus.

So erstellen Sie einen IAM-Benutzer mit Mindestberechtigungen zur Verwendung des DMS-Datenkollektors

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Klicken Sie im Navigationsbereich auf Users (Benutzer).

3. Wählen Sie Add Users (Benutzer hinzufügen).
4. Geben Sie auf der Seite Benutzer hinzufügen unter **FleetAdvisorCollectorUser** als Benutzername* ein. Wählen Sie Zugriffsschlüssel – Programmgesteuerter Zugriff unter AWS-Zugriffstyp auswählen. Wählen Sie Weiter: Berechtigungen aus.
5. Wählen Sie im Abschnitt Berechtigungen einstellen die Option Vorhandene Richtlinien direkt anfügen.
6. Verwenden Sie das Suchsteuerelement, um die DMSCollectorPolicy- und FleetAdvisorS3Policy-Richtlinien zu finden und auszuwählen, die Sie zuvor erstellt haben. Wählen Sie Weiter: Markierungen.
7. Wählen Sie auf der Seite Markierungen die Option Weiter: Prüfen aus.
8. Wählen Sie auf der Seite Review die Option Create user aus. Wählen Sie auf der nächsten Seite .csv herunterladen, um die neuen Benutzeranmeldedaten zu speichern. Verwenden Sie diese Anmeldeinformationen mit DMS Fleet Advisor für die erforderlichen Mindestzugriffsberechtigungen.

So erstellen Sie eine IAM-Rolle für DMS Fleet Advisor und Ihren Datenkollektor zum Zugriff auf Amazon S3

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Roles.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie auf der Seite Vertrauenswürdige Entität auswählen unter Typ der vertrauenswürdigen Entität die Option AWS-Service aus. Unter Anwendungsfälle für andere AWS-Services wählen Sie DMS.
5. Markieren Sie das Kontrollkästchen DMS und wählen Sie Weiter.
6. Wählen Sie auf der Seite Berechtigungen hinzufügen die Option FleetAdvisorS3Policy aus. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Benennen, Prüfen und Erstellen **FleetAdvisorS3Role** für Rollename ein und wählen Sie dann Rolle erstellen.
8. Geben Sie auf der Seite Rollen **FleetAdvisorS3Role** für Rollename ein. Wählen Sie FleetAdvisorS3Role aus.
9. Wählen Sie auf der Seite FleetAdvisorS3Role die Registerkarte Vertrauensstellungen aus. Wählen Sie Vertrauensrichtlinie bearbeiten aus.

10. Fügen Sie auf der Seite Vertrauensrichtlinie bearbeiten den folgenden JSON-Code in den Editor ein und ersetzen Sie den vorhandenen Text.

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

Die vorstehende Richtlinie erteilt den Services, die AWS DMS zum Import erfasster Daten aus dem Amazon-S3-Bucket verwendet, die `sts:AssumeRole`-Berechtigung.

11. Wählen Sie Richtlinie aktualisieren.

Erstellen der serviceverknüpften DMS-Fleet-Advisor-Rolle

DMS Fleet Advisor verwendet eine serviceverknüpfte Rolle, um Amazon- CloudWatch Metriken in Ihrem zu verwaltenAWS-Konto. DMS Fleet Advisor verwendet diese serviceverknüpfte Rolle, um die erfassten Datenbankleistungsmetriken CloudWatch in in Ihrem Namen zu veröffentlichen.

So erstellen Sie die serviceverknüpfte Rolle für DMS Fleet Advisor

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Roles. Wählen Sie anschließend Create role (Rolle erstellen) aus.
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS-Service aus.
4. Wählen Sie unter Anwendungsfälle für andere AWS-Services DMS – Fleet Advisor.
5. Markieren Sie das Kontrollkästchen DMS – Fleet Advisor und wählen Sie Weiter.

- Wählen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) die Option Next (Weiter) aus.
- Wählen Sie auf der Seite Benennen, Überprüfen und Erstellen die Option Rolle erstellen aus.

Alternativ können Sie diese serviceverknüpfte Rolle auch über die AWS-API oder AWS-CLI erstellen. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle für AWS DMS Fleet Advisor](#).

Nachdem Sie die serviceverknüpfte Rolle für DMS Fleet Advisor erstellt haben, können Sie die Leistungsmetriken für Ihre Quelldatenbanken in den Zielempfehlungen sehen. Außerdem können Sie diese Metriken und in Ihrem CloudWatch Konto sehen. Weitere Informationen finden Sie unter [Zielempfehlungen](#).

So erstellen Sie eine IAM-Richtlinie, die für die Serviceverknüpfte DMS-Fleet-Advisor-Rolle erforderlich ist

Die zum Erstellen der serviceverknüpften Rolle erforderlichen Mindestberechtigungen sind in der DMSFleetAdvisorCreateServiceLinkedRolePolicy-Richtlinie festgelegt. Erstellen Sie diese IAM-Richtlinie für Ihr Konto, wenn Sie die serviceverknüpfte Rolle nicht erstellen können.

- Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
- Wählen Sie im Navigationsbereich Richtlinien.
- Wählen Sie Richtlinie erstellen aus.
- Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte JSON aus.
- Fügen Sie den folgenden JSON-Code in den Editor ein und ersetzen Sie dabei den Beispielcode.

```
{
  "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. Wählen Sie **Weiter: Tags** und **Weiter: Prüfen** aus.
7. Geben Sie unter **DMSFleetAdvisorCreateServiceLinkedRolePolicyName** den Namen ein und wählen Sie dann **Richtlinie erstellen** aus.

Jetzt können Sie diese Richtlinie verwenden, um die serviceverknüpfte Rolle für DMS Fleet Advisor zu erstellen.

Erstellen von Datenbankbenutzern für AWS DMS Fleet Advisor

In diesem Abschnitt wird beschrieben, wie Sie Datenbankbenutzer mit den Mindestberechtigungen, die für den DMS-Datenkollektor erforderlich sind, erstellen.

In diesem Abschnitt werden folgende Themen behandelt:

- [Verwenden eines Datenbankbenutzers mit AWS DMS Fleet Advisor](#)
- [Erstellen eines Datenbankbenutzers mit MySQL](#)
- [Erstellen einer VPC zur Verwendung mit Oracle](#)
- [Erstellen eines Datenbankbenutzers mit PostgreSQL](#)
- [Erstellen eines Datenbankbenutzers mit Microsoft SQL Server](#)
- [Löschen von Datenbankbenutzern](#)

Verwenden eines Datenbankbenutzers mit AWS DMS Fleet Advisor

Sie können einen anderen Datenbankbenutzer als `root` mit dem DMS-Datenkollektor verwenden. Geben Sie den Benutzernamen und das Passwort an, nachdem Sie die Datenbank zum Inventar hinzugefügt haben, jedoch bevor Sie Ihren Datenkollektor ausführen. Weitere Informationen zum Hinzufügen von Datenbanken zum Inventar finden Sie unter [Verwalten von überwachten Objekten](#).

Nachdem Sie die Verwendung des DMS-Datenkollektors beendet haben, können Sie die von Ihnen erstellten Datenbankbenutzer löschen. Weitere Informationen finden Sie unter [Löschen von Datenbankbenutzern](#).

Important

Ersetzen Sie in den folgenden Beispielen *{your_user_name}* durch den Namen des Datenbankbenutzers, den Sie für Ihre Datenbank erstellt haben. Ersetzen Sie dann *{your_password}* durch ein sicheres Passwort.

Erstellen eines Datenbankbenutzers mit MySQL

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer in einer MySQL-Quelldatenbank zu erstellen. Stellen Sie sicher, dass Sie eine Version der GRANT-Anweisung beibehalten, die von der Version Ihrer MySQL-Datenbank abhängt.

```
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;

```

Erstellen einer VPC zur Verwendung mit Oracle

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer in einer Oracle-Quelldatenbank zu erstellen.

Um dieses SQL-Skript auszuführen, stellen Sie mithilfe von SYSDBA-Berechtigungen eine Verbindung zu Ihrer Oracle-Datenbank her. Nachdem Sie dieses SQL-Skript ausgeführt haben, stellen Sie mit den Anmeldeinformationen des Benutzers, den Sie mit diesem Skript erstellt haben, eine Verbindung zu Ihrer Datenbank her. Verwenden Sie außerdem die Anmeldeinformationen dieses Benutzers, um den DMS-Datenkollektor auszuführen.

Das folgende Skript fügt das C##-Präfix dem Namen des Benutzers für Oracle-Multitenant-Container-Datenbanken (CDB) hinzu.

```

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;
```

Erstellen eines Datenbankbenutzers mit PostgreSQL

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer in einer PostgreSQL-Quelldatenbank zu erstellen.

```
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}";
```

Erstellen eines Datenbankbenutzers mit Microsoft SQL Server

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer in einer Microsoft-SQL-Server-Quelldatenbank zu erstellen.

```
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}]
```

Löschen von Datenbankbenutzern

Nachdem Sie alle Datenerfassungsaufgaben abgeschlossen haben, können Sie die Datenbankbenutzer löschen, die Sie für den DMS-Datenkollektor erstellt haben. Sie können die folgenden Skripts verwenden, um die Benutzer mit Mindestberechtigungen aus Ihren Datenbanken zu löschen.

Führen Sie das folgende Skript aus, um den Benutzer aus Ihrer MySQL-Datenbank zu löschen.

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

Führen Sie das folgende Skript aus, um den Benutzer aus Ihrer Oracle-Datenbank zu löschen.

```
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;
```

Führen Sie das folgende Skript aus, um den Benutzer aus Ihrer PostgreSQL-Datenbank zu löschen.

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

Führen Sie das folgende Skript aus, um den Benutzer aus Ihrer SQLServer-Datenbank zu löschen.

```
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
```

Ermitteln von Datenbanken für die Migration mithilfe von Datenkollektoren

Um Ihre Quelldateninfrastruktur zu ermitteln, können Sie entweder [AWS Application Discovery Service Agentless Collector](#) oder AWS DMS-Datenkollektoren verwenden. Der ADS Agentless Collector ist eine On-Premises-Anwendung, die mithilfe agentenloser Methoden Informationen über Ihre On-Premises-Umgebung sammelt, einschließlich Serverprofilinformationen (z. B. Betriebssystem, Anzahl der CPUs, Größe des Arbeitsspeichers), Datenbankmetadaten und Nutzungsmetriken. Installieren Sie den Agentless Collector als virtuelle Maschine (VM) in Ihrer VMware-vCenter-Server-Umgebung mithilfe einer Open Virtualization Archive (OVA)-Datei. Ein -AWS DMS-Datenkollektor ist eine Windows-Anwendung, die Sie in Ihrer lokalen Umgebung installieren. Diese Anwendung stellt eine Verbindung zu Ihrer Datenumgebung her und erfasst Metadaten und Leistungskennzahlen von Ihren On-Premises-Datenbank- und Analyseservern. Sobald Datenbankmetadaten und Leistungskennzahlen entweder über den ADS Agentless Collector oder einen DMS-Datenkollektor erfasst wurden, erstellt DMS Fleet Advisor eine Bestandsaufnahme von Servern, Datenbanken und Schemata, die Sie zur AWS Cloud migrieren können.

Der DMS-Datenkollektor ist eine Windows-Anwendung, die .NET-Bibliotheken, Konnektoren und Datenanbieter verwendet, um eine Verbindung zu Ihren Quelldatenbanken für die Datenbankerkennung und Datenerfassung herzustellen.

Der DMS-Datenkollektor wird unter Windows ausgeführt. Ihr DMS-Datenkollektor kann jedoch Daten von allen unterstützten Datenbankanbietern erfassen, unabhängig davon, auf welchem Betriebssystemserver sie ausgeführt werden.

Der DMS-Datenkollektor verwendet ein geschütztes RTPS-Protokoll mit TLS-Verschlüsselung, um eine sichere Verbindung zu DMS Fleet Advisor herzustellen. Daher werden Ihre Daten bei der Übertragung standardmäßig verschlüsselt.

AWS DMS hat die maximale Anzahl von Datenkollektoren, die Sie für Ihr AWS-Konto erstellen können. Im folgenden Abschnitt finden Sie Informationen zu AWS DMS-Servicekontingenten [Kontingente für AWS Database Migration Service](#).

Themen

- [Berechtigungen für einen DMS-Datenkollektor](#)
- [Erstellen eines Datenkollektors für AWS DMS Fleet Advisor](#)
- [Installieren und Konfigurieren eines Datenkollektors](#)
- [Ermitteln zu überwachender Betriebssystem- und Datenbankserver](#)
- [Verwalten von überwachten Objekten](#)
- [Verwenden von SSL mit AWS DMS Fleet Advisor](#)
- [Erfassen von Daten für AWS DMS Fleet Advisor](#)
- [Problembehandlung für den DMS-Datenkollektor](#)

Berechtigungen für einen DMS-Datenkollektor

Die Datenbankbenutzer, die Sie für den DMS-Datenkollektor erstellen, sollten über Leserechte verfügen. In einigen Fällen benötigt der Datenbankbenutzer jedoch die EXECUTE-Berechtigung. Weitere Informationen finden Sie unter [Erstellen von Datenbankbenutzern für AWS DMS Fleet Advisor](#).

Der DMS-Datenkollektor benötigt zusätzliche Berechtigungen, um die Ermittlungs-Skripts auszuführen.

- Für die Betriebssystemermittlung benötigt der DMS-Datenkollektor Anmeldeinformationen für den Domain-Server, um Anfragen mithilfe des LDAP-Protokolls ausführen zu können.
- Für die Datenbankermittlung unter Linux benötigt der DMS-Datenkollektor Anmeldeinformationen mit sudo SSH-Berechtigungen. Außerdem sollten Sie Ihre Linux-Server so konfigurieren, dass sie die Remote-Ausführung von SSH-Skripts ermöglichen.
- Für die Datenbankermittlung in Windows benötigt der DMS-Datenkollektor Anmeldeinformationen mit Berechtigungen, um Abfragen mit Windows Management Instrumentation (WMI) und WMI Query Language (WQL) ausführen und die Registrierung lesen zu können. Außerdem sollten Sie Ihre Windows-Server so konfigurieren, dass Remote-WMI-, WQL- und PowerShell -Skripts ausgeführt werden können.

Erstellen eines Datenkollektors für AWS DMS Fleet Advisor

Erfahren Sie, wie Sie einen DMS-Datenkollektor erstellen und herunterladen.

Verwenden Sie die IAM-Konsole, um eine serviceverknüpfte Rolle für DMS Fleet Advisor zu erstellen, bevor Sie einen Datenkollektor erstellen. Diese Rolle ermöglicht es Prinzipalen, Metrikdatenpunkte in Amazon zu veröffentlichen CloudWatch. DMS Fleet Advisor verwendet diese Rolle, um Diagramme mit Datenbankmetriken anzuzeigen. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle für AWS DMS Fleet Advisor](#).

So erstellen Sie einen DMS-Datenkollektor und laden ihn herunter

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.



Wählen Sie die Region aus, in der Sie den DMS Fleet Advisor verwenden.

2. Wählen Sie im Navigationsbereich Datenkollektoren unter Ermitteln. Die Seite Datenkollektoren wird geöffnet.
3. Wählen Sie Datenkollektor erstellen aus. Die Seite Datenkollektor erstellen wird geöffnet.

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. Geben Sie im Abschnitt Allgemeine Konfiguration unter Name den Namen Ihres Datenkollektors ein.
5. Wählen Sie im Abschnitt Konnektivität die Option S3 durchsuchen aus. Wählen Sie den Amazon-S3-Bucket aus der Liste aus, den Sie vorkonfiguriert haben.

AWS DMS speichert Ihre DMS-Fleet-Advisor-Inventarmetadaten in diesem S3-Bucket. Stellen Sie sicher, dass sich Ihr Amazon-S3-Bucket in derselben AWS-Region befindet, in der Ihr AWS DMS Fleet Advisor derzeit ausgeführt wird.

 Note

DMS Fleet Advisor unterstützt nur SSE-S3-verschlüsselte Buckets.

6. Wählen Sie in der Liste der IAM-Rollen die IAM-Rolle aus, die Sie vorkonfiguriert haben. Diese Rolle gewährt AWS DMS-Berechtigungen für den Zugriff auf den angegebenen Amazon-S3-Bucket.
7. Wählen Sie Datenkollektor erstellen aus. Die Seite Datenkollektoren wird geöffnet und der erstellte Datenkollektor wird in der Liste angezeigt.

Wenn Sie Ihren ersten Datenkollektor erstellen, konfiguriert AWS DMS eine Umgebung in Ihrem Amazon-S3-Bucket, die Daten formatiert und Attribute für die Verwendung mit DMS Fleet Advisor speichert.

8. Wählen Sie in dem Informationsbanner die Option Lokalen Kollektor herunterladen, um Ihren neu erstellten Datenkollektor herunterzuladen. In einer Meldung werden Sie darüber informiert, dass der Download im Gange ist. Nach Abschluss des Downloads können Sie auf die `AWS_DMS_Collector_Installer_<version_number>.msi`-Datei zugreifen.

Sie können jetzt den DMS-Datenkollektor auf Ihrem Client installieren. Weitere Informationen finden Sie unter [Installieren und Konfigurieren eines Datenkollektors](#).

Installieren und Konfigurieren eines Datenkollektors

Erfahren Sie, wie Sie Ihren DMS-Datenkollektor installieren, wie Sie Zugangsdaten für die Datenweiterleitung angeben und wie Sie Ihrem Projekt einen LDAP-Server hinzufügen.

In der folgenden Tabelle werden die Hardware- und Softwareanforderungen für die Installation eines DMS-Datenkollektors beschrieben.

Minimum	Empfohlen
Prozessor: 2 Kerne mit einem CPU-Benchmark-Score von mehr als 8 000	Prozessor: 4 Kerne mit einem CPU-Benchmark-Score von mehr als 11 000
RAM: 8 GB	RAM: 16 GB
Festplattengröße: 256 GB	Festplattengröße: 512 GB
Betriebssystem: Microsoft Windows Server 2012 oder höher	Betriebssystem: Windows Server 2016 oder höher

So installieren Sie einen Datenkollektor auf einem Client in Ihrem Netzwerk

1. Führen Sie das .MSI-Installationsprogramm aus. Die Seite mit dem Einrichtungsassistenten für AWS DMS Fleet Advisor Collector wird angezeigt.
2. Wählen Sie Weiter. Die Endbenutzer-Lizenzvereinbarung wird angezeigt.
3. Lesen und akzeptieren Sie die Endbenutzer-Lizenzvereinbarung.
4. Wählen Sie Weiter aus. Die Seite mit dem Zielordner wird angezeigt.
5. Wählen Sie Weiter, um den Datenkollektor im Standardverzeichnis zu installieren.

Oder wählen Sie Ändern, um ein anderes Installationsverzeichnis einzugeben. Wählen Sie anschließend Weiter.

6. Wählen Sie auf der Seite Desktop-Shortcut das Kästchen aus, um ein Symbol auf Ihrem Desktop zu installieren.
7. Wählen Sie Installieren aus. Der Datenkollektor wird in dem Verzeichnis installiert, das Sie ausgewählt haben.
8. Wählen Sie auf der Seite DMS-Kollektor-Einrichtungsassistent abgeschlossen die Option AWS DMS-Kollektor starten und anschließend Fertig stellen.

Ihr DMS-Datenkollektor verwendet .NET-Bibliotheken, Konnektoren und Datenanbieter, um eine Verbindung zu Ihren Quelldatenbanken herzustellen. Das Installationsprogramm für den DMS-Datenkollektor installiert diese erforderliche Software automatisch für alle unterstützten Datenbanken auf Ihrem Server.

Nachdem Sie den Datenkollektor installiert haben, können Sie ihn in einem Browser ausführen, indem Sie die Adresse **http://localhost:11000/** eingeben. Wählen Sie optional im Start-Menü von Microsoft Windows in der Programmliste die Option AWS DMS-Kollektor aus. Wenn Sie den DMS-Datenkollektor zum ersten Mal ausführen, werden Sie aufgefordert, Anmeldeinformationen zu konfigurieren. Erstellen Sie den Benutzernamen und das Passwort für die Anmeldung beim Datenkollektor.

Auf der Startseite des DMS-Datenkollektors finden Sie Informationen zur Vorbereitung und Ausführung der Metadatenerfassung, einschließlich der folgenden Statusbedingungen:

- Status und Zustand Ihrer Datenerfassung.
- Zugriff auf Ihren Amazon-S3-Bucket und auf AWS DMS, damit der Datenkollektor Daten an AWS DMS weiterleiten kann.
- Konnektivität zu Ihren installierten Datenbanktreibern.
- Anmeldeinformationen eines LDAP-Servers für die erste Ermittlung.

The screenshot displays the AWS DMS Collector web interface. At the top, there is a navigation bar with the AWS logo, the text 'DMS Collector', and a 'Sign out' link. Below the navigation bar is a sidebar with a home icon and a 'Home' label. The main content area is divided into three panels:

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

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

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

Der DMS-Datenkollektor verwendet ein LDAP-Verzeichnis, um Informationen über die Computer und Datenbankserver in Ihrem Netzwerk zu erfassen. Das Lightweight Directory Access Protocol (LDAP) ist ein offenes Standardanwendungsprotokoll. Es wird für den Zugriff auf und die Verwaltung verteilter Verzeichnisinformationsservices über ein IP-Netzwerk verwendet. Sie können Ihrem Projekt einen vorhandenen LDAP-Server als Datenkollektor hinzufügen, mit dem Sie Informationen

über die Infrastruktur Ihrer Systeme ermitteln können. Wählen Sie dazu die Option +Server und geben Sie dann einen vollqualifizierten Domainnamen (FQDN) und die Anmeldeinformationen für Ihren Domain-Controller an. Nachdem Sie den Server hinzugefügt haben, validieren Sie die Verbindungsprüfung. Wie Sie den Ermittlungsprozess starten, erfahren Sie unter [Ermitteln zu überwachender Betriebssystem- und Datenbankserver](#).

Konfiguration von Anmeldeinformationen für die Datenweiterleitung

Stellen Sie nach der Installation des Datenkollektors sicher, dass diese Anwendung die erfassten Daten an AWS DMS Fleet Advisor senden kann.

So konfigurieren Sie Anmeldeinformationen für die Datenweiterleitung in AWS DMS Fleet Advisor

1. Wählen Sie auf der Startseite des DMS-Datenkollektors im Abschnitt Datenweiterleitung die Option Weiterleitung konfigurieren aus. Das Dialogfeld Anmeldeinformationen für die Datenweiterleitung konfigurieren wird geöffnet.
2. Wählen Sie die AWS-Region aus, in der Sie DMS Fleet Advisor verwenden möchten.
3. Geben Sie Ihre AWS-Zugriffsschlüssel-ID und Ihren geheimen AWS-Zugriffsschlüssel ein, den Sie zuvor bei der Erstellung von IAM-Ressourcen erhalten haben. Weitere Informationen finden Sie unter [Erstellen von IAM-Ressourcen](#).
4. Wählen Sie Datenkollektoren durchsuchen aus.

Wenn Sie in der angegebenen Region noch keinen Datenkollektor erstellt haben, erstellen Sie einen Datenkollektor, bevor Sie fortfahren. Weitere Informationen finden Sie unter [Erstellen eines Datenkollektors](#).

5. Wählen Sie im Fenster Datenkollektor auswählen einen Datenkollektor aus der Liste aus und klicken Sie auf Auswählen.
6. Wählen Sie im Dialogfeld Anmeldeinformationen für die Datenweiterleitung konfigurieren die Option Speichern.

Vergewissern Sie sich auf der Startseite von DMS Collector auf der Karte Datenweiterleitung, dass die Status Zugriff auf Amazon S3 und Zugriff auf AWS DMS auf Yes gesetzt sind.

Wenn Sie sehen, dass der Status von Zugriff auf Amazon S3 oder Zugriff auf AWS DMS auf Nein gesetzt ist, stellen Sie sicher, dass Sie IAM-Ressourcen für den Zugriff auf Amazon S3 und DMS Fleet Advisor erstellt haben. Nachdem Sie diese IAM-Ressourcen mit allen erforderlichen

Berechtigungen erstellt haben, konfigurieren Sie die Datenweiterleitung erneut. Weitere Informationen finden Sie unter [Erstellen von IAM-Ressourcen](#).

Ermitteln zu überwachender Betriebssystem- und Datenbankserver

Sie können den DMS-Datenkollektor verwenden, um alle verfügbaren Server in Ihrem Netzwerk zu finden und aufzulisten. Es wird empfohlen, alle verfügbaren Datenbankserver in Ihrem Netzwerk zu ermitteln, dies ist aber nicht erforderlich. Optional können Sie die Serverliste für die weitere Datenerfassung manuell hinzufügen oder hochladen. Weitere Informationen zum manuellen Hinzufügen einer Liste von Servern finden Sie unter [Verwalten von überwachten Objekten](#).

Es wird empfohlen, zuerst alle Betriebssystemserver (OS) zu ermitteln, bevor Sie Datenbanken auf diesen Servern ermitteln. Um Betriebssystemserver zu erkennen, benötigen Sie die Berechtigung zum Ausführen von Remote- PowerShell, Secure Shell (SSH)- und Windows Management Instrumentation (WMI)-Skripten und -Befehlen sowie Zugriff auf die Windows-Registrierung. Um Datenbankserver in Ihrem Netzwerk zu ermitteln und Metadaten von diesen zu erfassen, benötigen Sie Administratorrechte mit Lesezugriff für eine Remote-Datenbankverbindung. Stellen Sie sicher, dass Sie einen LDAP-Server hinzugefügt haben, bevor Sie mit der Ermittlung fortfahren. Weitere Informationen finden Sie unter [Konfiguration von Anmeldeinformationen für die Datenweiterleitung](#).

Führen Sie die folgenden Aufgaben aus, um mit der Arbeit mit dem DMS-Datenkollektor zu beginnen:

- Ermitteln Sie alle Betriebssystemserver in Ihrem Netzwerk.
- Fügen Sie bestimmte Betriebssystemserver als zu überwachende Objekte hinzu.
- Überprüfen Sie die Verbindungen für überwachte Betriebssystemserver.
- Ermitteln Sie Microsoft-SQL-Server-, MySQL-, Oracle- und PostgreSQL-Datenbanken, die auf Betriebssystemservern ausgeführt werden.
- Fügen Sie Datenbankserver für die Datenerfassung hinzu.
- Überprüfen Sie die Verbindungen zu den überwachten Datenbanken.

So ermitteln Sie Betriebssystemserver in Ihrem Netzwerk, die Sie überwachen können

1. Wählen Sie im Navigationsbereich des DMS-Datenkollektors die Option Ermitteln aus. Um den Navigationsbereich anzuzeigen, wählen Sie das Menüsymbol oben links auf der Startseite des DMS-Datenkollektors aus.

Die Seite Ermittlung wird geöffnet.

2. Vergewissern Sie sich, dass die Registerkarte Betriebssystemserver ausgewählt ist, und wählen Sie dann Ermittlung durchführen. Das Dialogfeld Parameter eingeben wird angezeigt.
3. Geben Sie die LDAP-Server ein, die Sie verwenden möchten, um Ihr Netzwerk zu scannen.
4. Wählen Sie Ermittlung ausführen. Auf der Seite wird eine Liste aller in Ihrem Netzwerk ermittelten Betriebssystemserver angezeigt, unabhängig davon, ob sie eine Datenbank ausführen.

Es wird empfohlen, zuerst alle Betriebssystemserver (OS) zu ermitteln, bevor Sie Datenbanken auf diesen Servern ermitteln. Ihre Anmeldeinformationen ermöglichen die Ermittlung zunächst für die Hostserver und dann für die Datenbanken, die sich auf diesen befinden. Sie sollten zuerst die Betriebssystemserver ermitteln, bevor Sie die Suche nach Datenbanken auf diesen Servern ausführen. Beachten Sie, dass die Anmeldeinformationen, die Sie für einen LDAP-Server verwenden, um Betriebssystemserver in Ihrem Netzwerk zu finden, sich von den Anmeldeinformationen unterscheiden können, die für die Suche nach Datenbanken auf einem bestimmten Betriebssystemserver erforderlich sind. Daher empfehlen wir, Betriebssystemserver zu überwachten Objekten hinzuzufügen, die Anmeldeinformationen zu überprüfen und sie gegebenenfalls zu korrigieren und anschließend die Konnektivität zu überprüfen, bevor Sie fortfahren.

In der Liste der ermittelten Betriebssystemserver in Ihrem Netzwerk können Sie jetzt die Server auswählen, die Sie zu den überwachten Objekten hinzufügen möchten.

So wählen Sie Betriebssystemserver als zu überwachende Objekte aus

1. Wählen Sie auf der Seite Ermittlung die Registerkarte Betriebssystemserver aus.
2. Markieren Sie in der Liste der ermittelten Betriebssystemserver das Kontrollkästchen neben jedem Server, den Sie überwachen möchten.
3. Wählen Sie Zu überwachten Objekten hinzufügen.

Sie können die Liste der Betriebssystemserver zur Überwachung und Überprüfung von Verbindungen auf der Seite Objekte überwachen anzeigen.

So prüfen Sie die Verbindungen der ausgewählten zu überwachenden Betriebssystemserver

1. Wählen Sie im Navigationsbereich des DMS-Datenkollektors die Option Überwachte Objekte aus.

2. Wählen Sie auf der Seite Überwachte Objekte die Registerkarte Betriebssystemserver aus. Eine Liste der ermittelten Betriebssystemserver, die überwacht werden sollen, wird angezeigt.
3. Markieren Sie das Kontrollkästchen oben in der Spalte, um alle aufgelisteten Betriebssystemserver auszuwählen.
4. Wählen Sie Aktionen und dann Verbindung überprüfen. Sehen Sie sich für jedes Serverobjekt die Ergebnisse in der Spalte Verbindungsstatus an.
5. Wählen Sie Server mit einem anderen Verbindungsstatus als Erfolgreich aus. Wählen Sie als Nächstes Aktionen und anschließend Bearbeiten. Das Dialogfeld Server bearbeiten wird geöffnet.
6. Vergewissern Sie sich, dass die Informationen korrekt sind, oder bearbeiten Sie sie bei Bedarf. Wenn Sie fertig sind, wählen Sie Speichern aus. Das Dialogfeld Anmeldeinformationen überschreiben wird geöffnet.
7. Wählen Sie Überschreiben. Der DMS-Datenkollektor überprüft und aktualisiert den Status für jede Verbindung als Erfolgreich.

Sie können jetzt Datenbanken ermitteln, die sich auf Servern befinden, die Sie zur Überwachung ausgewählt haben.

Ermitteln von auf Servern ausgeführten Datenbanken

1. Wählen Sie im Navigationsbereich des DMS-Datenkollektors die Option Ermitteln aus.
2. Wählen Sie die Registerkarte Datenbankserver und dann Ermittlung ausführen aus. Das Dialogfeld Ermittlungsparameter wird angezeigt.
3. Wählen Sie im Dialogfeld Ermittlungsparameter für Ermittlung nach die Option Überwachte Objekte aus. Wählen Sie unter Server die Betriebssystemserver aus, auf denen Sie die Datenbankermittlung ausführen möchten.
4. Wählen Sie Ermittlung ausführen. Auf der Seite wird eine Liste aller Datenbanken angezeigt, die sich auf den Betriebssystemservern befinden, die Sie überwachen möchten.

Zeigen Sie Informationen wie Datenbankadresse, Servername und Datenbank-Engine an, die Ihnen bei der Auswahl der zu überwachenden Datenbanken helfen.

So wählen Sie die zu überwachenden Datenbanken aus

1. Wählen Sie auf der Seite Ermittlung die Registerkarte Datenbankserver aus.

2. Markieren Sie in der Liste der ermittelten Betriebssystemserver das Kontrollkästchen neben jeder Datenbank, die Sie überwachen möchten.
3. Wählen Sie Zu überwachten Objekten hinzufügen.

Sie können jetzt die Verbindungen zu den Datenbanken überprüfen, die Sie überwachen möchten.

So überprüfen Sie die Verbindungen zu den überwachten Datenbanken

1. Wählen Sie im Navigationsbereich des DMS-Datenkollektors die Option Überwachte Objekte aus.
2. Wählen Sie auf der Seite Überwachte Objekte die Registerkarte Datenbankserver aus. Eine Liste der ermittelten Datenbankserver, die überwacht werden sollen, wird angezeigt.
3. Markieren Sie das Kontrollkästchen oben in der Spalte, um alle aufgelisteten Datenbankserver auszuwählen.
4. Wählen Sie Aktionen und dann Verbindung überprüfen. Sehen Sie sich für jede Datenbank die Ergebnisse in der Spalte Verbindungsstatus an.
5. Wählen Sie Verbindungen aus, die den Status „undefiniert (leer)“ oder den Status Fehler haben. Wählen Sie als Nächstes Aktionen und anschließend Bearbeiten. Das Dialogfeld Überwachte Objekte bearbeiten wird geöffnet.
6. Geben Sie Ihre Anmeldeinformationen und Ihr Passwort ein und wählen Sie dann Speichern. Das Dialogfeld Anmeldeinformationen ändern wird geöffnet.
7. Wählen Sie Überschreiben. Der DMS-Datenkollektor überprüft und aktualisiert den Status für jede Verbindung als Erfolgreich.

Nachdem Sie die zu überwachenden Betriebssystemserver und Datenbanken ermittelt haben, können Sie auch Aktionen zur Verwaltung der überwachten Objekte ausführen.

Verwalten von überwachten Objekten

Sie können Objekte auswählen, die überwacht werden sollen, wenn Sie den Serverermittlungsprozess ausführen, wie unter [Ermitteln von Betriebssystem- und Datenbankservern](#) beschrieben. Dazu können Sie Objekte wie Betriebssystemserver (OS) und Datenbankserver manuell verwalten. Sie können die folgenden Aktionen ausführen, um überwachte Objekte zu verwalten:

- Hinzufügen neuer zu überwachender Objekte
- Entfernen vorhandener Objekte

- Bearbeiten vorhandener Objekte
- Exportieren und importieren einer Liste von Objekten, die überwacht werden sollen
- Überprüfen der Verbindungen zu Objekten
- Starten der Datenerfassung

Beispielsweise können Sie manuell ein zu überwachendes Objekt hinzufügen.

So fügen Sie manuell ein zu überwachendes Objekt hinzu

1. Wählen Sie auf der Seite Überwachte Objekte die Option +Server aus. Das Dialogfeld Überwachtes Objekt hinzufügen wird geöffnet.
2. Fügen Sie Informationen über den Server hinzu und wählen Sie dann Speichern.

Sie können auch eine .csv-Datei verwenden, um eine große Liste von Objekten zu importieren, die überwacht werden sollen. Verwenden Sie das folgende .csv-Dateiformat, um eine Liste von Objekten in den DMS-Datenkollektor zu importieren.

```
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
```

So importieren Sie eine .CSV-Datei mit einer Liste von zu überwachenden Objekten

1. Wählen Sie Importieren aus. Die Seite Überwachte Objekte importieren wird geöffnet.

- Suchen Sie nach der `.csv`-Datei, die Sie importieren möchten, und wählen Sie dann **Weiter**.

Sie können alle Objekte anzeigen und diejenigen auswählen, für die Sie mit dem Erfassen von Metadaten beginnen möchten.

Zuordnen eines Betriebssystemservers zu einer manuell hinzugefügten Datenbank

DMS Fleet Advisor kann Leistungskennzahlen nicht direkt aus MySQL- und PostgreSQL-Datenbanken erfassen. Um die für Zielempfehlungen erforderlichen Kennzahlen zu sammeln, verwendet DMS Fleet Advisor Betriebssystemmetriken dort, wo Ihre Datenbanken ausgeführt werden.

Wenn Sie MySQL- und PostgreSQL-Datenbanken manuell zur Liste der überwachten Objekte hinzufügen, kann der DMS-Datenkollektor die Betriebssystemserver nicht identifizieren, auf denen diese Datenbanken ausgeführt werden. Aufgrund dieses Problems sollten Sie Ihre MySQL- und PostgreSQL-Datenbanken mit Betriebssystemservern verknüpfen.

Sie müssen Betriebssystemserver nicht manuell mit Datenbanken verknüpfen, die DMS Fleet Advisor automatisch ermittelt hat.

So verknüpfen Sie einen Betriebssystemserver mit Ihrer Datenbank

- Wählen Sie im Navigationsbereich des DMS-Datenkollektors die Option **Überwachte Objekte** aus.
- Wählen Sie auf der Seite **Überwachte Objekte** die Registerkarte **Datenbankserver** aus. Es wird eine Liste der Datenbankserver angezeigt.
- Markieren Sie das Kontrollkästchen neben dem MySQL- oder PostgreSQL-Datenbankserver, den Sie manuell hinzugefügt haben.
- Wählen Sie **Aktionen** und anschließend **Bearbeiten**. Das Dialogfeld **Datenbank bearbeiten** wird geöffnet.
- Wenn Ihr DMS-Datenkollektor den Betriebssystemserver, auf dem diese Datenbank ausgeführt wird, bereits erkannt hat, wählen Sie **Automatische Ermittlung**. Der DMS-Datenkollektor führt ein SQL-Skript aus, um den Betriebssystemserver, auf dem Ihre Datenbank ausgeführt wird, automatisch zu identifizieren. Anschließend ordnet der DMS-Datenkollektor diesen Betriebssystemserver Ihrer Datenbank zu. Überspringen Sie den nächsten Schritt und speichern Sie die Datenbankkonfiguration, die Sie bearbeitet haben.

Wenn der DMS-Datenkollektor den Betriebssystemserver für Ihre Datenbank nicht automatisch identifizieren kann, stellen Sie sicher, dass Sie die korrekten Anmeldeinformationen verwenden und Datenbankzugriffsberechtigungen angeben. Optional können Sie den Betriebssystemserver manuell hinzufügen.

6. Um Ihren Betriebssystemserver manuell hinzuzufügen, wählen Sie +Betriebssystemserver hinzufügen. Das Dialogfeld Host-Betriebssystemserver hinzufügen wird geöffnet.

Fügen Sie Informationen zu Ihrem Betriebssystemserver hinzu und wählen Sie dann Speichern.

7. Wählen Sie im Dialogfeld Datenbank bearbeiten die Option Verbindung prüfen aus, um sicherzustellen, dass Ihr DMS-Datenkollektor eine Verbindung zum Betriebssystemserver herstellen kann.
8. Nachdem Sie die Verbindung überprüft haben, wählen Sie Speichern.

Wenn Sie den zugehörigen Betriebssystemserver für Ihre Quelldatenbank ändern, verwendet DMS Fleet Advisor die aktualisierten Metriken, um Empfehlungen zu generieren. Die Amazon- CloudWatch Diagramme zeigen jedoch die alten Daten für Ihren Datenbankserver an. Weitere Informationen zu CloudWatch Diagrammen finden Sie unter [Einzelheiten der Empfehlung](#).

Verwenden von SSL mit AWS DMS Fleet Advisor

Um Ihre Daten zu schützen, kann AWS DMS Fleet Advisor SSL für den Zugriff auf Ihre Datenbanken verwenden.

Unterstützte Datenbanken

AWS DMS Fleet Advisor unterstützt die Verwendung von SSL für den Zugriff auf folgende Datenbanken:

- Microsoft SQL Server
- MySQL
- PostgreSQL

Einrichten von SSL

Um SSL für den Zugriff auf Ihre Datenbank zu verwenden, konfigurieren Sie Ihren Datenbankserver so, dass er SSL unterstützt. Weitere Informationen finden Sie in der folgenden Dokumentation zu Ihrer Datenbank:

- SQL Server: [Aktivieren verschlüsselter Verbindungen zur Datenbank-Engine](#)
- MySQL: [Konfiguration von MySQL für die Verwendung verschlüsselter Verbindungen](#)
- PostgreSQL: [Sichere TCP/IP-Verbindungen mit SSL](#)

Um SSL für die Verbindung zu Ihrer Datenbank zu verwenden, wählen Sie Serverzertifikat vertrauen und SSL verwenden, wenn Sie einen Server manuell hinzufügen. Für eine MySQL-Datenbank können Sie ein benutzerdefiniertes Zertifikat verwenden. Um ein benutzerdefiniertes Zertifikat zu verwenden, markieren Sie das Kontrollkästchen Zertifizierungsstelle verifizieren. Weitere Informationen zum Hinzufügen eines Servers finden Sie unter [Verwalten von überwachten Objekten](#).

Überprüfen des Zertifikats der Serverzertifizierungsstelle (CA) für SQL Server

Wenn Sie Ihr Zertifikat der Serverzertifizierungsstelle (CA) für SQL Server validieren möchten, löschen Sie beim Hinzufügen des Servers die Option Serverzertifikat vertrauen. Wenn Ihr Server eine bekannte Zertifizierungsstelle verwendet und die Zertifizierungsstelle standardmäßig auf Ihrem Betriebssystem installiert ist, sollte die Überprüfung normal funktionieren. Wenn DMS Fleet Advisor keine Verbindung zu Ihrem Datenbankserver herstellen kann, installieren Sie das CA-Zertifikat, das Ihr Datenbankserver verwendet. Weitere Informationen finden Sie unter [Konfigurieren des Clients](#).

Erfassen von Daten für AWS DMS Fleet Advisor

Um mit der Datenerfassung zu beginnen, wählen Sie die Objekte auf der Seite Überwachte Objekte aus und wählen Sie Datenerfassung ausführen. Der DMS-Datenkollektor kann bis zu 100 Datenbanken gleichzeitig erfassen. Außerdem kann der DMS-Datenkollektor bis zu acht parallele Threads verwenden, um eine Verbindung zu Datenbanken in Ihrer Umgebung herzustellen. Von diesen acht Threads kann der DMS-Datenkollektor bis zu fünf parallele Threads verwenden, um eine Verbindung zu einer einzelnen Datenbank-Instance herzustellen.

Important

Bevor Sie mit der Datenerfassung beginnen, sehen Sie sich den Abschnitt Softwareprüfung auf der Startseite des DMS-Datenkollektors an. Stellen Sie sicher, dass alle Datenbank-

Engines, die Sie überwachen möchten, den Status Bestanden haben. Wenn einige Datenbank-Engines den Status Fehlgeschlagen haben und Sie Datenbankserver mit entsprechenden Engines in Ihrer Liste der überwachten Objekte haben, beheben Sie das Problem, bevor Sie fortfahren. Tipps finden Sie neben dem Status Fehlgeschlagen im Abschnitt Softwareprüfung.

Der DMS-Datenkollektor kann in zwei Modi betrieben werden: Einzelbetrieb oder fortlaufende Überwachung. Nachdem Sie mit der Datenerfassung begonnen haben, wird das Dialogfeld Datenerfassung ausführen geöffnet. Wählen Sie dann eine der beiden folgenden Optionen:

Metadaten und Datenbankkapazität

Der DMS-Datenkollektor erfasst Informationen von der Datenbank oder den Betriebssystemservern. Dies beinhaltet Schemata, Versionen, Editionen, CPU, Arbeitsspeicher und Festplattenkapazität. Der DMS-Datenkollektor sammelt auch Metriken wie IOPS, I/O-Durchsatz und aktive Database-Serververbindungen und stellt diese bereit. Auf der Grundlage dieser Informationen können Sie Zielempfehlungen in DMS Fleet Advisor berechnen. Wenn die Quelldatenbank eine Über- oder Unter-Bereitstellung aufweist, werden auch die Zielempfehlungen entsprechend behandelt.

Dies ist die Standardoption.

Metadaten, Datenbankkapazität und Ressourcennutzung

Zusätzlich zu den Metadaten und Informationen zur Datenbankkapazität erfasst der DMS-Datenkollektor die tatsächlichen Nutzungskennzahlen der CPU-, Arbeitsspeicher- und Festplattenkapazität für die Datenbanken oder Betriebssystemserver. Der DMS-Datenkollektor sammelt auch Metriken wie IOPS, I/O-Durchsatz und aktive Database-Serververbindungen und stellt diese bereit. Die bereitgestellten Zielempfehlungen sind genauer, da sie auf den tatsächlichen Datenbank-Workloads basieren.

Wenn Sie diese Option auswählen, legen Sie den Zeitraum der Datenerfassung fest. Sie können Daten in den nächsten 7 Tagen erfassen oder den benutzerdefinierten Zeitraum von 1–60 Tagen festlegen.

Nach Beginn der Datenerfassung werden Sie zur Seite Datenerfassung weitergeleitet, auf der Sie sehen können, wie die Erfassungsabfragen ausgeführt werden, und den Live-Fortschritt überwachen können. Sie können den Gesamtstatus der Erfassung hier oder auf der Startseite des DMS-

Datenkollektors einsehen. Wenn der Gesamtstatus der Datenerfassung unter 100 Prozent liegt, müssen Sie möglicherweise Probleme im Zusammenhang mit der Erfassung beheben.

Wenn Sie den DMS-Datenkollektor im Modus Metadaten und Datenbankkapazität ausführen, können Sie die Anzahl der abgeschlossenen Abfragen auf der Seite Datenerfassung sehen.

Wenn Sie den DMS-Datenkollektor im Modus Metadaten, Datenbankkapazität und Ressourcenauslastung ausführen, können Sie die verbleibende Zeit sehen, bis Ihr DMS-Datenkollektor die Überwachung abschließt.

Auf der Seite Datenerfassung können Sie den Erfassungsstatus für jedes Objekt sehen. Wenn etwas nicht korrekt funktioniert, wird eine Meldung angezeigt, in der angegeben wird, wie viele Probleme aufgetreten sind. Um herauszufinden, wie ein Problem behoben werden kann, können Sie die Details überprüfen. Auf den folgenden Registerkarten sind mögliche Probleme aufgeführt:

- Zusammenfassung nach Abfragen – Zeigt den Status von Tests wie dem Ping-Test an. Sie können die Ergebnisse in der Spalte Status filtern. Die Spalte Status enthält eine Meldung, die angibt, wie viele Fehler bei der Datenerfassung aufgetreten sind.
- Zusammenfassung nach einem überwachten Objekt – Zeigt den Gesamtstatus pro Objekt an.
- Zusammenfassung nach Abfragetyp – Zeigt den Status für den Typ der Kollektor-Abfrage an, z. B. SQL-, Secure Shell- (SSH) oder Windows-Management-Instrumentation (WMI)-Aufrufe.
- Zusammenfassung nach Problem – Zeigt alle einzelnen aufgetretenen Probleme mit Namen und Häufigkeit des Auftretens der einzelnen Probleme an.

Data collection Export to CSV

Collection in progress... ✖ Stop collection
 Metadata, database capacity, and resource utilization data are being collected. Make sure you have proper connectivity to OS and database servers.
 0 d 23 hr 9 min remains

Summary by query | Summary by monitored object | Summary by query type | Summary by issue

Monitored object add...	Co...	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

Um die Erfassungsergebnisse zu exportieren, wählen Sie Zu CSV exportieren aus.

Nachdem Sie die Probleme identifiziert und behoben haben, wählen Sie Erfassung starten und führen Sie den Datenerfassungsprozess erneut aus. Nach der Datenerfassung verwendet der Datenkollektor sichere Verbindungen, um die gesammelten Daten in ein DMS-Fleet-Advisor-Inventar hochzuladen. DMS Fleet Advisor speichert Informationen in Ihrem Amazon-S3-Bucket. Informationen zur Konfiguration der Anmeldeinformationen für die Datenweiterleitung finden Sie unter [Konfiguration von Anmeldeinformationen für die Datenweiterleitung](#).

Erfassung von Kennzahlen zur Kapazität und Ressourcennutzung mit AWS DMS Fleet Advisor

Sie können Metadaten und Leistungskennzahlen in zwei Modi erfassen: Einzelbetrieb oder fortlaufende Überwachung. Je nachdem, welche Option Sie auswählen, verfolgt Ihr DMS-Datenkollektor unterschiedliche Metriken in Ihrer Datenumgebung. Im Einzelbetrieb verfolgt Ihr DMS-Datenkollektor nur Metadatenmetriken von Ihren Datenbank- und Betriebssystemservern. Bei der fortlaufenden Überwachung verfolgt Ihr DMS-Datenkollektor die tatsächliche Nutzung Ihrer Ressourcen.

AWS DMS erfasst die folgenden Metadaten und Metriken während einer einzigen Ausführung Ihres DMS-Datenkollektors.

- Verfügbarer Arbeitsspeicher auf Ihren Betriebssystemservern
- Verfügbarer Speicherplatz auf Ihren Betriebssystemservern
- Datenbankversion und Edition
- Anzahl der CPUs auf Ihren Betriebssystemservern
- Anzahl der Schemata
- Anzahl gespeicherter Prozeduren
- Anzahl der Tabellen
- Anzahl der Auslöser
- Anzahl der Ansichten
- Schema-Struktur

DMS Fleet Advisor verwendet diese Metriken, um eine Bestandsaufnahme Ihrer Datenbank- und Betriebssystemserver zu erstellen. Außerdem verwendet DMS Fleet Advisor diese Metadaten und Metriken, um Ihre Quelldatenbankschemata zu analysieren.

DMS Fleet Advisor kann Zielempfehlungen mithilfe der Metriken generieren, die während einer einzigen Ausführung des Datenkollektors erfasst wurden. In diesem Fall wird die Zielempfehlung für Ihre überdimensionierten Quelldatenbanken jedoch ebenfalls überdimensioniert. Daher fallen zusätzliche Kosten für die Wartung Ihrer Ressourcen in der anAWS Cloud. Bei Quelldatenbanken mit unzureichender Bereitstellung enthält die Zielempfehlung ebenfalls unzureichende Bereitstellungen, was zu Leistungsproblemen führen kann. Wir empfehlen, die Daten mithilfe der laufenden Überwachung zu sammeln, indem Sie die Metadaten, die Datenbankkapazität und den Ressourcennutzungsmodus für den DMS-Datenkollektor auswählen.

AWS DMS erfasst bei fortlaufender Überwachung die folgenden Metriken. Sie können Ihren DMS-Datenkollektor für einen Zeitraum von 1 bis 60 Tagen betreiben.

- E/A-Durchsatz auf Ihren Datenbankservern
- Eingabe-/Ausgabevorgänge pro Sekunde (IOPS) auf Ihren Datenbankservern
- Anzahl der CPUs auf Ihren Betriebssystemservern
- Arbeitsspeichernutzung auf Ihren Betriebssystemservern
- Anzahl der aktiven Datenbank- und Betriebssystemserververbindungen

DMS Fleet Advisor verwendet diese Metriken, um präzise Zielempfehlungen zu generieren, so dass Ihre Zieldatenbanken Ihren Leistungsanforderungen entsprechen. Dies kann zusätzliche Kosten für die Wartung Ihrer Ressourcen in der verhindern AWS Cloud.

Wie erfasst der AWS DMS Fleet Advisor Metriken zur Kapazitäts- und Ressourcennutzung?

DMS Fleet Advisor erfasst jede Minute Leistungsmetriken.

Für Oracle und SQL Server führt DMS Fleet Advisor SQL-Abfragen aus, um Werte für jede Datenbankmetrik zu erfassen.

Für MySQL und PostgreSQL erfasst DMS Fleet Advisor Leistungsmetriken von dem Betriebssystemserver, auf dem Ihre Datenbank ausgeführt wird. Unter Windows führt DMS Fleet Advisor WQL-Skripts (WMI Query Language) aus und empfängt WMI-Daten. Unter Linux führt DMS Fleet Advisor Befehle aus, die die Metriken von Betriebssystemservern erfassen.

Important

Das Ausführen von Remote-SQL-Skripts kann sich auf die Leistung Ihrer Produktionsdatenbanken auswirken. Die Datenerfassungsabfragen enthalten jedoch keine Berechnungslogik. Daher wird der Datenerfassungsprozess wahrscheinlich nicht mehr als 1 Prozent Ihrer Datenbankressourcen beanspruchen.

Sie können alle Abfragen anzeigen, die der Datenkollektor zur Erfassung von Metriken ausführt. Öffnen Sie dazu die `DMSCollector.Collections.json`-Datei. Sie finden diese Datei in dem `etc`-Ordner, der sich in demselben Ordner befindet, in dem Sie den Datenkollektor installiert haben. Der Standardpfad ist `C:\ProgramData\Amazon\AWS DMS Collector\etc\DMSCollector.Collections.json`.

Der DMS-Datenkollektor verwendet das lokale Dateisystem als temporären Speicher für alle erfassten Daten. Der DMS-Datenkollektor speichert die erfassten Daten im JSON-Format. Sie können den lokalen Kollektor im Offline-Modus verwenden und die erfassten Dateien manuell überprüfen oder verifizieren, bevor Sie die Datenweiterleitung konfigurieren. Sie finden alle erfassten Dateien in dem `out`-Ordner, der sich in demselben Ordner befindet, in dem Sie den DMS-Datenkollektor installiert haben. Der Standardpfad ist `C:\ProgramData\Amazon\AWS DMS Collector\out`.

⚠ Important

Wenn Sie Ihren DMS-Datenkollektor im Offline-Modus ausführen und die erfassten Daten länger als 14 Tage auf Ihrem Server speichern, können Sie Amazon nicht verwenden, CloudWatch um diese Metriken anzuzeigen. DMS Fleet Advisor verwendet diese Daten jedoch weiterhin, um Empfehlungen zu generieren. Weitere Informationen zu CloudWatch Diagrammen finden Sie unter [Einzelheiten der Empfehlung](#).

Sie können die erfassten Datendateien auch im Online-Modus überprüfen oder verifizieren. Der DMS-Datenkollektor leitet alle Daten an den Amazon-S3-Bucket weiter, den Sie in den DMS-Datenkollektor-Einstellungen angegeben haben.

Sie können Ihren DMS-Datenkollektor verwenden, um Daten aus On-Premises-Datenbanken zu sammeln. Sie können auch Daten aus Amazon-RDS- und Aurora-Datenbanken erfassen. Aufgrund der Unterschiede zwischen Amazon-RDS- oder Aurora- und On-Premises-DB-Instances können Sie jedoch nicht alle DMS-Datenkollektor-Abfragen erfolgreich in der Cloud ausführen. Da der DMS-Datenkollektor Nutzungsmetriken für MySQL- und PostgreSQL-Datenbanken vom Host-Betriebssystem erfasst, funktioniert dieser Ansatz nicht mit Amazon RDS und Aurora.

Problembehandlung für den DMS-Datenkollektor

In der folgenden Liste finden Sie Maßnahmen, die Sie ergreifen können, wenn beim Erfassen von Daten mit dem DMS-Datenkollektor Probleme auftreten.

Themen

- [Probleme bei der Datenerfassung im Zusammenhang mit Netzwerk- und Serververbindungen](#)
- [Probleme bei der Datenerfassung im Zusammenhang mit Windows Management Instrumentation](#)
- [Probleme bei der Datenerfassung im Zusammenhang mit Windows Webpage Composer](#)
- [Probleme bei der Datenerfassung im Zusammenhang mit SSL](#)

Probleme bei der Datenerfassung im Zusammenhang mit Netzwerk- und Serververbindungen

NET: Bei einer Ping-Anfrage ist eine Ausnahme aufgetreten.

Überprüfen Sie den Namen des Computers, um festzustellen, ob dieser sich in einem Zustand befindet, in dem er nicht zu einer IP-Adresse aufgelöst werden kann.

Prüfen Sie beispielsweise, ob der Computer ausgeschaltet, vom Netzwerk getrennt oder außer Betrieb genommen ist.

NET: Zeitlimit überschritten

Aktivieren Sie die eingehende Firewallregel „Datei- und Druckerfreigabe (Echo Request - ICMPv4-In)“. Beispielsweise:

- * Inbound ICMPv4

NET: DestinationHostUnreachable

Überprüfen Sie die IP-Adresse des Computers. Prüfen Sie insbesondere, ob dieser sich im selben Subnetz befindet wie der Computer, auf dem der DMS-Datenkollektor ausgeführt wird, und ob er auf ARP-Anfragen (Address Resolution Protocol) reagiert.

Befindet sich der Computer in einem anderen Subnetz, kann die IP-Adresse des Gateways nicht zur MAC-Adresse (Media Access Control) aufgelöst werden.

Prüfen Sie auch, ob der Computer ausgeschaltet, vom Netzwerk getrennt oder außer Betrieb genommen ist.

Probleme bei der Datenerfassung im Zusammenhang mit Windows Management Instrumentation

WMI: Der RPC-Server ist nicht verfügbar. (Ausnahme von HRESULT: 0x800706BA)

Aktivieren Sie die eingehende Firewallregel „Windows Management Instrumentation (DCOM-IN)“. Beispielsweise:

- * Inbound TCP/IP at local port 135.

Aktivieren Sie außerdem die eingehende Firewallregel „Windows Management Instrumentation (WMI-IN)“. Beispielsweise:

- * Inbound TCP/IP at local port 49152 - 65535 für Windows Server 2008 und höhere Versionen.
- * Inbound TCP/IP at local port 1025 - 5000 für Windows Server 2003 und niedrigere Versionen.

WMI: Zugriff verweigert. (Ausnahme von HRESULT: 0x80070005 (E_ACCESSDENIED))

Gehen Sie wie folgt vor:

- Fügen Sie den DMS-Datenkollektor der Windows-Gruppe Distributed COM Users oder Administrators hinzu.
- Starten Sie den Windows-Management-Instrumentation-Service und legen Sie dessen Starttyp auf „Automatisch“ fest.
- Stellen Sie sicher, dass Ihr Benutzername für den DMS-Datenkollektor das \-Format hat.

WMI: Zugriff verweigert.

Fügen Sie dem Benutzer des DMS-Datenkollektors im Stamm-WMI-Namespace die Berechtigung „Remote Enable“ hinzu.

Verwenden Sie die erweiterten Einstellungen und stellen Sie sicher, dass die Berechtigungen für „diesen Namespace und die Subnamespaces“ gelten.

WMI: Der Aufruf wurde durch den Nachrichtenfilter abgebrochen. (Ausnahme von HRESULT: 0x80010002...)

Starten Sie den Windows-Management-Instrumentation-Service neu.

Probleme bei der Datenerfassung im Zusammenhang mit Windows Webpage Composer

WPC: Der Netzwerkpfad wurde nicht gefunden

Aktivieren Sie die eingehende Firewallregel „Datei- und Druckerfreigabe (SMB-In)“.
Beispielsweise:

- * Inbound TCP/IP at local port 445.

Starten Sie außerdem den Remote Registry Service und legen Sie den Starttyp auf „Automatisch“ fest.

WPC: Zugriff verweigert

Fügen Sie den Benutzer des DMS-Datenkollektors zur Gruppe der Benutzer oder Administratoren des Performance Monitors hinzu.

WPC: Kategorie ist nicht vorhanden

Führen Sie `loader /r` aus, um den Performance-Counter-Cache neu zu erstellen, und starten Sie dann Ihren Computer neu.

Note

Informationen zur Behebung von Problemen bei der Datenmigration mit AWS Database Migration Service (AWS DMS) finden Sie unter [Problembehandlung und Diagnoseunterstützung](#).

Probleme bei der Datenerfassung im Zusammenhang mit SSL

SSL-Fehler

Ihre Datenbank benötigt eine sichere SSL-Verbindung und Sie haben die Optionen CA prüfen und SSL verwenden für die Verbindung nicht aktiviert. Aktivieren Sie diese Optionen und stellen Sie sicher, dass auf Ihrem lokalen Betriebssystem die Zertifizierungsstelle installiert ist, die Ihre Datenbank verwendet. Weitere Informationen finden Sie unter [Einrichten von SSL](#).

Verwenden von Inventaren für Analysen in AWS DMS Fleet Advisor

Um die Durchführbarkeit potenzieller Datenbankmigrationen zu überprüfen, können Sie mit Inventaren ermittelter Datenbanken und Schemata arbeiten. Anhand der Informationen in diesen Inventaren können Sie ermitteln, welche Datenbanken und Schemata sich für eine Migration eignen.

Sie können über die Konsole auf Datenbank- und Schemainventare zugreifen. Wählen Sie dazu in der Konsole Inventar aus.

DMS Fleet Advisor analysiert Ihre Datenbankschemata, um die Ähnlichkeit verschiedener Schemata festzustellen. Bei dieser Analyse wird der tatsächliche Code für Objekte nicht verglichen. DMS Fleet Advisor vergleicht nur die Namen von Schemaobjekten, wie Funktionen und Prozeduren, um ähnliche Objekte in verschiedenen Datenbankschemata zu identifizieren.

Themen

- [Verwendung eines Datenbankinventars für die Analyse](#)
- [Verwendung eines Schemainventars für die Analyse](#)

Verwendung eines Datenbankinventars für die Analyse

Gehen Sie wie folgt vor, um eine Liste aller Datenbanken auf allen ermittelten Servern in Ihrem Netzwerk anzuzeigen, von denen Daten erfasst wurden.

So zeigen Sie eine Liste der Datenbanken auf Ihren Netzwerkservern an, von denen Daten erfasst wurden

1. Wählen Sie in der Konsole Inventar aus.

Die Seite Inventar wird geöffnet.

2. Wählen Sie die Registerkarte Databases (Datenbanken) aus.

Eine Liste von ermittelten Datenbanken wird angezeigt.

The screenshot shows the 'Inventory' page in the AWS DMS console. At the top, there is a header 'Inventory' with an 'Info' link and a description: 'Database servers, schema information, and metadata discovered by data collectors.' To the right is an orange button labeled 'Analyze inventories'. Below this is a blue information box with an 'i' icon and the 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](#)'.

Below the information box are two tabs: 'Databases' (selected) and 'Schemas'. Under the 'Databases' tab, there is a section titled 'Databases (7)' with a refresh button, an 'Export to CSV' button, and a 'Delete' button. A search bar contains the text 'Find database inventory'. Below the search bar is a table with the following columns: 'Database', 'Server', 'Number of s...', 'Engine', 'Engine version', and 'Engine ...'. The table contains three rows:

<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

- Wählen Sie Inventare analysieren, um Schemaeigenschaften wie Ähnlichkeit und Komplexität zu ermitteln. Die Dauer dieses Vorgangs hängt von der Anzahl der zu analysierenden Objekte ab, sollte jedoch eine Stunde nicht überschreiten. Die Ergebnisse der Analyse finden Sie auf der Registerkarte Schemata auf der Seite Inventar.

DMS Fleet Advisor analysiert Schemata für alle erkannten Datenbanken, um die Schnittmenge ihrer Objekte zu definieren. Das Analyseergebnis wird in Prozent ausgedrückt. DMS Fleet Advisor betrachtet Schemata mit Schnittmengen von mehr als 50 Prozent als Duplikate. Das ursprüngliche Schema wird als das Schema identifiziert, für das Duplikate gefunden werden. Dies hilft, ursprüngliche Schemata zu identifizieren, die zuerst konvertiert oder migriert werden sollen.

Das gesamte Inventar wird zusammen analysiert, um doppelte Schemata zu identifizieren.

Verwendung eines Schemainventars für die Analyse

Sie können eine Liste der Datenbankschemata anzeigen, die auf Servern in Ihrem Netzwerk ermittelt wurden, von denen die Daten erfasst wurden. Führen Sie die folgenden Schritte aus.

So zeigen Sie eine Liste der Schemata auf Ihren Netzwerkservern an, von denen Daten erfasst wurden

1. Wählen Sie in der Konsole Inventar aus. Die Seite Inventar wird geöffnet.
2. Wählen Sie die Registerkarte Schemata aus. Eine Liste von Schemata wird angezeigt.
3. Wählen Sie ein Schema in der Liste aus, um Informationen zu diesem Schema, einschließlich Server, Datenbank, Größe und Komplexität, anzuzeigen.

Für jedes Schema können Sie eine Objektzusammenfassung mit Informationen zu Objekttypen, Anzahl der Objekte, Objektgröße und Codezeilen anzeigen.

4. (Optional) Wählen Sie Inventare analysieren, um doppelte Schemata zu identifizieren. DMS Fleet Advisor analysiert Datenbankschemata, um die Schnittmenge ihrer Objekte zu definieren.
5. Sie können Inventarinformationen zur weiteren Überprüfung in eine .csv-Datei exportieren.

The screenshot shows the AWS DMS console interface for the Schema Inventory feature. At the top, there is a green header with 'Analysis complete' and a 'Schema inventory' tab. Below the header, the breadcrumb 'DMS > Discover: Inventory' is visible. 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 of this sub-heading is an orange button labeled 'Analyze inventories'. Below this is a light blue information box with an 'i' icon, stating: '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 information box are two tabs: 'Databases' and 'Schemas', with 'Schemas' being the active tab. Under the 'Schemas' tab, there is a section titled 'Schemas (13)' with a sub-heading 'Schema inventories that were discovered by data collectors.' To the right of this section are two buttons: a refresh icon and 'Export to CSV'. Below these is a search bar with the placeholder text 'Find schema inventory' and a pagination control showing '< 1 >' and a settings gear icon. The main content is a table 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

Um zu migrierende Schemata zu identifizieren und das Migrationsziel zu bestimmen, können Sie AWS Schema Conversion Tool (AWS SCT) oder DMS Schema Conversion verwenden. Weitere Informationen finden Sie unter [Verwenden eines neuen Projektassistenten in AWS SCT](#).

Nachdem Sie die zu migrierenden Schemata identifiziert haben, können Sie Schemata mithilfe von AWS SCT DMS Schema Conversion konvertieren. Weitere Informationen zu DMS Schema Conversion finden Sie unter [Konvertieren von Datenbankschemata mithilfe von DMS Schema Conversion](#).

Verwenden des AWS DMS-Fleet-Advisor-Features „Zielempfehlungen“

Um ein optimales Migrationsziel zu ermitteln und auszuwählen, können Sie in DMS Fleet Advisor Zielempfehlungen für Ihre On-Premises-Quelldatenbanken generieren. Eine Empfehlung beinhaltet eine oder mehrere mögliche AWS-Ziel-Engines, die Sie für die Migration Ihrer On-Premises-Quelldatenbank auswählen können. Aus diesen möglichen Ziel-Engines schlägt DMS Fleet Advisor eine einzelne Ziel-Engine als Migrationsziel mit der richtigen Größe vor und gibt dieses Ziel als DMS-empfohlen an. Um dieses Migrationsziel mit der richtigen Größe zu ermitteln, verwendet DMS Fleet Advisor die von Ihrem Datenkollektor erfassten Inventarmetadaten und Metriken.

Sie können Empfehlungen vor Beginn einer Migration verwenden, um Migrationsoptionen zu ermitteln, Kosten zu sparen und Risiken zu reduzieren. Sie können Empfehlungen als CSV-Datei exportieren und sie an wichtige Beteiligte weitergeben, um die Entscheidungsfindung zu erleichtern. Sie können Empfehlungen in die exportieren AWS Pricing Calculator, um die Wartungskosten weiter zu optimieren. Weitere Informationen finden Sie unter <https://calculator.aws/#/>.

Sie können Zielempfehlungen in DMS Fleet Advisor nicht ändern. Daher können Sie DMS Fleet Advisor nicht für Was-wäre-wenn-Analysen verwenden. Bei einer Was-wäre-wenn-Analyse werden die Zielparameter geändert, um festzustellen, wie sich diese Änderungen auf die Preisschätzung Ihrer Empfehlung auswirken. Sie können eine Was-wäre-wenn-Analyse in AWS Pricing Calculator durchführen, indem Sie die empfohlenen Zielparameter als Ausgangspunkt in AWS Pricing Calculator verwenden. Weitere Informationen finden Sie unter <https://calculator.aws/#/>.

Wir empfehlen, die Empfehlung von DMS Fleet Advisor als Ausgangspunkt für Ihre Migrationsplanung zu berücksichtigen. Sie können dann entscheiden, die empfohlenen Instance-Parameter zu ändern, um die Kosten oder die Leistung Ihrer Datenbank-Workloads zu optimieren.

Themen

- [Empfohlene Ziel-Instances](#)
- [Wie bestimmt DMS Fleet Advisor die Spezifikationen der Ziel-Instance für die Empfehlung?](#)
- [Generierung von Zielempfehlungen mit AWS DMS Fleet Advisor](#)

- [Informationen zu den Einzelheiten von Zielempfehlungen mit AWS DMS Fleet Advisor](#)
- [Exportieren von Zielempfehlungen mit AWS DMS Fleet Advisor](#)
- [Erkennen und Analysieren von Migrationsbeschränkungen mit AWS DMS Fleet Advisor](#)
- [Fehlerbehebung für Zielempfehlungen](#)

Empfohlene Ziel-Instances

Bei Zielempfehlungen berücksichtigt DMS Fleet Advisor die folgenden Allzweck-, arbeitsspeicheroptimierten und Spitzenlastleistungen von Amazon-RDSDB-Instances.

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

Weitere Informationen zu Amazon-RDS-DB-Instance-Klassen finden Sie unter [DB-Instance-Klassen](#) im Amazon-RDS-Benutzerhandbuch.

Wie bestimmt DMS Fleet Advisor die Spezifikationen der Ziel-Instance für die Empfehlung?

DMS Fleet Advisor kann Empfehlungen entweder auf der Grundlage der Datenbankkapazität oder der Datenbanknutzung generieren.

- Wenn Sie sich dafür entscheiden, die Empfehlung auf der Grundlage der Datenbankkapazität zu generieren, ordnet DMS Fleet Advisor die vorhandene Datenbankkapazität den Spezifikationen der nächstgelegenen Instance-Klasse zu.
- Wenn Sie sich dafür entscheiden, die Empfehlung auf der Grundlage der Ressourcennutzung zu generieren, bestimmt DMS Fleet Advisor den 95. Perzentilwert für Kennzahlen wie CPU, Arbeitsspeicher, E/A-Durchsatz und IOPS. Das 95. Perzentil bedeutet, dass 95 Prozent der

erfassten Daten unter diesem Wert liegen. Anschließend ordnet DMS Fleet Advisor diese Werte den Spezifikationen der Instance-Klasse zu, die diesen am nächsten kommt.

Um die Größe der Zieldatenbank zu bestimmen, erfasst DMS Fleet Advisor Informationen über die Größe Ihrer Quelldatenbank. Anschließend empfiehlt DMS Fleet Advisor, dieselbe Größe für den Zielspeicher zu verwenden. Wenn Ihr Quelldatenbankspeicher übermäßige Bereitstellung aufweist, gilt dies auch für die empfohlene Größe des Zielspeichers.

Wenn Sie Daten mithilfe von AWS DMS migrieren möchten, müssen Sie möglicherweise die IOPS-Bereitstellung für Ihre Ziel-DB-Instance erhöhen. Wenn DMS Fleet Advisor Zielempfehlungen generiert, berücksichtigt der Service nur Ihre Quelldatenbank-Metriken. DMS Fleet Advisor berücksichtigt keine zusätzlichen IOPS, die Sie möglicherweise für die Ausführung von Datenmigrationsaufgaben benötigen. Weitere Informationen finden Sie unter [Migrationsaufgaben werden langsam ausgeführt](#).

Zur Schätzung der IOPS-Kosten verwendet DMS Fleet Advisor eine one-to-one Zuordnung Ihrer Quell-IOPS-Nutzung als Grundlage. DMS Fleet Advisor betrachtet die Spitzenlast als Basiswert und eine 100-prozentige Auslastung bei der Preisgestaltung für IOPS.

Für PostgreSQL- und MySQL-Quelldatenbanken kann DMS Fleet Advisor Aurora- und Amazon-RDS-DB-Instances in die Zielempfehlungen aufnehmen. Wenn eine Aurora-Konfiguration den Quellenanforderungen entspricht, markiert DMS Fleet Advisor diese Option als empfohlen.

Generierung von Zielempfehlungen mit AWS DMS Fleet Advisor

Nachdem Sie die Datenerfassung und Inventarisierung Ihrer Datenbank- und Analyseflotte abgeschlossen haben, können Sie in DMS Fleet Advisor Zielempfehlungen generieren. Wählen Sie dazu Quelldatenbanken aus und konfigurieren Sie die Einstellungen, anhand derer das Feature „Zielempfehlungen“ von DMS Fleet Advisor die Größe der Ziel-Instances bestimmt. Außerdem verwendet dieses Feature von DMS Fleet Advisor die Kapazitäts- und Nutzungsmetriken, die aus Ihren Quelldatenbanken erfasst wurden.

So generieren Sie Zielempfehlungen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.

Stellen Sie sicher, dass Sie die AWS-Region ausgewählt haben, in der Sie den DMS Fleet Advisor verwenden.

2. Wählen Sie im Navigationsbereich Empfehlungen unter Bewerten und dann Empfehlungen generieren.
3. Markieren Sie im Bereich Quelldatenbanken auswählen die Kontrollkästchen für die Namen der Datenbanken, die Sie zu AWS Cloud migrieren möchten.

Geben Sie unter Quelldatenbanken durchsuchen den Namen Ihrer Datenbank ein, um Ihr Inventar zu filtern.

DMS Fleet Advisor kann Empfehlungen für bis zu 100 Datenbanken gleichzeitig generieren.

4. Wählen Sie für Verfügbarkeit und Dauerhaftigkeit die bevorzugte Bereitstellungsoption.

Um Zielempfehlungen für Ihre Produktionsdatenbanken zu berechnen, wählen Sie Produktion (Multi-AZ). DMS Fleet Advisor umfasst zwei DB-Instances in verschiedenen Availability Zones in Ihrer Zielempfehlung. Diese Multi-AZ-Bereitstellungsoption bietet hohe Verfügbarkeit, Datenredundanz und Failover-Unterstützung.

Wenn Aurora die empfohlene Ziel-Engine und Verfügbarkeit und Beständigkeit eine Multi-AZ-Bereitstellung ist, enthält die Zielempfehlung eine Reader- und Writer-DB-Instance.

Um Zielempfehlungen für Datenbanken zu berechnen, die Sie für Entwicklungs- oder Testzwecke verwenden, wählen Sie Dev/Test (Single-AZ). DMS Fleet Advisor nimmt eine einzelne DB-Instance in Ihre Zielempfehlung auf. Diese Single-AZ-Bereitstellungsoption reduziert die Wartungskosten.

5. Wählen Sie für die Größe der Ziel-Instance die bevorzugte Option aus, die DMS Fleet Advisor zur Berechnung der Zielempfehlungen verwendet.

Um die Zielempfehlungen auf der Grundlage Ihrer Quelldatenbank- oder Betriebssystemserverkonfiguration zu berechnen, wählen Sie Gesamtkapazität aus. DMS Fleet Advisor verwendet Metriken wie die gesamte CPU-, Arbeitsspeicher- und Festplattenkapazität Ihrer Quelldatenbanken oder Betriebssystemserver, um Zielempfehlungen zu generieren. Anschließend ordnet DMS Fleet Advisor Ihre Datenbankkapazitätsmetriken den Spezifikationen der nächstgelegenen Amazon-RDS-DB-Instance-Klasse zu.

Um Zielempfehlungen auf der Grundlage der tatsächlichen Nutzung Ihrer Quelldatenbank oder Ihres Betriebssystemservers zu berechnen, wählen Sie Ressourcennutzung. DMS Fleet Advisor verwendet Nutzungsmetriken wie die gesamte CPU-, Arbeitsspeicher- und Festplattenkapazität Ihrer Quelldatenbanken oder Betriebssystemserver, um Zielempfehlungen zu generieren. DMS Fleet Advisor berechnet anhand der Nutzungsmetriken das 95. Perzentil für jede Metrik. Das

95. Perzentil bedeutet, dass 95 Prozent der Daten innerhalb des Zeitraums unter diesem Wert liegen. Anschließend ordnet DMS Fleet Advisor diese Werte den Spezifikationen der Amazon-RDS-DB-Instance-Klasse zu, die diesen am nächsten kommt.

Es wird empfohlen, die Option Ressourcennutzung zu verwenden, um genauere Empfehlungen zu erhalten. Dazu müssen Sie die Metriken zur Gesamtkapazität und zur Ressourcennutzung erfasst haben.

6. Wählen Sie Generate (Erstellen).

DMS Fleet Advisor generiert Zielempfehlungen für die ausgewählten Datenbanken. Für erfolgreich generierte Empfehlungen setzt DMS Fleet Advisor den Status auf Berechnet. Außerdem verwendet DMS Fleet Advisor AWS Pricing Calculator um die geschätzten monatlichen Kosten für die empfohlene Ziel-DB-Instance zu ermitteln. Jetzt können Sie die generierten Empfehlungen im Detail untersuchen. Weitere Informationen finden Sie unter [Einzelheiten der Empfehlung](#).

Um die monatlichen Gesamtkosten für Ihr Dateninventar zu schätzen, markieren Sie die Kontrollkästchen für Datenbanken, die Sie in die Cloud verschieben möchten. DMS Fleet Advisor zeigt die geschätzten monatlichen Gesamtkosten und die Zusammenfassung Ihrer Zieldatenbanken in der AWS Cloud an. DMS Fleet Advisor verwendet die AWS-Preisliste-Abfrage-API, um Preisdetails nur zu Ihrer Information bereitzustellen. Ihre tatsächlichen Gebühren hängen von einer Vielzahl von Faktoren ab, einschließlich Ihrer tatsächlichen Nutzung von AWS-Services. Weitere Informationen zu den AWS-Service-Preisen finden Sie unter [Cloud-Services – Preise](#).

Informationen zu den Einzelheiten von Zielempfehlungen mit AWS DMS Fleet Advisor

Nachdem DMS Fleet Advisor Zielempfehlungen generiert hat, können Sie die wichtigsten Parameter des empfohlenen Migrationsziels in der Tabelle mit den Empfehlungen einsehen. Zu diesen zentralen Parametern gehören die Ziel-Engine, die Instance-Klasse, die Anzahl der virtuellen CPUs, der Arbeitsspeicher, der Speicher und der Speichertyp. Zusätzlich zu diesen Parametern zeigt DMS Fleet Advisor die geschätzten monatlichen Kosten für dieses empfohlene Migrationsziel an.

Jede Empfehlung kann eine oder mehrere mögliche AWS-Ziel-Engines enthalten. Wenn Ihre Empfehlung mehrere Ziel-Engines enthält, markiert AWS DMS eine davon als empfohlen. Außerdem zeigt AWS DMS die Parameter und die geschätzten monatlichen Kosten für diese empfohlene Option in der Tabelle mit den Empfehlungen an.

Um die Zielempfehlungen mit der Nutzung und Kapazität Ihrer Quelldatenbank zu vergleichen, sollten Sie Ihre Empfehlungen im Detail untersuchen. Außerdem können Sie sich die Migrationsbeschränkungen für eine ausgewählte Empfehlung ansehen. Zu diesen Einschränkungen gehören nicht unterstützte Datenbank-Features, Aktionselemente und weitere Überlegungen zur Migration.

So untersuchen Sie die Empfehlung im Detail

1. Generieren Sie Zielempfehlungen mit DMS Fleet Advisor. Weitere Informationen finden Sie unter [Generieren von Zielempfehlungen](#).
2. Wählen Sie den Namen der Empfehlung aus der Tabelle mit den Empfehlungen aus. Die Seite „Empfehlungen“ wird geöffnet.
3. Wenn Ihre Empfehlung mehr als eine Zieloption umfasst, wählen Sie unter Zielempfehlungen die Zieloption aus.
4. Erweitern Sie den Abschnitt Quellennutzung und Kapazität. DMS Fleet Advisor zeigt Diagramme zur Ressourcennutzung für die folgenden Metriken an.
 - Anzahl der CPUs
 - Arbeitsspeicher
 - E/A-Durchsatz
 - Eingabe-/Ausgabevorgänge pro Sekunde (IOPS)
 - Speicher
 - Anzahl der aktiven Datenbankserververbindungen

Verwenden Sie diese Diagramme, um Ihre Quelldatenbank-Metriken aus Ihrem DMS-Datenkollektor mit den Metriken der ausgewählten Ziel-Engine zu vergleichen.

Wenn Sie Diagramme nicht sehen können, nachdem Sie den Abschnitt Quellauslastung und -kapazität erweitert haben, stellen Sie sicher, dass Sie Ihrem IAM-Benutzer die Berechtigungen zum Anzeigen von Amazon- CloudWatch Dashboards erteilt haben. Weitere Informationen finden Sie unter [Verwenden von Amazon- CloudWatch Dashboards](#) im Amazon- CloudWatch Benutzerhandbuch.

5. Wählen Sie den Link mit dem Namen Ihrer ausgewählten Ziel-Engine. Die Seite Ziel-Detail wird geöffnet.

6. Um die Zielempfehlungen nach CSV zu exportieren, wählen Sie in der Dropdownliste Aktionen die Option Nach CSV exportieren aus.
7. Um die Zielempfehlungen nach zu exportierenAWS Pricing Calculator, wählen Sie im Dropdown-Menü Aktionen die Option Kosten optimieren mit AWS Pricing Calculator aus.
8. Vergleichen Sie im Abschnitt Konfiguration die Werte Ihrer Quelldatenbankparameter mit den Parametern der Ziel-Engine. Für die Ziel-Engine zeigt DMS Fleet Advisor die geschätzten monatlichen Kosten für Ihre Cloud-Ressourcen an. DMS Fleet Advisor verwendet die AWS-Preisliste-Abfrage-API, um Preisdetails nur zu Ihrer Information bereitzustellen. Ihre tatsächlichen Gebühren hängen von einer Vielzahl von Faktoren ab, einschließlich Ihrer tatsächlichen Nutzung von AWS-Services. Weitere Informationen zu den AWS-Service-Preisen finden Sie unter <https://aws.amazon.com/pricing/Cloud-Services> – Preise.
9. Sehen Sie sich im Abschnitt Migrationseinschränkungen die Migrationseinschränkungen an. Wir empfehlen, diese Einschränkungen zu berücksichtigen, wenn Sie Ihre Quelldatenbank zur AWS Cloud migrieren.

Exportieren von Zielempfehlungen mit AWS DMS Fleet Advisor

Nachdem Sie Zielempfehlungen generiert haben, können Sie eine Kopie der Empfehlungsliste als CSV-Datei speichern.

So generieren Sie Zielempfehlungen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.

Stellen Sie sicher, dass Sie die AWS-Region ausgewählt haben, in der Sie den DMS Fleet Advisor verwenden.

2. Wählen Sie im Navigationsbereich Empfehlungen unter Bewerten und dann die Empfehlungen, die Sie in Ihre CSV-Datei aufnehmen möchten.
3. Wählen Sie Zu CSV exportieren, geben Sie den Dateinamen ein und wählen Sie den Ordner auf Ihrem PC aus, in dem diese Datei gespeichert werden soll.
4. Öffnen Sie die CSV-Datei.

Die CSV-Datei mit Empfehlungen enthält die folgenden Informationen.

- CreatedDate – Das Datum, an dem DMS Fleet Advisor die Ziel-Engine-Empfehlung erstellt hat.

- **DatabaseId** – Die Kennung der Quelldatenbank, für die DMS Fleet Advisor diese Empfehlung erstellt hat.
- **DeploymentOption** – Die Bereitstellungsoption für die empfohlene Amazon RDS-DB-Instance.
- **EngineEdition** – Die empfohlene Amazon RDS-Engine-Zieledition.
- **EngineName** – Der Name der Ziel-Engine.
- **InstanceMemory** – Die Speichermenge auf der empfohlenen Amazon RDS-DB-Instance.
- **InstanceSizingType** – Die Größe Ihrer Ziel-Instance.
- **InstanceType** – Der empfohlene Amazon-RDS-Ziel-Instance-Typ.
- **InstanceVcpu** – Die Anzahl der virtuellen CPUs auf der empfohlenen Amazon RDS-DB-Instance.
- **Bevorzugt** – Eine boolesches Markierung, die angibt, dass diese Zieloption empfohlen wird.
- **Status** – Der Status der Empfehlung für die Ziel-Engine.
- **Storagelops** – Die Anzahl der pro Sekunde abgeschlossenen E/A-Operationen (IOPS) auf der empfohlenen Amazon-RDS-DB-Instance.
- **StorageSize** – Die Speichergröße der empfohlenen Amazon RDS-DB-Instance.
- **StorageType** – Der Speichertyp der empfohlenen Amazon RDS-DB-Instance.
- **WorkloadType** – Die Bereitstellungsoption für Ihre Ziel-Engine, z. B. Multi-AZ- oder Single-AZ-Bereitstellung.

Erkennen und Analysieren von Migrationsbeschränkungen mit AWS DMS Fleet Advisor

Sie können den DMS-Datenkollektor verwenden, um Datenbank-Features zu ermitteln, die Ihre Ziel-Engine nicht unterstützt. Bei der Auswahl des korrekten Migrationsziels sollten Sie diese Einschränkungen berücksichtigen.

Der DMS-Datenkollektor erkennt bestimmte Funktionen der Quelldatenbank. Anschließend analysiert DMS Fleet Advisor Quellfunktionen von einem Migrationsstandpunkt zum angegebenen Ziel und stellt zusätzliche Informationen über die Einschränkung bereit und enthält empfohlene Maßnahmen zur Behebung oder Vermeidung dieser Einschränkung. Außerdem berechnet DMS Fleet Advisor die Auswirkungen dieser Einschränkungen.

Die Liste der Einschränkungen ist auf der Detailseite der Ziel-Engine verfügbar. Navigieren Sie von der Seite Empfehlungen im linken Navigationsmenü zu dieser Seite. Wählen Sie aus der Liste der

Ziele die zu untersuchende Ziel-Engine aus. Die Liste der Einschränkungen befindet sich unten auf der Seite.

Die folgende Tabelle enthält MySQL-Datenbank-Features, die Amazon RDS für MySQL nicht unterstützt.

Einschränkung	Beschreibung	Auswirkung
Authentifizierungs-Plugins	Amazon RDS unterstützt keine MySQL-Authentifizierungs-Plugins.	Niedrig
Fehlerprotokollierung im Systemprotokoll	Amazon RDS unterstützt nicht das Schreiben des Fehlerprotokolls in das Systemprotokoll.	Niedrig
Globale Transaktionskennungen	Sie können globale Transaktionskennungen mit allen Versionen von RDS für MySQL 5.7 und RDS für MySQL 8.0.26 und höheren MySQL-8.0-Versionen verwenden.	Niedrig
Gruppenreplikation	Amazon RDS unterstützt nicht das MySQL-Gruppenreplikations-Plugin.	Niedrig
InnoDB-Tablespace-Verschlüsselung	Amazon RDS unterstützt nicht die InnoDB-Tablespace-Verschlüsselung.	Niedrig
Reserviertes Wort InnoDB	InnoDB ist ein reserviertes Wort für Amazon RDS für MySQL. Sie können diesen Namen für eine MySQL-Datenbank nicht verwenden.	Niedrig

Einschränkung	Beschreibung	Auswirkung
Schlüsselbund-Plugin	Amazon RDS unterstützt nicht das MySQL-Schlüsselbund-Plugin.	Niedrig
Plugin für die Passwortvalidierung	Amazon RDS unterstützt nicht das MySQL-validate_password -Plugin.	Niedrig
Persistente Systemvariablen	Amazon RDS unterstützt keine persistenten MySQL-Systemvariablen.	Niedrig
Eingeschränkter Zugriff	Amazon RDS schränkt den Zugriff auf einige Systemprozeduren und Tabellen ein, die erweiterte Berechtigungen erfordern. Amazon RDS erlaubt dazu keinen direkten Hostzugriff auf eine DB-Instanz über Telnet, Secure Shell (SSH) oder Windows Remote Desktop Connection.	Niedrig
Rewriter-Plugin für das Umschreiben von Abfragen	Amazon RDS unterstützt nicht das MySQL-Rewriter-Plugin für das Umschreiben von Abfragen.	Niedrig
Semi-synchrone Replikation	Amazon RDS unterstützt nicht die semisynchrone MySQL-Replikation.	Niedrig
Transportierbare Tablespaces	Amazon RDS unterstützt nicht das MySQL-Feature Transportable Tablespaces.	Niedrig

Einschränkung	Beschreibung	Auswirkung
X-Plugin	Amazon RDS unterstützt nicht das MySQL-X-Plugin.	Niedrig

Die folgende Tabelle enthält MySQL-Datenbank-Features, die Amazon RDS für Oracle nicht unterstützt.

Einschränkung	Beschreibung	Auswirkung
Active Data Guard	Sie können Active Data Guard nicht mit Oracle-Multitenant-Container-Datenbanken (CDB) verwenden.	Mittelschwer
Automatic Storage Management	Amazon RDS unterstützt nicht Oracle Automatic Storage Management (Oracle ASM).	Mittelschwer
Datenbankaktivitäts-Streams	Amazon RDS unterstützt keine Oracle Database Activity Streams für die Single-Tenant-Architektur.	Hoch
Einschränkungen der Dateigröße	Die maximale Dateigröße auf DB-Instances von RDS für Oracle beträgt 16 TiB.	Mittelschwer
FTP und SFTP	Amazon RDS unterstützt weder FTP noch SFTP.	Mittelschwer
Hybride partitionierte Tabellen	Amazon RDS unterstützt keine hybriden partitionierten Oracle-Tabellen.	Hoch

Einschränkung	Beschreibung	Auswirkung
Oracle Data Guard	Amazon RDS unterstützt nicht Oracle Data Guard für die Single-Tenant-Architektur.	Hoch
Oracle Database Vault	Amazon RDS unterstützt nicht Oracle Database Vault.	Hoch
Oracle DBA-Berechtigungen	Amazon RDS hat Einschränkungen für Oracle-DBA-Berechtigungen. Weitere Informationen finden Sie unter Einschränkungen für Oracle-DBA-Berechtigungen .	Hoch
Oracle Enterprise Manager	Amazon RDS unterstützt nicht Oracle Enterprise Manager für die Single-Tenant-Architektur.	Hoch
Oracle Enterprise Manager Agent	Amazon RDS unterstützt nicht Oracle Enterprise Manager Agent für die Single-Tenant-Architektur.	Mittelschwer
Oracle Enterprise Manager Cloud Control Management Repository	Sie können keine DB-Instance von Amazon RDS für Oracle für das Oracle Enterprise Manager Cloud Control Management Repository verwenden.	Hoch
Oracle Flashback Database	Amazon RDS Oracle unterstützt nicht das Feature Oracle Flashback Database.	Hoch

Einschränkung	Beschreibung	Auswirkung
Oracle Label Security	Amazon RDS unterstützt nicht Oracle Data Security für die Single-Tenant-Architektur. Sie können Oracle Label Security nur mit Multitenant-Container-Datenbanken (Oracle CDB) verwenden.	Hoch
Oracle Messaging Gateway	Amazon RDS unterstützt nicht Oracle Messaging Gateway.	Hoch
Oracle Real Application Clusters	Amazon RDS unterstützt nicht Oracle Real Application Clusters (Oracle RAC).	Hoch
Oracle Real Application Testing	Amazon RDS unterstützt nicht Oracle Real Application Testing.	Hoch
Oracle-Snapshot-Standby-Datenbanken	Amazon RDS unterstützt keine Oracle-Snapshot-Standby-Datenbanken.	Hoch
Öffentliche Synonyme	Amazon RDS unterstützt keine öffentlichen Synonyme für von Oracle bereitgestellte Schemata.	Mittelschwer
Schematas für nicht unterstützte Funktionen	Amazon RDS unterstützt keine Schemata für Oracle-Features und -Komponenten, für die Systemberechtigungen erforderlich sind.	Hoch

Einschränkung	Beschreibung	Auswirkung
Reines einheitliches Auditing	Amazon RDS unterstützt kein reines einheitliches Auditing. Sie können das einheitliche Auditing im gemischten Modus verwenden.	Mittelschwer
Workspace Manager	Amazon RDS unterstützt nicht das WMSYS-Schema von Oracle Database Workspace Manager.	Hoch

Die folgende Tabelle enthält MySQL-Datenbank-Features, die Amazon RDS für PostgreSQL nicht unterstützt.

Einschränkung	Beschreibung	Auswirkung
Gleichzeitige Verbindungen	Die maximale Anzahl gleichzeitiger Verbindungen zu Ihrer RDS-für-PostgreSQL-Instance wird durch den Parameter <code>max_connections</code> begrenzt.	Niedrig
Neueste Versionen	Amazon RDS wendet Hauptversions-Upgrades nicht automatisch an. Um ein Hauptversions-Upgrade durchzuführen, ändern Sie Ihre DB-Instance manuell. Weitere Informationen finden Sie unter Auswählen eines Hauptversions-Upgrades für PostgreSQL .	Niedrig

Einschränkung	Beschreibung	Auswirkung
Reservierte Verbindungen	Amazon RDS reserviert bis zu 3 Verbindungen für die Systemwartung. Wenn Sie einen Wert für die Benutzerverbindungsparameter festlegen, fügen Sie der voraussichtlich verwendeten Anzahl an Verbindungen drei hinzu.	Niedrig
Unterstützte Erweiterungen	RDS für PostgreSQL unterstützt eine begrenzte Zahl von Erweiterungen für die PostgreSQL-Datenbank-Engine. Sie finden eine Liste der unterstützten Erweiterungen in der Standard-DB-Parametergruppe für Ihre PostgreSQL-Version. Sie können sich auch die Liste der aktuellen Erweiterungen mit <code>psql</code> verwenden, indem Sie den Parameter <code>rds.extensions</code> anzeigen.	Niedrig
Aufteilung oder Isolierung von Tablespaces	Sie können keine Tablespaces für E/A-Aufteilung oder -Isolierung verwenden. In RDS für PostgreSQL befindet sich der gesamte Speicher auf einem einzigen logischen Volume.	Niedrig

Die folgende Tabelle enthält SQL-Server-Datenbank-Features, die Amazon RDS für SQL Server nicht unterstützt.

Einschränkung	Beschreibung	Auswirkung
Sichern in Microsoft Azure Blob Storage	RDS für SQL Server unterstützt nicht die Sicherung auf Microsoft Azure Blob Storage.	Mittelschwer
Erweiterung des Puffer-Pools	RDS für SQL Server unterstützt nicht die Pufferpool-Erweiterung.	Hoch
Richtlinien für benutzerdefinierte Kennwörter	RDS für SQL Server unterstützt keine benutzerdefinierten Passwortrichtlinien.	Mittelschwer
Data Quality Services	RDS für SQL Server unterstützt nicht die SQL Server Data Quality Services (DQS).	Hoch
Datenbank-Protokollversand	RDS für SQL Server unterstützt nicht das Versenden von Datenbankprotokollen.	Hoch
Datenbanknamen	Datenbanknamen haben die folgenden Einschränkungen: Sie dürfen nicht mit „rdsadmin“ beginnen, sie dürfen nicht mit einem Leerzeichen oder einem Tabulator beginnen oder enden, sie dürfen keines der Zeichen enthalten, die eine neue Zeile erzeugen, und sie dürfen kein einzelnes Anführungszeichen (') enthalten.	Mittelschwer

Einschränkung	Beschreibung	Auswirkung
Datenbank-Snapshots	RDS für SQL Server unterstützt keine Datenbank-Snapshots. Sie können in Amazon RDS nur DB-Instance-Snapshots verwenden.	Mittelschwer
Erweiterte gespeicherte Prozeduren	RDS für SQL Server unterstützt keine erweiterten gespeicherten Prozeduren, einschließlich <code>xp_cmdshell</code> .	Hoch
Datenbanktabellen	RDS für SQL Server unterstützt keine Dateitabellen.	Mittelschwer
FILESTREAM-Unterstützung	RDS für SQL Server bietet keine FILESTREAM-Unterstützung.	Mittelschwer
Verknüpfte Server	RDS für SQL Server bietet eingeschränkte Unterstützung für verknüpfte Server.	Hoch
Machine-Learning- und R-Services	RDS für SQL Server unterstützt keine Machine-Learning- und R-Services, da Sie Betriebssystemzugriff benötigen, um diese Services zu installieren.	Hoch
Wartungspläne	RDS für SQL Server unterstützt keine Wartungspläne.	Hoch
Performance-Datenaufbilder	RDS für SQL Server unterstützt nicht den Performance Data Collector.	Hoch

Einschränkung	Beschreibung	Auswirkung
Richtlinienbasierte Verwaltung	RDS für SQL Server unterstützt keine richtlinienbasierte Verwaltung.	Mittelschwer
PolyBase	RDS für SQL Server unterstützt nicht PolyBase.	Hoch
Replikation	RDS für SQL Server unterstützt keine Replikation.	Mittelschwer
Ressourcenkontrolle	RDS für SQL Server unterstützt nicht Resource Governor.	Hoch
Auslöser auf Serverebene	RDS für SQL Server unterstützt keine Auslöser auf Serverebene.	Mittelschwer
Service Broker-Endpunkte	RDS für SQL Server unterstützt keine Service-Broker-Endpunkte.	Hoch
SSAS	Beachten Sie die Einschränkungen, die für die Ausführung von SQL Server Analysis Services (SSAS) auf RDS für SQL Server gelten. Weitere Informationen finden Sie unter Limitations .	Niedrig
SSIS	Beachten Sie die Einschränkungen, die für die Ausführung von SQL Server Integration Services (SSIS) auf RDS für SQL Server gelten. Weitere Informationen finden Sie unter Limitations .	Niedrig

Einschränkung	Beschreibung	Auswirkung
SSRS	Beachten Sie die Einschränkungen, die für die Ausführung von SQL Server Reporting Services (SSRS) auf RDS für SQL Server gelten. Weitere Informationen finden Sie unter Limitations .	Niedrig
Speichergröße für SQL-Server-DB-Instances	Die maximale Speichergröße für SQL-Server-General-Purpose (SSD)-Speicher und bereitgestellte IOPS-Speicherinstances beträgt 16 TiB. Die maximale Speichergröße für SQL-Server-Magnetic-Storage-Instances beträgt 1 TiB.	Hoch
Stretch-Datenbank	RDS für SQL Server unterstützt nicht das Feature SQL Server Stretch Database.	Mittelschwer
T-SQL-Endpunkte	RDS für SQL Server unterstützt nicht alle Vorgänge, die CREATE ENDPOINT verwenden.	Hoch
Datenbankeigenschaft TRUSTWORTHY	RDS für SQL Server unterstützt nicht die Datenbankeigenschaft TRUSTWORTHY, da diese die sysadmin-Rolle erfordert.	Mittelschwer

Die folgende Tabelle enthält eine Liste von Empfehlungsproblemen. DMS Fleet Advisor analysiert Quell- und Zieldatenbankfunktionen und stellt diese Migrationsbeschränkungen bereit. Die Einschränkung der Blocker-Auswirkung bedeutet, dass DMS Fleet Advisor keine Zielempfehlungen für die Quelldatenbank generieren kann.

Einschränkung	Beschreibung	Auswirkung
Die entsprechende Instance wurde nicht gefunden	AWS DMS kann keine Ziel-Instance finden, die als richtiges Migrationsziel für eine Kombination Ihrer Quelldatenbankmetriken funktionieren kann.	Blocker
Die entsprechende Instance wird von IOPS nicht gefunden	Die Quelldatenbank verwendet eine Reihe von IOPS, was die maximale Anzahl von IOPS für die möglichen Ziel-DB-Instances überschreitet.	Blocker
Die entsprechende Instance wird vom RAM nicht gefunden	Die Quelldatenbank verwendet eine Anzahl von GB RAM, die die maximale Größe des RAM für die möglichen Ziel-DB-Instances überschreitet.	Blocker
Die richtige Instance wird nach Speichergröße nicht gefunden	Die Quelldatenbank verwendet eine Reihe von TB Speicher, was die maximale Speichergröße für die möglichen Ziel-DB-Instances überschreitet.	Blocker
Die richtige Instance wird von der Edition nicht gefunden	Die Quelldatenbank verfügt über eine Edition, die von Amazon RDS nicht unterstützt wird.	Blocker

Einschränkung	Beschreibung	Auswirkung
Die richtige Instance wird von CPU-Kernen nicht gefunden	Die Quelldatenbank verfügt über eine Reihe von CPU-Kernen, die die maximale Anzahl von CPU-Kernen für die möglichen Ziel-DB-Instances überschreiten.	Blocker
Die entsprechende Instance wird von der -Version nicht gefunden	Ihre Quelldatenbank hat die Version , die AWS DMS nicht erkennt.	Blocker
CPU-Parameter ist nicht definiert	Der DMS-Datenkollektor hat keine Informationen über die CPU erfasst, die Ihre Quelldatenbank verwendet. Stellen Sie sicher, dass Sie die erforderlichen Metriken und konfigurierten Anmeldeinformationen für die Datenweiterleitung in Ihrem Datenkollektor erfasst haben. Siehe Konfiguration von Anmeldeinformationen für die Datenweiterleitung .	Blocker

Einschränkung	Beschreibung	Auswirkung
Der Speicherparameter ist nicht definiert	Der DMS-Datenkollektor hat keine Informationen über den Speicher erfasst, den Ihre Quelldatenbank verwendet. Stellen Sie sicher, dass Sie die erforderlichen Metriken und konfigurierten Anmeldeinformationen für die Datenweiterleitung in Ihrem Datenkollektor erfasst haben. Siehe Konfiguration von Anmeldeinformationen für die Datenweiterleitung .	Blocker
Speichergrößenparameter ist nicht definiert	Der DMS-Datenkollektor hat keine Informationen über die Speichergröße erfasst, die Ihre Quelldatenbank verwendet. Stellen Sie sicher, dass Sie die erforderlichen Metriken und konfigurierten Anmeldeinformationen für die Datenweiterleitung in Ihrem Datenkollektor erfasst haben. Siehe Konfiguration von Anmeldeinformationen für die Datenweiterleitung .	Blocker

Einschränkung	Beschreibung	Auswirkung
Speicher-IOPS-Parameter ist nicht definiert	Der DMS-Datenkollektor hat die Speicher-IOPS-Metriken für Ihre Quelldatenbank nicht erfasst. Stellen Sie sicher, dass Sie die erforderlichen Metriken und konfigurierten Anmeldeinformationen für die Datenweiterleitung in Ihrem Datenkollektor erfasst haben.	Blocker
Nicht genügend Daten	Der DMS-Datenkollektor hat nicht genügend Daten gesammelt, um eine Zielempfehlung zu generieren. Stellen Sie sicher, dass Sie Anmeldeinformationen für die Datenweiterleitung in Ihrem Datenkollektor konfiguriert haben. Siehe Konfiguration von Anmeldeinformationen für die Datenweiterleitung .	Blocker
Database Edition ist nicht definiert	Der DMS-Datenkollektor hat keine Informationen über Ihre Quelldatenbank-Edition erfasst. Stellen Sie sicher, dass Sie die erforderlichen Metriken und konfigurierten Anmeldeinformationen für die Datenweiterleitung in Ihrem Datenkollektor erfasst haben. Siehe Konfiguration von Anmeldeinformationen für die Datenweiterleitung .	Blocker

Einschränkung	Beschreibung	Auswirkung
Unbekannter Fehler	DMS Fleet Advisor kann keine Zielempfehlungen für Ihre Quelldatenbank generieren.	Blocker
Die Datenbankversion ist nicht definiert	Der DMS Fleet Advisor hat keine Informationen über Ihre Quelldatenbankversion erfasst. DMS Fleet Advisor empfiehlt, die neueste Datenbankversion für Ihre Quelldatenbank zu verwenden. Wenn Sie diese Empfehlung wählen, müssen Sie Ihre Datenbankversion aktualisieren. Überprüfen Sie die generierten Zielempfehlungen für Ihre Quelldatenbank und stellen Sie sicher, dass diese Empfehlungen Ihren Anforderungen entsprechen.	Hoch
Erhöhen der Anzahl der Datenbankverbindungen in RDS-Einstellungen	Ihre Quelldatenbank erfordert eine bestimmte Anzahl von Verbindungen. Standardmäßig ist die Anzahl der verfügbaren Verbindungen für Amazon-RDS-Datenbank-Instanzen unterschiedlich. Stellen Sie sicher, dass Sie diesen Standardwert ändern, wenn Sie Ihre RDS-Datenbank-Instanz erstellen. Aktualisieren Sie dazu den <code>max_connections</code> Parameterwert in Parametergruppen.	Mittelschwer

Einschränkung	Beschreibung	Auswirkung
Die Ziel-Edition ist kompatibel	Die Zielempfehlung für Ihre Quelldatenbank verwendet eine andere Datenbank-Edition. Ihre Quelldatenbank-Edition unterstützt dieselben Funktionen wie die empfohlene Ziel-Edition. Die Auswahl dieser neuen Datenbank-Edition kann jedoch Ihre Kosten erhöhen.	Mittelschwer
Der Speicherdurchsatzparameter ist nicht definiert	Der DMS-Datenkollektor hat die Speicherdurchsatzmetriken für Ihre Quelldatenbank nicht erfasst. Überprüfen Sie die generierten Zielempfehlungen für Ihre Quelldatenbank und stellen Sie sicher, dass diese Empfehlungen Ihren Anforderungen entsprechen.	Mittelschwer
Der Parameter für die Datenbankverbindungsnummer ist nicht definiert	Der DMS-Datenkollektor hat keine Informationen über die Anzahl der Verbindungen erfasst, die Ihre Quelldatenbank verwendet. Überprüfen Sie die generierten Zielempfehlungen für Ihre Quelldatenbank und stellen Sie sicher, dass diese Empfehlungen Ihren Anforderungen entsprechen. Alternativ können Sie eine Kontingenterhöhung beantragen.	Mittelschwer

Einschränkung	Beschreibung	Auswirkung
Version für Datenbank-Downgrade	Ihre Quelldatenbank wird auf einer höheren Version als die Amazon-RDS-Datenbank ausgeführt. Um Ihre Datenbankversion herunterzustufen, stellen Sie sicher, dass Sie nicht die Funktionen verwenden, die nicht in einer niedrigeren Version implementiert sind. Alternativ können Sie Amazon EC2 als Migrationziel verwenden.	Mittelschwer
Die Ziel-Edition ist anders	Die Zielempfehlung für Ihre Quelldatenbank verwendet eine andere Datenbank-Edition. Ihre Quelldatenbank-Edition ist mit der empfohlenen Ziel-Edition kompatibel. Die empfohlene Zieldatenbank-Edition unterstützt jedoch einige Funktionen Ihrer Quelldatenbank-Edition nicht. Die Wahl dieser neuen Datenbank-Edition kann Ihre Ausgaben erhöhen.	Mittelschwer

Einschränkung	Beschreibung	Auswirkung
Upgrade von einer nicht unterstützten Version	<p>Ihre Quelldatenbank hat das Ende der Supportphase erreicht. Um die neueste DB-Engine-Version als Ziel zu verwenden, aktualisieren Sie Ihre Datenbank vor der Migration. Alternativ können Sie Amazon EC2 als Migrationziel verwenden.</p> <p>Verwenden Sie je nach Datenbank-Engine einen der folgenden Links für Weitere Informationen:</p> <p>Aktualisieren von MySQL</p> <p>SQL Server aktualisieren</p> <p>OracleDB aktualisieren</p> <p>Aktualisieren von PostgreSQL</p>	Mittelschwer

Fehlerbehebung für Zielempfehlungen

In der folgenden Liste finden Sie Maßnahmen, die Sie ergreifen können, wenn Sie auf Probleme mit dem Zielempfehlungs-Feature von DMS Fleet Advisor stoßen.

Themen

- [Ich kann keine Preisschätzungen für Zielempfehlungen](#)
- [Ich kann keine Diagramme zur Ressourcennutzung sehen](#)
- [Ich kann den Status der Metrikerfassung nicht sehen](#)

Ich kann keine Preisschätzungen für Zielempfehlungen

Wenn Sie für eine Empfehlung mit dem Status Erfolgreich Keine Daten zu den geschätzten monatlichen Kosten sehen, stellen Sie sicher, dass Sie Ihrem IAM-Benutzer die Zugriffsrechte für die AWS-Preisliste-Service-API erteilt haben. Dazu müssen Sie die Richtlinie erstellen, die die `pricing:GetProducts`-Berechtigung enthält, und sie Ihrem IAM-Benutzer hinzufügen, wie unter [Erstellen von IAM-Ressourcen](#) beschrieben.

DMS Fleet Advisor berechnet nicht die geschätzten monatlichen Kosten für Empfehlungen mit dem Status Fehlgeschlagen.

Ich kann keine Diagramme zur Ressourcennutzung sehen

Wenn Sie die Meldung Metrik konnte nicht geladen werden sehen, nachdem Sie den Abschnitt Quellauslastung und -kapazität erweitert haben, stellen Sie sicher, dass Sie Ihrem IAM-Benutzer die Berechtigungen zum Anzeigen von Amazon- CloudWatch Dashboards erteilt haben. Dazu müssen Sie Ihrem IAM-Benutzer die erforderliche Richtlinie hinzufügen, wie unter [Erstellen von IAM-Ressourcen](#) beschrieben.

Alternativ können Sie eine benutzerdefinierte Richtlinie erstellen, die die Berechtigungen `cloudwatch:GetDashboard`, `cloudwatch:ListDashboards`, `cloudwatch:PutDashboard` und `cloudwatch:DeleteDashboards` umfasst. Weitere Informationen finden Sie unter [Verwenden von Amazon- CloudWatch Dashboards](#) im Amazon- CloudWatch Benutzerhandbuch.

Ich kann den Status der Metrikerfassung nicht sehen

Wenn Sie bei der Auswahl von Empfehlungen generieren Keine Daten verfügbar für Erfassung von Metriken sehen, stellen Sie sicher, dass Sie Daten erfasst haben. Weitere Informationen finden Sie unter [Erfassen von Daten für AWS DMS Fleet Advisor](#).

Wenn dieses Problem auftritt, nachdem Sie Daten erfasst haben, stellen Sie sicher, dass Sie Ihrem IAM-Benutzer die `cloudwatch:Get*` Berechtigung für den Zugriff auf Amazon erteilt haben CloudWatch. DMS Fleet Advisor verwendet eine serviceverknüpfte Rolle, um die erfassten Datenbankleistungsmetriken CloudWatch in in Ihrem Namen zu veröffentlichen. Stellen Sie sicher, dass Sie eine serviceverknüpfte Rolle für die Verwendung mit DMS Fleet Advisor erstellen. Weitere Informationen finden Sie unter [Erstellen von IAM-Ressourcen](#).

Einschränkungen für DMS Fleet Advisor

Zu den Einschränkungen bei der Verwendung von DMS Fleet Advisor gehören die folgenden:

- DMS Fleet Advisor generiert one-to-one Empfehlungen. Für jede Quelldatenbank bestimmt DMS Fleet Advisor eine einzelne Ziel-Engine. DMS Fleet Advisor unterstützt keine Multitenant-Server und gibt keine Empfehlungen für den Betrieb mehrerer Datenbanken auf einer einzigen Ziel-DB-Instance.
- DMS Fleet Advisor gibt keine Empfehlungen zu verfügbaren Datenbankversions-Upgrades.
- DMS Fleet Advisor kann Empfehlungen für bis zu 100 Datenbanken gleichzeitig generieren.
- Wenn Sie den DMS-Datenkollektor installieren, bei dem es sich um eine Windows-Anwendung handelt, stellen Sie sicher, dass Sie auch .NET Framework 4.8 und PowerShell 6.0 und höher installieren. Informationen zu den Hardwareanforderungen finden Sie unter [Installieren eines Datenkollektors](#).
- Der DMS-Datenkollektor benötigt Berechtigungen, um Anfragen mithilfe des LDAP-Protokolls auf Ihrem Domain-Server auszuführen.
- Für den DMS-Datenkollektor ist das sudo-SSH-Skript erforderlich, das unter Linux ausgeführt wird.
- Der DMS-Datenkollektor benötigt Berechtigungen zum Ausführen von Remote- PowerShell, Windows Management Instrumentation (WMI), WMI Query Language (WQL) und Registrierungsskripten in Windows.
- Bei MySQL und PostgreSQL kann DMS Fleet Advisor Leistungskennzahlen nicht direkt aus der Datenbank erfassen. Stattdessen erfasst DMS Fleet Advisor die Betriebssystem-Servermetriken. Daher können Sie keine Empfehlungen auf der Grundlage von Nutzungsmetriken für MySQL- und PostgreSQL-Datenbanken generieren, die auf Amazon RDS und Aurora ausgeführt werden.

Konvertieren von Datenbankschemata mithilfe von DMS Schema Conversion

DMS Schema Conversion in AWS Database Migration Service (AWS DMS) macht Datenbankmigrationen zwischen verschiedenen Datenbanktypen vorhersehbarer. Mithilfe der DMS Schema Conversion können Sie die Komplexität Ihrer Migration für Ihren Quelldatenanbieter bewerten und anschließend Datenbankschemata und Codeobjekte konvertieren. Sie können dann den konvertierten Code auf Ihre Zieldatenbank anwenden.

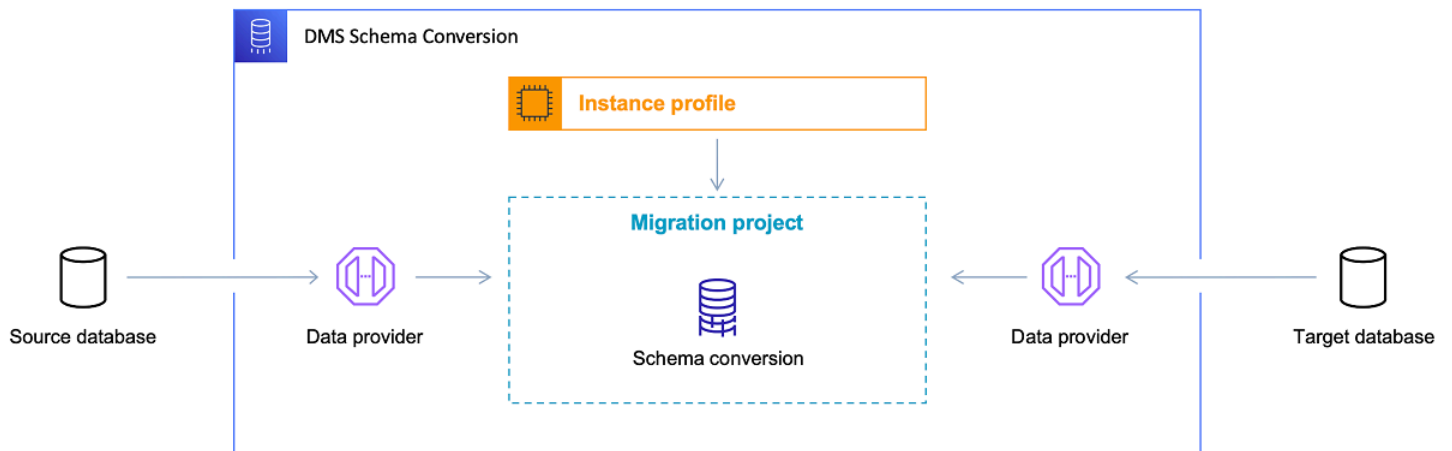
DMS Schema Conversion konvertiert Ihre Quelldatenbankschemata und die meisten Datenbankcodeobjekte automatisch in ein mit der Zieldatenbank kompatibles Format. Diese Konvertierung umfasst Tabellen, Ansichten, gespeicherte Prozeduren, Funktionen, Datentypen, Synonyme usw. Alle Objekte, die DMS Schema Conversion nicht automatisch konvertieren kann, sind deutlich gekennzeichnet. Um die Migration abzuschließen, können Sie diese Objekte manuell konvertieren.

Allgemein arbeitet [DMS Schema Conversion](#) mit den folgenden drei Komponenten: Instance-Profile, Datenanbieter und Migrationsprojekte. Ein Instance-Profil spezifiziert Netzwerk- und Sicherheitseinstellungen. Ein Datenanbieter speichert Anmeldeinformationen für Datenbankverbindungen. Ein Migrationsprojekt enthält Datenanbieter, ein Instance-Profil und Migrationsregeln. AWS DMS verwendet Datenanbieter und ein Instance-Profil, um einen Prozess zu entwerfen, der Datenbankschemata und Codeobjekte konvertiert.

Eine Liste der unterstützten Quelldatenbanken finden Sie unter [Quellen für DMS Schema Conversion](#).

Eine Liste der unterstützten Zieldatenbanken finden Sie unter [Ziele für DMS Schema Conversion](#).

Das folgende Diagramm illustriert den Prozess von DMS Schema Conversion:



Informieren Sie sich in den folgenden Themen, wie Sie die DMS Schema Conversion verwenden können.

Themen

- [Unterstützt AWS-Regionen](#)
- [Features für die Schemakonvertierung](#)
- [Einschränkungen für die Schemakonvertierung](#)
- [Erste Schritte mit DMS Schema Conversion](#)
- [Einrichten eines Netzwerks für DMS Schema Conversion](#)
- [Erstellen von Quelldatenanbietern in DMS Schema Conversion](#)
- [Erstellen von Zieldatenanbietern in DMS Schema Conversion](#)
- [Verwalten von Migrationsprojekten in DMS Schema Conversion](#)
- [Erstellung von Bewertungsberichten zur Datenbankmigration mit DMS Schema Conversion](#)
- [Verwenden von DMS Schema Conversion](#)
- [Verwenden von Erweiterungspaketen bei DMS Schema Conversion](#)

Unterstützt AWS-Regionen

Sie können ein Migrationsprojekt für DMS Schema Conversion in den folgenden erstellen AWS-Regionen. In anderen Regionen können Sie AWS Schema Conversion Tool verwenden. Weitere Informationen zu AWS SCT finden Sie im [AWS Benutzerhandbuch zum Schema Conversion Tool](#).

Name der Region	Region
USA Ost (Nord-Virginia)	us-east-1
USA Ost (Ohio)	us-east-2
USA West (Oregon)	us-west-2
Asien-Pazifik (Tokio)	ap-northeast-1
Asien-Pazifik (Singapur)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Europa (Frankfurt)	eu-central-1
Europa (Stockholm)	eu-north-1
Europa (Irland)	eu-west-1

Features für die Schemakonvertierung

DMS Schema Conversion bietet die folgenden Features:

- DMS Schema Conversion verwaltet automatisch die AWS Cloud Ressourcen, die für Ihr Datenbankmigrationsprojekt erforderlich sind. Zu diesen Ressourcen gehören Instance-Profile, Datenanbieter und AWS Secrets Manager Secrets. Dazu gehören auch AWS Identity and Access Management (IAM)-Rollen, Amazon S3-Buckets und Migrationsprojekte.
- Sie können DMS Schema Conversion verwenden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, die Metadaten zu lesen und Bewertungsberichte zur Datenbankmigration zu erstellen. Anschließend können Sie den Bericht in einem Amazon-S3-Bucket speichern. Dieser Bericht fasst alle Aufgaben der Schemakonvertierung zusammen und führt detailliert die Elemente auf, die DMS Schema Conversion nicht automatisch in die Zieldatenbank konvertieren kann. Anhand der Bewertungsberichte zur Datenbankmigration können Sie beurteilen, wie viel von Ihrem Migrationsprojekt mit DMS Schema Conversion automatisiert werden kann. Außerdem helfen diese Berichte dabei, den Umfang des manuellen Aufwands abzuschätzen, der für den Abschluss der Konvertierung erforderlich ist. Weitere Informationen finden Sie unter [Erstellung von Bewertungsberichten zur Datenbankmigration mit DMS Schema Conversion](#).

- Nachdem Sie eine Verbindung zu Ihren Quell- und Zieldatenanbietern hergestellt haben, kann DMS Schema Conversion Ihre vorhandenen Quelldatenbankschemata in die Zieldatenbank-Engine konvertieren. Sie können beliebige Schemaelemente aus der Quelldatenbank für die Konvertierung auswählen. Nachdem Sie Ihren Datenbankcode in DMS Schema Conversion konvertiert haben, können Sie Ihren Quellcode und den konvertierten Code überprüfen. Sie können den konvertierten SQL-Code auch in einem Amazon-S3-Bucket speichern.
- Bevor Sie Ihre Quelldatenbankschemata konvertieren, können Sie Transformationsregeln einrichten. Sie können Transformationsregeln festlegen, um den Datentyp von Spalten zu ändern, Objekte von einem Schema in ein anderes zu verschieben und Objektnamen zu bearbeiten. Sie können Transformationsregeln auf Datenbanken, Schemata, Tabellen und Spalten anwenden. Weitere Informationen finden Sie unter [Einrichten von Transformationsregeln](#).
- Sie können die Konvertierungseinstellungen ändern, um die Leistung des konvertierten Codes zu verbessern. Diese Einstellungen sind für jedes Konvertierungspaar spezifisch und hängen von den Features der Quelldatenbank ab, die Sie in Ihrem Code verwenden. Weitere Informationen finden Sie unter [Angaben von Einstellungen für die Schemakonvertierung](#).
- In einigen Fällen kann DMS Schema Conversion die Quelldatenbank-Features nicht in gleichwertige Amazon-RDS-Features konvertieren. In diesen Fällen erstellt DMS Schema Conversion ein Erweiterungspaket in Ihrer Zieldatenbank, um die Features zu emulieren, die nicht konvertiert wurden. Weitere Informationen finden Sie unter [Verwenden von Erweiterungspaketen](#).
- Sie können den konvertierten Code und das Erweiterungspaket-Schema auf Ihre Zieldatenbank anwenden. Weitere Informationen finden Sie unter [Anwenden des konvertierten Codes](#).
- DMS Schema Conversion unterstützt alle Funktionen in der neuesten AWS SCT Version. Weitere Informationen finden Sie unter [Die neuesten Versionshinweise für AWS SCT](#).
- Sie können konvertierten SQL-Code bearbeiten, bevor DMS ihn in die Zieldatenbank migriert. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihres konvertierten SQL-Codes](#).

Einschränkungen für die Schemakonvertierung

DMS Schema Conversion ist eine Webversion von AWS Schema Conversion Tool (AWS SCT). DMS Schema Conversion unterstützt weniger Datenbankplattformen und bietet im Vergleich zur AWS SCT -Desktop-Anwendung eingeschränktere Funktionen. Um Data-Warehouse-Schemata, Big-Data-Frameworks, Anwendungs-SQL-Code und ETL-Prozesse zu konvertieren, verwenden Sie AWS SCT. Weitere Informationen zu AWS SCT finden Sie im [AWS Benutzerhandbuch zum Schema Conversion Tool](#).

Die folgenden Einschränkungen gelten, wenn Sie DMS Schema Conversion für die Datenbankschema-Konvertierung verwenden:

- Sie können ein Migrationsprojekt nicht speichern und es im Offline-Modus verwenden.
- Sie können SQL-Code für die Quelle in einem Migrationsprojekt für DMS Schema Conversion nicht bearbeiten. Verwenden Sie Ihren regulären SQL-Editor, um den SQL-Code Ihrer Quelldatenbank zu bearbeiten. Wählen Sie Aus Datenbank aktualisieren, um den aktualisierten Code zu Ihrem Migrationsprojekt hinzuzufügen.
- Die Migrationsregeln in DMS Schema Conversion unterstützen das Ändern der Spaltensortierung nicht. Außerdem können Sie keine Migrationsregeln verwenden, um Objekte in ein neues Schema zu verschieben.
- Sie können keine Filter auf Ihre Quell- und Zieldatenbankbäume anwenden, um nur die Datenbankobjekte anzuzeigen, die der Filterklausel entsprechen.
- Das Erweiterungspaket DMS Schema Conversion enthält keine AWS Lambda Funktionen, die das Senden von E-Mails, die Auftragsplanung und andere Funktionen in Ihrem konvertierten Code emulieren.
- DMS Schema Conversion verwendet keine vom Kunden verwalteten KMS-Schlüssel für den Zugriff auf AWS Kundenressourcen. DMS Schema Conversion unterstützt beispielsweise nicht die Verwendung eines vom Kunden verwalteten KMS-Schlüssels für den Zugriff auf Kundendaten in Amazon S3.

Erste Schritte mit DMS Schema Conversion

Nutzen Sie das folgende Tutorial für die ersten Schritte mit DMS Schema Conversion. In diesem Tutorial erfahren Sie, wie Sie DMS Schema Conversion einrichten, ein Migrationsprojekt erstellen und eine Verbindung mit Ihren Datenanbietern herstellen. Anschließend erfahren Sie, wie Sie die Komplexität Ihrer Migration bewerten und Ihre Quelldatenbank in ein mit Ihrer Zieldatenbank kompatibles Format konvertieren. Darüber hinaus erfahren Sie, wie Sie den konvertierten Code auf Ihre Zieldatenbank anwenden.

Im folgenden Tutorial werden die erforderlichen Aufgaben erläutert und die Konvertierung einer Datenbank von Amazon RDS für SQL Server in Amazon RDS für MySQL demonstriert. Sie können jeden unterstützten Quell- und Zieldatenanbieter verwenden. Weitere Informationen finden Sie unter [Quelldatenanbieter für DMS Schema Conversion](#).

Weitere Informationen zur DMS-Schemakonvertierung finden Sie in den Walkthroughs zur step-by-step Migration von [Oracle zu PostgreSQL](#) und von SQL Server zu [MySQL](#).

[Dieses Video](#) stellt die Benutzeroberfläche von DMS Schema Conversion vor und hilft Ihnen, sich mit den Kernkomponenten dieses Services vertraut zu machen.

Themen

- [Voraussetzungen für die Arbeit mit DMS Schema Conversion](#)
- [Schritt 1: Erstellen eines Instance-Profiles](#)
- [Schritt 2: Konfigurieren Ihrer Datenanbieter](#)
- [Schritt 3: Erstellen eines Migrationsprojekts](#)
- [Schritt 4: Erstellen eines Bewertungsberichts](#)
- [Schritt 5: Konvertieren Ihres Quellcodes](#)
- [Schritt 6: Anwenden des konvertierten Codes](#)
- [Schritt 7: Bereinigen und Fehlersuche](#)

Voraussetzungen für die Arbeit mit DMS Schema Conversion

Führen Sie die folgenden Aufgaben aus, um DMS Schema Conversion einzurichten. Anschließend können Sie ein Instance-Profil einrichten, Datenanbieter hinzufügen und ein Migrationsprojekt erstellen.

Themen

- [Erstellen einer VPC basierend auf Amazon VPC](#)
- [Erstellen eines Amazon-S3-Buckets](#)
- [Speichern Sie Datenbankmeldeinformationen in AWS Secrets Manager](#)
- [Erstellen von IAM-Rollen](#)

Erstellen einer VPC basierend auf Amazon VPC

In diesem Schritt erstellen Sie eine virtuelle private Cloud (VPC) in Ihrem AWS-Konto. Diese VPC basiert auf dem Service Amazon Virtual Private Cloud (Amazon VPC) und enthält Ihre AWS Ressourcen.

So erstellen Sie eine VPC für DMS Schema Conversion

1. Melden Sie sich bei der Amazon VPC-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie VPC erstellen aus.
3. Geben Sie auf der Seite VPC erstellen die folgenden Einstellungen ein:
 - Zu erstellende Ressourcen – VPC und mehr
 - Automatische Generierung des Namens-Tags – Wählen Sie Automatisch generieren aus und geben Sie einen global eindeutigen Namen ein. Geben Sie z. B. ei **sc-vpc**.
 - IPv4 CIDR block (IPv4-CIDR-Block) – **10.0.1.0/24**
 - NAT-Gateways – In 1 AZ
 - VPC endpoints (VPC-Endpunkte) – None (Keine)
4. Behalten Sie die übrigen Einstellungen unverändert bei und wählen Sie dann VPC erstellen aus.
5. Wählen Sie Subnetze aus und notieren Sie sich die öffentlichen und privaten Subnetz-IDs.

Um eine Verbindung mit Ihren Amazon-RDS-Datenbanken herzustellen, erstellen Sie eine Subnetzgruppe, die öffentliche Subnetze enthält.

Um eine Verbindung mit Ihren On-Premises-Datenbanken herzustellen, erstellen Sie eine Subnetzgruppe, die private Subnetze enthält. Weitere Informationen finden Sie unter [Schritt 1: Erstellen eines Instance-Profils](#).

6. Wählen Sie NAT-Gateways aus. Wählen Sie Ihr NAT-Gateway aus und notieren Sie sich Ihre Elastic IP-Adresse.

Konfigurieren Sie Ihr Netzwerk so, dass AWS DMS Sie über die öffentliche IP-Adresse dieses NAT-Gateways auf Ihre lokale Quelldatenbank zugreifen können. Weitere Informationen finden Sie unter [Verwenden einer Internetverbindung zu einer VPC](#).

Verwenden Sie diese VPC, wenn Sie Ihr Instance-Profil und Ihre Zieldatenbanken in Amazon RDS erstellen.

Erstellen eines Amazon-S3-Buckets

Erstellen Sie einen Amazon-S3-Bucket, in dem Informationen zu Ihrem Migrationsprojekt gespeichert werden sollen. DMS Schema Conversion verwendet diesen Amazon-S3-Bucket zum Speichern von

Elementen wie Bewertungsberichten, konvertiertem SQL-Code, Informationen zu Datenbankschema-Objekten usw.

So erstellen Sie einen Amazon-S3-Bucket für DMS Schema Conversion

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Bucket erstellen aus.
3. Wählen Sie auf der Seite Bucket erstellen einen global eindeutigen Namen für Ihren S3-Bucket aus. Geben Sie z. B. ei **sc-s3-bucket**.
4. Wählen Sie für AWS-Region Ihre Region aus.
5. Wählen Sie für Bucket-Versionsverwaltung die Option Aktivieren aus.
6. Behalten Sie die übrigen Einstellungen unverändert bei und wählen Sie dann Bucket erstellen aus.

Speichern Sie Datenbankankmeldeinformationen in AWS Secrets Manager

Speichern Sie Ihre Quell- und Zieldatenbankankmeldedaten in AWS Secrets Manager. Stellen Sie sicher, dass Sie diese Geheimnisse auf Ihre AWS-Region replizieren. DMS Schema Conversion verwendet diese Secrets, um eine Verbindung mit Ihren Datenbanken im Migrationsprojekt herzustellen.

Um Ihre Datenbankankmeldedaten zu speichern in AWS Secrets Manager

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Secrets Manager Konsole unter <https://console.aws.amazon.com/secretsmanager/>.
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Die Seite Geheimnis-Typ auswählen wird geöffnet. Wählen Sie als Secret-Typ den Typ der zu speichernden Datenbank-Anmeldeinformation aus:
 - Anmeldeinformationen für eine Amazon-RDS-Datenbank – Wählen Sie diese Option, um Anmeldeinformationen für Ihre Amazon-RDS-Datenbank zu speichern. Geben Sie als Anmeldeinformationen die Anmeldeinformationen für Ihre Datenbank ein. Als Datenbank wählen Sie Ihre Datenbank aus.

- Anmeldeinformationen für eine andere Datenbank – Wählen Sie diese Option, um Anmeldeinformationen für Ihre Oracle- oder SQL-Server-Quelldatenbanken zu speichern. Geben Sie als Anmeldeinformationen die Anmeldeinformationen für Ihre Datenbank ein.
 - Anderer Geheimnistyp – Wählen Sie diese Option, um nur den Benutzernamen und das Passwort für die Verbindung mit Ihrer Datenbank zu speichern. Wählen Sie Zeile hinzufügen aus, um zwei Schlüssel-Wert-Paare hinzuzufügen. Stellen Sie sicher, dass Sie **username** und **password** für die Schlüsselnamen verwenden. Geben Sie für Werte, die sich auf diese Schlüssel beziehen, die Anmeldeinformationen für Ihre Datenbank ein.
4. Wählen Sie unter Verschlüsselungsschlüssel den AWS KMS Schlüssel aus, mit dem Secrets Manager den geheimen Wert verschlüsselt. Wählen Sie Weiter aus.
 5. Geben Sie auf der Seite Geheimnis konfigurieren einen aussagekräftigen Wert für Secret-Name ein. Geben Sie beispielsweise **sc-source-secret** als **sc-target-secret** ein.
 6. Wählen Sie Replizieren von Secrets und anschließend für AWS-Region Ihre Region aus. Wählen Sie Weiter aus.
 7. Wählen Sie auf der Seite Konfigurieren der Rotation die Option Weiter aus.
 8. Prüfen Sie auf der Seite Review (Prüfen) die Secret-Details und wählen Sie Store (Speichern).

Wiederholen Sie diese Schritte, um Anmeldeinformationen für Ihre Quell- und Zieldatenbanken zu speichern.

Erstellen von IAM-Rollen

Erstellen Sie AWS Identity and Access Management (IAM) -Rollen zur Verwendung in Ihrem Migrationsprojekt. DMS Schema Conversion verwendet diese IAM-Rollen für den Zugriff auf Ihren Amazon-S3-Bucket und Ihre in AWS Secrets Manager gespeicherten Datenbank-Anmeldeinformationen.

So erstellen Sie eine IAM-Rolle, die Zugriff auf Ihren Amazon-S3-Bucket ermöglicht

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie auf der Seite Vertrauenswürdige Entitäten auswählen die Option AWS -Service aus. Wählen Sie DMS aus.

5. Wählen Sie Weiter aus. Die Seite Berechtigungen hinzufügen wird geöffnet.
6. Geben Sie für Filterrichtlinien den Wert **S3** ein. Wählen Sie AmazonS3 FullAccess.
7. Wählen Sie Weiter aus. Die Seite Name, prüfen und erstellen wird geöffnet.
8. Geben Sie für Rollename einen beschreibenden Namen ein. Geben Sie z. B. ei **sc-s3-role**. Wählen Sie Rolle erstellen aus.
9. Geben Sie auf der Seite Rollen **sc-s3-role** für Rollename ein. Wählen Sie sc-s3-role aus.
10. Wählen Sie auf der Seite sc-s3-role die Registerkarte Vertrauensstellungen aus. Wählen Sie Vertrauensrichtlinie bearbeiten aus.
11. Bearbeiten Sie auf der Seite Vertrauensrichtlinie bearbeiten die Vertrauensstellungen für die Rolle, um den Service-Prinzipal `schema-conversion.dms.amazonaws.com` als vertrauenswürdige Entität zu verwenden.
12. Wählen Sie Update trust policy (Vertrauensrichtlinie aktualisieren) aus.

Um eine IAM-Rolle zu erstellen, die Zugriff bietet auf AWS Secrets Manager

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.](https://console.aws.amazon.com/iam/)
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie auf der Seite Vertrauenswürdige Entitäten auswählen die Option AWS -Service aus. Wählen Sie DMS aus.
5. Wählen Sie Weiter aus. Die Seite Berechtigungen hinzufügen wird geöffnet.
6. Geben Sie für Filterrichtlinien den Wert **Secret** ein. Wählen Sie SecretsManagerReadWrite.
7. Wählen Sie Weiter aus. Die Seite Name, prüfen und erstellen wird geöffnet.
8. Geben Sie für Rollename einen beschreibenden Namen ein. Geben Sie z. B. ei **sc-secrets-manager-role**. Wählen Sie Rolle erstellen aus.
9. Geben Sie auf der Seite Rollen **sc-secrets-manager-role** für Rollename ein. Wähle sc-secrets-manager-role.
10. Wählen Sie auf der sc-secrets-manager-roleSeite die Registerkarte Vertrauensbeziehungen aus. Wählen Sie Vertrauensrichtlinie bearbeiten aus.
11. Bearbeiten Sie auf der Seite Vertrauensrichtlinie bearbeiten die Vertrauensbeziehungen für die zu verwendende Rolle `schema-conversion.dms.amazonaws.com` und Ihren AWS DMS

regionalen Dienstprinzipal als vertrauenswürdige Entitäten. Dieser AWS DMS regionale Service Principal hat das folgende Format.

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

Ersetzen Sie *region-name* durch den Namen Ihrer Region, z. B. us-east-1.

Das folgende Code-Beispiel veranschaulicht den Prinzipal für die Region us-east-1.

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

Das folgende Codebeispiel zeigt eine Vertrauensrichtlinie für den Zugriff auf die AWS DMS Schemakonvertierung.

```
{
  "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. Wählen Sie Update trust policy (Vertrauensrichtlinie aktualisieren) aus.

Schritt 1: Erstellen eines Instance-Profiles

Bevor Sie ein Instance-Profil erstellen, konfigurieren Sie eine Subnetzgruppe für Ihr Instance-Profil. Weitere Informationen zum Erstellen einer Subnetzgruppe für Ihr AWS DMS Migrationsprojekt finden Sie unter [Erstellen einer Subnetzgruppe](#).

Sie können ein Instance-Profil entsprechend dem folgenden Verfahren erstellen. In diesem Instance-Profil geben Sie Netzwerk- und Sicherheitseinstellungen für Ihr DMS-Schema-Conversion-Projekt an.

So erstellen Sie ein Instance-Profil

1. Melden Sie sich unter <https://console.aws.amazon.com/dms/v2/> bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich Instance-Profile und dann Instance-Profil erstellen aus.
3. Geben Sie unter Name einen eindeutigen Namen für Ihr Instance-Profil ein. Geben Sie z. B. ein **sc-instance**.
4. Wählen Sie als Netzwerktyp die Option IPv4 aus, um ein Instance-Profil zu erstellen, das nur IPv4-Adressierung unterstützt. Wenn Sie ein Instance-Profil erstellen möchten, das IPv4- und IPv6-Adressierung unterstützt, wählen Sie Dual-Stack-Modus aus.
5. Wählen Sie als Virtual Private Cloud (VPC) die VPC aus, die Sie im Schritt zu den Voraussetzungen erstellt haben.
6. Wählen Sie als Subnetzgruppe die Subnetzgruppe für Ihr Instance-Profil aus. Um eine Verbindung zu Amazon-RDS-Datenbanken herzustellen, verwenden Sie eine Subnetzgruppe, die öffentliche Subnetze enthält. Um eine Verbindung zu On-Premises-Datenbanken herzustellen, verwenden Sie eine Subnetzgruppe, die private Subnetze enthält.
7. Wählen Sie Instance-Profil erstellen aus.

Verwenden Sie dieses Instance-Profil, um ein Migrationsprojekt zu erstellen.

Schritt 2: Konfigurieren Ihrer Datenanbieter

Als Nächstes erstellen Sie Datenanbieter, die Ihre Quell- und Zieldatenbanken beschreiben. Für jeden Datenanbieter geben Sie einen Datenspeichertyp und Informationen zum Speicherort an. Sie speichern Ihre Datenbank-Anmeldeinformationen nicht in einem Datenanbieter.

So erstellen Sie einen Datenanbieter für eine On-Premises-Quelldatenbank

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich Datenanbieter und dann Datenanbieter erstellen aus.
3. Geben Sie als Name einen eindeutigen Namen für Ihren Quelldatenanbieter ein. Geben Sie z. B. **ei sc - source**.
4. Wählen Sie als Engine-Typ den Typ der Datenbank-Engine für Ihren Datenanbieter aus.
5. Geben Sie Ihre Verbindungsinformationen für die Quelldatenbank an. Die Verbindungsparameter hängen von Ihrer Quelldatenbank-Engine ab. Weitere Informationen finden Sie unter [Erstellen von Datenanbietern](#).
6. Wählen Sie für Secure Sockets Layer (SSL)-Modus den Typ der SSL-Durchsetzung aus.
7. Wählen Sie Datenanbieter erstellen aus.

So erstellen Sie einen Datenanbieter für eine Amazon-RDS-Zieldatenbank

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich Datenanbieter und dann Datenanbieter erstellen aus.
3. Wählen Sie für Konfiguration die Option RDS-Datenbank-Instance aus.
4. Wählen Sie für Datenbank aus RDS die Option Durchsuchen und dann Ihre Datenbank aus. DMS Schema Conversion ruft die Informationen über den Engine-Typ, den Servernamen und den Port automatisch ab.
5. Geben Sie als Name einen eindeutigen Namen für Ihren Zieldatenanbieter ein. Geben Sie z. B. **ei sc - target**.
6. Geben Sie unter Database name (Datenbankname) den Namen Ihrer Datenbank ein.
7. Wählen Sie für Secure Sockets Layer (SSL)-Modus den Typ der SSL-Durchsetzung aus.
8. Wählen Sie Datenanbieter erstellen aus.

Schritt 3: Erstellen eines Migrationsprojekts

Jetzt können Sie ein Migrationsprojekt erstellen. Im Migrationsprojekt geben Sie Ihre Quell- und Zieldatenanbieter sowie Ihr Instance-Profil an.

So erstellen Sie ein Migrationsprojekt

1. Wählen Sie Migrationsprojekte und dann Migrationsprojekt erstellen aus.

2. Geben Sie unter Name einen eindeutigen Namen für Ihr Migrationsprojekt ein. Geben Sie z. B. ein **sc-project**.
3. Wählen Sie für Instance-Profil die Option **sc-instance** aus.
4. Wählen Sie für Quelle die Option Durchsuchen und dann **sc-source** aus.
5. Wählen Sie für Secret-ID die Option **sc-source-secret** aus.
6. Wählen Sie für IAM Role (IAM-Rolle) **sc-secrets-manager-role** aus.
7. Wählen Sie für Ziel die Option Durchsuchen und dann **sc-target** aus.
8. Wählen Sie für Secret-ID die Option **sc-target-secret** aus.
9. Wählen Sie für IAM Role (IAM-Rolle) **schema-conversion-role** aus.
10. Wählen Sie Migrationsprojekt erstellen aus.

Schritt 4: Erstellen eines Bewertungsberichts

Sie erstellen den Bewertungsbericht zur Datenbankmigration, um die Komplexität der Migration zu bewerten. Dieser Bericht enthält die Liste aller Datenbankobjekte, die DMS Schema Conversion nicht automatisch konvertieren kann.

So erstellen Sie einen Bewertungsbericht

1. Wählen Sie Migrationsprojekte und dann **sc-project** aus.
2. Wählen Sie Schemakonvertierung und danach Schemakonvertierung starten aus.
3. Wählen Sie im Bereich zur Quelldatenbank das Datenbankschema aus, das bewertet werden soll. Aktivieren Sie außerdem das Kontrollkästchen für den Namen dieses Schemas.
4. Wählen Sie im Bereich zur Quelldatenbank Bewerten im Menü Aktionen aus. Das Dialogfeld Bewerten wird angezeigt.
5. Wählen Sie Bewerten in dem Dialogfeld aus, um Ihre Auswahl zu bestätigen.

Auf der Registerkarte Zusammenfassung wird die Anzahl der Elemente angezeigt, die DMS Schema Conversion automatisch für Datenbank-Speicherobjekte und Datenbank-Codeobjekte konvertieren kann.

6. Wählen Sie Aktionselemente aus, um die Liste aller Datenbankobjekte anzuzeigen, die DMS Schema Conversion nicht automatisch konvertieren kann. Sehen Sie sich die empfohlenen Aktionen für die einzelnen Elemente an.

7. Wählen Sie Ergebnisse exportieren aus, um eine Kopie Ihres Bewertungsberichts zu speichern. Wählen Sie dann eins der folgenden Formate aus: CSV oder PDF. Das Dialogfeld Exportieren wird angezeigt.
8. Wählen Sie Exportieren aus, um Ihre Auswahl zu bestätigen.
9. Wählen Sie S3-Bucket aus. Die Amazon-S3-Konsole wird geöffnet.
10. Wählen Sie Herunterladen aus, um Ihren Bewertungsbericht zu speichern.

Schritt 5: Konvertieren Ihres Quellcodes

Sie können Ihr Quelldatenbankschema anhand des folgenden Verfahrens konvertieren. Anschließend können Sie den konvertierten Code als SQL-Skripts in einer Textdatei speichern.

So konvertieren Sie Ihr Datenbankschema

1. Wählen Sie im Bereich zur Quelldatenbank das Datenbankschema aus, das konvertiert werden soll. Aktivieren Sie außerdem das Kontrollkästchen für den Namen dieses Schemas.
2. Wählen Sie im Bereich zur Quelldatenbank Konvertieren im Menü Aktionen aus. Das Dialogfeld Konvertieren wird angezeigt.
3. Wählen Sie Konvertieren im Dialogfeld aus, um Ihre Auswahl zu bestätigen.
4. Wählen Sie im Bereich zur Quelldatenbank ein Datenbankobjekt aus. DMS Schema Conversion zeigt den Quellcode und den konvertierten Code für dieses Objekt an. Sie können den konvertierten SQL-Code für ein Datenbankobjekt mit der Funktion „SQL bearbeiten“ bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihres konvertierten SQL-Codes](#).
5. Wählen Sie im Bereich zur Zieldatenbank das konvertierte Datenbankschema aus. Aktivieren Sie außerdem das Kontrollkästchen für den Namen dieses Schemas.
6. Wählen Sie für Aktionen die Option Als SQL speichern aus. Das Dialogfeld Speichern wird angezeigt.
7. Wählen Sie Als SQL speichern aus, um Ihre Auswahl zu bestätigen.
8. Wählen Sie S3-Bucket aus. Die Amazon-S3-Konsole wird geöffnet.
9. Wählen Sie Herunterladen aus, um Ihre SQL-Skripts zu speichern.

Schritt 6: Anwenden des konvertierten Codes

DMS Schema Conversion wendet den konvertierten Code nicht sofort auf Ihre Zieldatenbank an. Sie können die im vorherigen Schritt erstellten SQL-Skripts verwenden, um Ihre Zieldatenbank zu aktualisieren. Befolgen Sie alternativ das folgende Verfahren, um den konvertierten Code aus DMS Schema Conversion anzuwenden.

So wenden Sie den konvertierten Code an

1. Wählen Sie im Bereich zur Zieldatenbank das konvertierte Datenbankschema aus. Aktivieren Sie außerdem das Kontrollkästchen für den Namen dieses Schemas.
2. Wählen Sie für Aktionen die Option Änderungen anwenden aus. Das Dialogfeld Änderungen anwenden wird angezeigt.
3. Wählen Sie Anwenden aus, um Ihre Auswahl zu bestätigen.

Schritt 7: Bereinigen und Fehlersuche

Sie können Amazon verwenden CloudWatch , um Ihre DMS-Schemakonvertierungsprotokolle zu überprüfen oder zu teilen.

So überprüfen Sie die Protokolle zu DMS Schema Conversion

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie Protokolle und dann Protokollgruppen aus.

Der Name Ihrer Protokollgruppe zu DMS Schema Conversion beginnt mit `dms-tasks-sct`. Sie können die Protokollgruppen nach Erstellungszeit sortieren, um die Protokollgruppe zu DMS Schema Conversion zu finden.

Darüber hinaus enthält der Name Ihrer Protokollgruppe den Amazon-Ressourcennamen (ARN) Ihres Migrationsprojekts. Sie können den ARN Ihres Projekts auf der Seite Migrationsprojekte in DMS Schema Conversion sehen. Stellen Sie sicher, dass Sie ARN unter Präferenzen auswählen.

3. Wählen Sie den Namen Ihrer Protokollgruppe und anschließend den Namen Ihres Protokollstreams aus.

4. Wählen Sie unter Aktionen die Option Ergebnisse exportieren aus, um Ihr Protokoll zu DMS Schema Conversion zu speichern.

Bereinigen Sie Ihre Ressourcen, wenn die Schemakonvertierung in DMS Schema Conversion abgeschlossen ist.

So bereinigen Sie Ihre Ressourcen von DMS Schema Conversion

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich Migrationsprojekte aus.
 - a. Wählen Sie **sc-project**.
 - b. Wählen Sie Schemakonvertierung und anschließend Schemakonvertierung schließen aus.
 - c. Wählen Sie Löschen aus und bestätigen Sie Ihre Auswahl.
3. Wählen Sie im Navigationsbereich Instance-Profile aus.
 - a. Wählen Sie **sc-instance**.
 - b. Wählen Sie Löschen aus und bestätigen Sie Ihre Auswahl.
4. Wählen Sie im Navigationsbereich Datenanbieter aus.
 - a. Wählen Sie **sc-source** und **sc-target** aus.
 - b. Wählen Sie Löschen aus und bestätigen Sie Ihre Auswahl.

Stellen Sie außerdem sicher, dass Sie andere AWS Ressourcen, die Sie erstellt haben, bereinigen, z. B. Ihren Amazon S3 S3-Bucket, Datenbankgeheimnisse in AWS Secrets Manager, IAM-Rollen und Virtual Private Cloud (VPC).

Einrichten eines Netzwerks für DMS Schema Conversion

DMS Schema Conversion erstellt eine Schemakonvertierungs-Instance in einer Virtual Private Cloud (VPC), die auf dem Service Amazon VPC basiert. Wenn Sie Ihr Instance-Profil erstellen, geben Sie die VPC an, die Sie verwenden möchten. Sie können Ihre Standard-VPC für Ihr Konto und Ihre AWS-Region verwenden oder eine neue VPC erstellen.

Sie können verschiedene Netzwerkkonfigurationen verwenden, um die Interaktion für Ihre Quell- und Zieldatenbanken mit DMS Schema Conversion einzurichten. Diese Konfigurationen hängen vom

Standort Ihres Quelldatenanbieters und Ihren Netzwerkeinstellungen ab. Die folgenden Themen enthalten Beschreibungen gängiger Netzwerkkonfigurationen.

Themen

- [Verwendung einer einzigen VPC für Quell- und Zieldatenanbieter](#)
- [Verwendung mehrerer VPCs für Quell- und Zieldatenanbieter](#)
- [Verwenden von AWS Direct Connect oder einer VPN zum Konfigurieren eines Netzwerks für eine VPC](#)
- [Verwenden einer Internetverbindung zu einer VPC](#)
- [Verwenden einer Umgebung ohne Internet-Gateway](#)

Verwendung einer einzigen VPC für Quell- und Zieldatenanbieter

Die einfachste Netzwerkkonfiguration für DMS Schema Conversion ist eine Konfiguration mit einer VPC. Hier befinden sich Ihr Quelldatenanbieter, Ihr Instance-Profil und der Zieldatenanbieter alle in derselben VPC. Sie können diese Konfiguration verwenden, um Ihre Quelldatenbank auf einer Amazon-EC2-Instance zu konvertieren.

Wenn Sie diese Konfiguration verwenden möchten, stellen Sie sicher, dass die vom Instance-Profil verwendete VPC-Sicherheitsgruppe Zugriff auf die Datenanbieter hat. Sie können beispielsweise entweder einen Classless Inter-Domain Routing (CIDR)-VPC-Bereich oder die Elastic-IP-Adresse für Ihr Network Address Translation (NAT)-Gateway zulassen.

Verwendung mehrerer VPCs für Quell- und Zieldatenanbieter

Wenn sich Ihre Quell- und Zieldatenanbieter in verschiedenen VPCs befinden, können Sie Ihr Instance-Profil in einer der VPCs erstellen. Anschließend können Sie diese beiden VPCs mithilfe von VPC-Peering verknüpfen. Sie können diese Konfiguration verwenden, um Ihre Quelldatenbank auf einer Amazon-EC2-Instance zu konvertieren.

Eine VPC-Peering-Verbindung ist eine Netzwerkverbindung zwischen zwei VPCs. Diese aktiviert das Routing über die private IP-Adresse jeder VPC, als befänden sie sich im selben Netzwerk. Sie können eine VPC-Peering-Verbindung zwischen Ihren eigenen VPCs, mit einer VPC in einem anderen AWS-Konto oder mit einer VPC in einer anderen AWS-Region herstellen. Weitere Informationen zu VPC Peering finden Sie unter [VPC Peering](#) im Amazon VPC Benutzerhandbuch.

Befolgen Sie zum Implementieren von VPC-Peering die Anweisungen unter [Arbeiten mit VPC-Peering-Verbindungen](#) im Benutzerhandbuch für Amazon VPC. Stellen Sie sicher, dass die Routing-

Tabelle einer VPC den CIDR-Block der anderen enthält. Nehmen wir beispielsweise an, dass VPC A das Ziel 10.0.0.0/16 und VPC B das Ziel 172.31.0.0 verwendet. In diesem Fall muss die Routing-Tabelle von VPC A 172.31.0.0 und die Routing-Tabelle von VPC B 10.0.0.0/16 enthalten. Ausführlichere Informationen finden Sie unter [Aktualisieren Sie ihre Routing-Tabellen für eine VPC-Peering-Verbindung](#) im Handbuch für Amazon-VPC-Peering.

Verwenden von AWS Direct Connect oder einer VPN zum Konfigurieren eines Netzwerks für eine VPC

Für Remote-Netzwerke stehen für die Verbindung mit einer VPC mehrere Optionen zur Verfügung, z. B. AWS Direct Connect oder eine Software- oder Hardware-VPN-Verbindung. Sie können diese Optionen verwenden, um vorhandene lokale Services zu integrieren, indem ein internes Netzwerk in AWS Cloud erweitert wird. Sie können lokale Services wie beispielsweise Überwachung, Authentifizierung, Sicherheit, Daten oder andere Systeme integrieren. Mit dieser Art von Netzwerkerweiterung können Sie problemlos lokale Services mit von AWS gehosteten Ressourcen wie beispielsweise VPCs verbinden. Sie können diese Konfiguration verwenden, um Ihre On-Premises-Quelldatenbank zu konvertieren.

Bei dieser Konfiguration muss die VPC-Sicherheitsgruppe eine Routing-Regel enthalten, die den für einen VPC-CIDR-Bereich oder eine spezifische IP-Adresse bestimmten Datenverkehr an einen Host sendet. Dieser Host muss dazu in der Lage sein, den von der VPC kommenden Datenverkehr in ein lokales VPN weiterzuleiten. In diesem Fall enthält der NAT-Host seine eigenen Sicherheitsgruppeneinstellungen. Diese Einstellungen müssen Datenverkehr von Ihrem VPC-CIDR-Bereich oder Ihrer Sicherheitsgruppe zur NAT-Instance zulassen. Weitere Informationen finden Sie unter [Create a Site-to-Site VPN connection](#) im Benutzerhandbuch für AWS Site-to-Site VPN.

Verwenden einer Internetverbindung zu einer VPC

Wenn Sie weder ein VPN noch AWS Direct Connect für die Verbindung mit AWS-Ressourcen verwenden, können Sie eine Internetverbindung verwenden. Diese Konfiguration umfasst ein privates Subnetz in einer VPC mit einem Internet-Gateway. Das Gateway enthält den Zieldatenanbieter und das Instance-Profil. Sie können diese Konfiguration verwenden, um Ihre On-Premises-Quelldatenbank zu konvertieren.

Informationen zum Anfügen eines Internet-Gateways zu Ihrer VPC finden Sie unter [Anfügen eines Internet-Gateways](#) im Amazon VPC Benutzerhandbuch.

Die VPC-Routing-Tabelle muss Routing-Regeln enthalten, die den nicht für die VPC bestimmten Datenverkehr standardmäßig an das Internet-Gateway senden. In dieser Konfiguration scheint die

Verbindung zum Datenanbieter von der öffentlichen IP-Adresse Ihres NAT-Gateways zu stammen. Weitere Informationen finden Sie unter [VPC Route Tables](#) im Benutzerhandbuch für Amazon VPC.

Verwenden einer Umgebung ohne Internet-Gateway

Gehen Sie wie folgt vor, um eine Umgebung für die Schemakonvertierung ohne Internet-Gateway zu erstellen.

1. Führen Sie die Schritte 1 bis 3 im Tutorial [Erste Schritte](#) aus und nehmen Sie dabei die folgenden Änderungen vor:
 - Wählen Sie private Subnetze anstelle von öffentlichen.
 - Wählen Sie während der Instance-Erstellung für Öffentliche IP zuweisen die Option Nein aus.
2. Öffnen Sie die Amazon-VPC-Konsole.
3. Wählen Sie Endpunkte und dann Endpunkt erstellen aus.
4. Führen Sie auf der Seite Endpunkt erstellen die folgenden Schritte aus:
 - Wählen Sie für Servicekategorie die Option AWS-Services aus.
 - Wählen Sie in der Liste Services die Option com.amazonaws.**{region}**.secretsmanager aus.
 - Wählen Sie im Abschnitt VPC die von Ihnen erstellte VPC aus.
 - Wählen Sie die Subnetze für Ihre VPC aus.
 - Wählen Sie die Sicherheitsgruppe für Ihre VPC aus.
 - Lassen Sie für Richtlinie Option Vollzugriff ausgewählt.
5. Beenden Sie den Rest des Tutorials [Erste Schritte](#).

Erstellen von Quelldatenanbietern in DMS Schema Conversion

Sie können eine Microsoft SQL Server-, Oracle- oder PostgreSQL-Datenbank als Quelldatenanbieter in Migrationsprojekten für die DMS-Schemakonvertierung verwenden. Ihr Quelldatenanbieter kann eine selbstverwaltete Engine sein, die On-Premises oder auf einer Amazon-EC2-Instance ausgeführt wird.

Konfigurieren Sie das Netzwerk so, dass Interaktionen zwischen Ihrem Quelldatenanbieter und DMS Schema Conversion möglich sind. Weitere Informationen finden Sie unter [Einrichten eines Netzwerks für DMS Schema Conversion](#).

Themen

- [Verwenden einer Datenbank von Microsoft SQL Server als Quelle in DMS Schema Conversion](#)
- [Verwenden einer Oracle-Datenbank als Quelle in DMS Schema Conversion](#)
- [Verwenden einer Oracle-Data-Warehouse-Datenbank als Quelle in DMS Schema Conversion](#)
- [Verwenden einer PostgreSQL-Datenbank als Quelle bei der DMS-Schemakonvertierung](#)
- [Verwendung einer MySQL-Datenbank als Quelle bei der DMS-Schemakonvertierung](#)

Verwenden einer Datenbank von Microsoft SQL Server als Quelle in DMS Schema Conversion

Sie können SQL-Server-Datenbanken als Migrationsquelle in DMS Schema Conversion verwenden.

Sie können DMS Schema Conversion verwenden, um Datenbankcodeobjekte aus SQL Server in die folgenden Ziele zu konvertieren:

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

Informationen zu den unterstützten SQL-Server-Datenbankversionen finden Sie unter [Quelldatenanbieter für DMS Schema Conversion](#).

Weitere Informationen zur Verwendung der DMS-Schemakonvertierung mit einer SQL Server-Quelldatenbank finden Sie in der [step-by-step exemplarischen Vorgehensweise zur Migration von SQL Server zu MySQL](#).

Berechtigungen für Microsoft SQL Server als Quelle

Die folgenden Berechtigungen sind für Microsoft SQL Server als Quelle erforderlich:

- VIEW DEFINITION
- VIEW DATABASE STATE

Die Berechtigung `VIEW DEFINITION` erlaubt es Benutzern mit öffentlichem Zugriff, Objektdefinitionen anzuzeigen. DMS Schema Conversion verwendet die Berechtigung `VIEW DATABASE STATE`, um die Features der SQL Server Enterprise Edition zu überprüfen.

Gewähren Sie die erforderlichen Berechtigungen für alle Datenbanken, deren Schemata Sie konvertieren.

Gewähren Sie außerdem die folgenden Berechtigungen für die Datenbank `master`:

- `VIEW SERVER STATE`
- `VIEW ANY DEFINITION`

DMS Schema Conversion verwendet die Berechtigung `VIEW SERVER STATE`, um Servereinstellungen und -konfigurationen abzurufen. Stellen Sie sicher, dass Sie die Berechtigung `VIEW ANY DEFINITION` zum Anzeigen von Datenanbietern gewähren.

Führen Sie den folgenden Befehl in der Datenbank `master` aus, um Informationen über Microsoft Analysis Services zu lesen.

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

Ersetzen Sie im vorherigen Beispiel den Platzhalter `<user_name>` durch den Namen des Benutzers, dem Sie zuvor die erforderlichen Berechtigungen erteilt haben.

Um Informationen über den SQL Server-Agent zu lesen, fügen Sie Ihren Benutzer zur `AgentUser` SQL-Rolle hinzu. Führen Sie den folgenden Befehl für die Datenbank `msdb` aus.

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

Ersetzen Sie im vorherigen Beispiel den Platzhalter `<SQLAgentRole>` durch den Namen der SQL-Server-Agent-Rolle. Ersetzen Sie dann den Platzhalter `<user_name>` durch den Namen des Benutzers, dem Sie zuvor die erforderlichen Berechtigungen erteilt haben. Weitere Informationen finden Sie unter [Hinzufügen eines Benutzers zur AgentUser SQL-Rolle](#) im Amazon RDS-Benutzerhandbuch.

Gewähren Sie die Berechtigung `SELECT on dbo.log_shipping_primary_databases` für die Datenbank `msdb`, um den Versand von Protokollen zu erkennen.

Gewähren Sie die Berechtigung `RECEIVE ON <schema_name>.<queue_name>` für Ihre Quelldatenbanken, um den Benachrichtigungsansatz der Data Definition Language (DDL)-Replikation zu verwenden. Ersetzen Sie in diesem Beispiel den Platzhalter `<schema_name>` durch den Schemanamen Ihrer Datenbank. Ersetzen Sie dann den Platzhalter `<queue_name>` durch den Namen einer Warteschlangen-Tabelle.

Verwenden einer Oracle-Datenbank als Quelle in DMS Schema Conversion

Sie können Oracle-Datenbanken als Migrationsquelle in DMS Schema Conversion verwenden.

Verwenden Sie die Oracle-System-ID (SID), um eine Verbindung mit Ihrer Oracle-Datenbank herzustellen. Um die Oracle-SID abzurufen, senden Sie die folgende Abfrage an Ihre Oracle-Datenbank:

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

Sie können DMS Schema Conversion verwenden, um Datenbankcodeobjekte aus Oracle Database in die folgenden Ziele zu konvertieren:

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

Informationen zu den unterstützten Oracle-Datenbankversionen finden Sie unter [Quelldatenanbieter für DMS Schema Conversion](#).

Weitere Informationen zur Verwendung der DMS-Schemakonvertierung mit einer Oracle-Quelldatenbank finden Sie in der Walkthrough zur Migration von [Oracle nach PostgreSQL](#). step-by-step

Berechtigungen für Oracle als Quelle

Die folgenden Berechtigungen sind für Oracle als Quelle erforderlich:

- CONNECT
- SELECT_CATALOG_ROLE

- SELECT ANY DICTIONARY
- SELECT ON SYS.ARGUMENT\$

Verwenden einer Oracle-Data-Warehouse-Datenbank als Quelle in DMS Schema Conversion

Sie können Oracle-Data-Warehouse-Datenbanken als Migrationsquelle in DMS Schema Conversion verwenden, um Datenbankcodeobjekte und Anwendungscode zu Amazon Redshift zu konvertieren.

Informationen zu den unterstützten Oracle-Datenbankversionen finden Sie unter [Quelldatenanbieter für DMS Schema Conversion](#). Weitere Informationen zur Verwendung der DMS-Schemakonvertierung mit einer Oracle-Quelldatenbank finden Sie in der Walkthrough zur Migration von [Oracle nach PostgreSQL](#). step-by-step

Berechtigungen zur Verwendung einer Oracle-Data Warehouse-Datenbank als Quelle

Die folgenden Berechtigungen sind für Oracle Data Warehouse als Quelle erforderlich:

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY

Einstellungen für die Konvertierung von Oracle Data Warehouse zu Amazon Redshift

Weitere Informationen zum Bearbeiten der Einstellungen für DMS Schema Conversion finden Sie unter [Angeben von Einstellungen für die Schemakonvertierung für Migrationsprojekte](#).

Einstellungen für die Konvertierung von Oracle Data Warehouse zu Amazon Redshift sind u.a.:

- Dem konvertierten Code Kommentare für die Aktionselemente mit dem ausgewählten oder einem höheren Schweregrad hinzufügen: Diese Einstellung begrenzt die Anzahl der Kommentare mit Aktionselementen im konvertierten Code. DMS fügt dem konvertierten Code Kommentare zu Aktionselementen mit dem ausgewählten und höheren Schweregraden hinzu.

Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.

- Die maximale Anzahl von Tabellen für den Amazon-Redshift-Ziel-Cluster: Diese Einstellung legt die maximale Anzahl von Tabellen fest, die DMS auf Ihrem Amazon-Redshift-Ziel-Cluster anwenden kann. Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Diese Einstellung unterstützt die folgenden Werte:
 - Automatisch: DMS bestimmt je nach Knotentyp die Anzahl der Tabellen, die auf Ihrem Amazon-Redshift-Ziel-Cluster angewendet werden sollen.
 - Einen Wert festlegen: Stellen Sie die Anzahl der Tabellen manuell ein.

DMS konvertiert alle Ihre Quelltabellen, auch wenn die Anzahl der Tabellen größer ist als die, die Ihr Amazon-Redshift-Cluster speichern kann. DMS speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon-Redshift-Cluster-Kontingent für die Tabellen erreichen, zeigt DMS eine Warnmeldung an. Außerdem wendet DMS Tabellen auf Ihren Amazon-Redshift-Ziel-Cluster an, bis die Anzahl der Tabellen das Limit erreicht.

Weitere Informationen über Amazon-Redshift-Tabellenkontingente finden Sie unter [Kontingente und Limits in Amazon Redshift](#).

- Verwenden der UNION ALL-Ansicht: Mit dieser Einstellung können Sie die maximale Anzahl von Zieltabellen festlegen, die DMS für eine einzelne Quelltable erstellen kann.

Amazon Redshift unterstützt nicht die Tabellenpartitionierung. Um die Tabellenpartitionierung zu emulieren und Abfragen schneller auszuführen, kann DMS jede Partition Ihrer Quelltable in eine separate Tabelle in Amazon Redshift migrieren. Anschließend erstellt DMS eine Ansicht, die Daten aus allen erstellten Zieltabellen enthält.

DMS ermittelt automatisch die Anzahl an Partitionen in Ihrer Quelltable. Je nach Art der Quelltablepartitionierung kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon-Redshift-Cluster anwenden können. Um zu verhindern, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die DMS für Partitionen einer einzelnen Quelltable erstellen kann. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage im Jahr entspricht, plus zwei Tabellen für die Partitionen NO RANGE und UNKNOWN.

- Datetime-Formatelemente, die Sie im Oracle-Code verwenden, ähneln numerischen Datetime-Formatzeichenfolgen in Amazon Redshift: Verwenden Sie diese Einstellung, um Formatierungsfunktionen für Datetime-Datentypen wie TO_CHAR, TO_DATE und TO_NUMBER mit Datetime-Formatelementen zu konvertieren, die Amazon Redshift nicht unterstützt. Standardmäßig verwendet DMS Funktionen von Erweiterungspaketen, um diese nicht unterstützten Formatelemente im konvertierten Code zu emulieren.

Das Datetime-Formatmodell in Oracle enthält mehr Elemente als die Datetime-Formatzeichenfolgen in Amazon Redshift. Wenn Ihr Quellcode nur Datetime-Formatelemente enthält, die Amazon Redshift unterstützt, legen Sie diesen Wert fest, um Erweiterungspaket-Funktionen im konvertierten Code zu vermeiden. Wenn Sie die Erweiterungsfunktionen vermeiden, wird der konvertierte Code schneller ausgeführt.

- Numerische Formatelemente, die Sie im Oracle-Code verwenden, ähneln numerischen Formatzeichenfolgen in Amazon Redshift: Verwenden Sie diese Einstellung, um Formatierungsfunktionen für numerische Datentypen zu konvertieren, die Amazon Redshift nicht unterstützt. Standardmäßig verwendet DMS Funktionen von Erweiterungspaketen, um diese nicht unterstützten Formatelemente im konvertierten Code zu emulieren.

Das numerische Formatmodell in Oracle enthält mehr Elemente als die numerischen Formatzeichenfolgen in Amazon Redshift. Wenn Ihr Quellcode nur numerische Formatelemente enthält, die Amazon Redshift unterstützt, legen Sie diesen Wert fest, um Erweiterungspaket-Funktionen im konvertierten Code zu vermeiden. Wenn Sie die Erweiterungsfunktionen vermeiden, wird der konvertierte Code schneller ausgeführt.

- Verwenden der NVL-Funktion, um das Verhalten der LEAD- und LAG-Funktionen von Oracle zu emulieren: Wenn Ihr Quellcode nicht die Standardwerte für Offset in den LEAD- und LAG-Funktionen verwendet, kann DMS diese Funktionen mit der NVL-Funktion emulieren. Standardmäßig löst DMS für jede Verwendung der LEAD- und LAG-Funktionen ein Aktionselement aus. Wenn Sie diese Funktionen mit NVL emulieren, wird der konvertierte Code schneller ausgeführt.
- Emulieren des Verhaltens von Primär- und Einzelschlüsseln: Legen Sie diese Einstellung fest, damit DMS das Verhalten von Einschränkungen für Primär- und Einzelschlüssel auf dem Amazon-Redshift-Ziel-Cluster emuliert. Amazon Redshift erzwingt keine Einschränkungen für primäre und eindeutige Schlüssel und verwendet sie nur zu Informationszwecken. Wenn Ihr Quellcode Einschränkungen für Primär- oder Einzelschlüssel verwendet, legen Sie diese Einstellung fest, um sicherzustellen, dass DMS deren Verhalten emuliert.
- Komprimierungskodierung verwenden: Legen Sie diese Einstellung fest, um die Komprimierungskodierung auf Amazon-Redshift-Tabellenspalten anzuwenden. DMS weist mithilfe des standardmäßigen Redshift-Algorithmus automatisch die Komprimierungskodierung zu. Informationen zur Komprimierungskodierung finden Sie unter [Komprimierungskodierungen](#) im Amazon-Redshift-Datenbank-Entwicklerhandbuch.

Amazon Redshift wendet standardmäßig keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Um die Komprimierung auf diese Spalten anzuwenden,

legen Sie die Option Komprimierungskodierung für KEY-Spalten verwenden fest. Sie können diese Option nur auswählen, wenn Sie Komprimierungskodierung verwenden aktiviert haben.

Verwenden einer PostgreSQL-Datenbank als Quelle bei der DMS-Schemakonvertierung

Sie können PostgreSQL-Datenbanken als Migrationsquelle in der DMS-Schemakonvertierung verwenden.

Sie können die DMS-Schemakonvertierung verwenden, um Datenbankcodeobjekte aus der PostgreSQL-Datenbank in die folgenden Ziele zu konvertieren:

- MySQL
- Aurora MySQL

Die für PostgreSQL als Quelle erforderlichen Rechte lauten wie folgt:

- CONNECT ON DATABASE <database_name>
- USAGE ON SCHEMA <database_name>
- SELECT ON ALL TABLES IN SCHEMA <database_name>
- SELECT ON ALL SEQUENCES IN SCHEMA <database_name>

Verwendung einer MySQL-Datenbank als Quelle bei der DMS-Schemakonvertierung

Sie können MySQL-Datenbanken als Migrationsquelle in der DMS-Schemakonvertierung verwenden.

Sie können die DMS-Schemakonvertierung verwenden, um Datenbankcodeobjekte aus der MySQL-Datenbank in die folgenden Ziele zu konvertieren:

- PostgreSQL
- Aurora PostgreSQL

Die für MySQL als Quelle erforderlichen Rechte lauten wie folgt:

- SELECT ON *.*

- `SHOW VIEW ON *.*`

Einstellungen für die Konvertierung von MySQL nach PostgreSQL

Weitere Informationen zum Bearbeiten der Einstellungen für DMS Schema Conversion finden Sie unter [Angeben von Einstellungen für die Schemakonvertierung für Migrationsprojekte](#).

Zu den Einstellungen für die Konvertierung von MySQL nach PostgreSQL gehören die folgenden:

- Kommentare im konvertierten SQL-Code: Legen Sie diese Einstellung fest, um dem konvertierten Code Kommentare für die Aktionselemente mit dem ausgewählten Schweregrad und höher hinzuzufügen.

Zulässige Werte:

- Nur Fehler
- Errors and warnings (Fehler und Warnungen)
- Alle Nachrichten

Erstellen von Zieldatenanbietern in DMS Schema Conversion

Sie können MySQL- und PostgreSQL-Datenbanken als Zieldatenanbieter in Migrationsprojekten für DMS Schema Conversion verwenden. Ihr Zieldatenanbieter kann eine Amazon-EC2-, Amazon-RDS- oder Amazon-Aurora-Instance sein.

Themen

- [Verwenden einer MySQL-Datenbank als Ziel in DMS Schema Conversion](#)
- [Verwenden einer PostgreSQL-Datenbank als Ziel in DMS Schema Conversion](#)
- [Verwenden eines Amazon-Redshift-Clusters als Ziel in DMS Schema Conversion](#)

Verwenden einer MySQL-Datenbank als Ziel in DMS Schema Conversion

Sie können MySQL-Datenbanken als Migrationsziel in DMS Schema Conversion verwenden.

Informationen zu den unterstützten Zieldatenbanken finden Sie unter [Zieldatenanbieter für DMS Schema Conversion](#).

Berechtigungen für MySQL als Ziel

Die folgenden Berechtigungen sind für MySQL als Ziel erforderlich:

- 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.*

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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';
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie dann *your_password* durch ein sicheres Passwort.

Setzen Sie den Parameter `lower_case_table_names` auf 1, um Amazon RDS für MySQL oder Aurora MySQL als Ziel zu verwenden. Dieser Wert bedeutet, dass der MySQL-Server Kennungen von Objektnamen wie Tabellen, Indizes, Auslösern und Datenbanken ohne Berücksichtigung der Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Ziel-Instance aktiviert haben, setzen Sie den Parameter `log_bin_trust_function_creators` auf 1. In diesem Fall müssen Sie die Eigenschaften `DETERMINISTIC`, `READS SQL DATA` oder `NO SQL` nicht verwenden, um gespeicherte Funktionen zu erstellen. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Verwenden einer PostgreSQL-Datenbank als Ziel in DMS Schema Conversion

Sie können PostgreSQL-Datenbanken als Migrationsziel in DMS Schema Conversion verwenden.

Informationen zu den unterstützten Zieldatenbanken finden Sie unter [Zieldatenanbieter für DMS Schema Conversion](#).

Berechtigungen für PostgreSQL als Ziel

DMS Schema Conversion erfordert die Berechtigung `CREATE ON DATABASE`, um PostgreSQL als Ziel verwenden zu können. Erstellen Sie einen Benutzer und gewähren Sie diesem Benutzer

diese Berechtigung für jede Datenbank, die Sie im Migrationsprojekt für DMS Schema Conversion verwenden möchten.

DMS Schema Conversion erfordert die Rolle `rds_superuser`, um Amazon RDS für PostgreSQL als Ziel verwenden zu können.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standard-Suchpfad der Datenbank mit folgendem Befehl.

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

Ersetzen Sie in diesem Beispiel den Platzhalter `<db_name>` durch den Namen Ihrer Datenbank.

In PostgreSQL kann nur der Schemaeigentümer oder ein `superuser` ein Schema entfernen. Der Eigentümer kann ein Schema und alle darin enthaltenen Objekte löschen, auch wenn der Eigentümer des Schemas nicht der Eigentümer von einigen enthaltenen Objekten ist.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, wird möglicherweise eine Fehlermeldung angezeigt, falls DMS Schema Conversion ein Schema nicht löschen kann. Verwenden Sie die Rolle `superuser`, um diese Fehlermeldung zu vermeiden.

Verwenden eines Amazon-Redshift-Clusters als Ziel in DMS Schema Conversion

Sie können Amazon-Redshift-Datenbanken als Migrationsziel in DMS Schema Conversion verwenden. Informationen zu den unterstützten Zieldatenbanken finden Sie unter [Zieldatenanbieter für DMS Schema Conversion](#).

Berechtigungen für Amazon Redshift als Ziel

Die Verwendung von Amazon Redshift als Ziel für DMS Schema Conversion erfordert die folgenden Berechtigungen:

- `CREATE ON DATABASE`: Ermöglicht DMS, neue Schemata in der Datenbank zu erstellen.
- `CREATE ON SCHEMA`: Ermöglicht DMS, Objekte in dem Datenbankschema zu erstellen.
- `GRANT USAGE ON LANGUAGE`: Ermöglicht DMS, neue Funktionen und Prozeduren in der Datenbank zu erstellen.

- GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog: Stellt dem Benutzer Systeminformationen zum Amazon-Redshift-Cluster bereit.
- GRANT SELECT ON pg_class_info: Stellt dem Benutzer Informationen zum Tabellenverteilungsstil bereit.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und ihm Berechtigungen zu gewähren. Ersetzen Sie die Beispielwerte durch eigene Werte.

```
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;
```

Wiederholen Sie den GRANT CREATE ON SCHEMA-Vorgang für jedes Zielschema, auf das Sie den konvertierten Code anwenden oder zu dem Sie Daten migrieren möchten.

Sie können ein Erweiterungspaket auf Ihre Amazon-Redshift-Zieldatenbank anwenden. Ein Erweiterungspaket ist ein Add-On-Modul, das die Funktionen in der Quelldatenbank emuliert, die erforderlich sind, wenn Objekte zu Amazon Redshift konvertiert werden. Weitere Informationen finden Sie unter [Verwenden von Erweiterungspaketen bei DMS Schema Conversion](#).

Verwalten von Migrationsprojekten in DMS Schema Conversion

Nachdem Sie ein Instance-Profil und kompatible Datenanbieter für die Schemakonvertierung erstellt haben, erstellen Sie ein Migrationsprojekt. Weitere Informationen finden Sie unter [Erstellen von Migrationsprojekten](#).

Um dieses neue Projekt in DMS Schema Conversion zu verwenden, wählen Sie Ihr Projekt in der Liste auf der Seite Migrationsprojekte aus. Wählen Sie anschließend auf der Registerkarte Schemakonvertierung die Option Schemakonvertierung starten aus.

Für den ersten Start von DMS Schema Conversion sind einige Einstellungen erforderlich. AWS Database Migration Service (AWS DMS) startet eine Schemakonvertierungs-Instance, was

bis zu 15 Minuten dauert. Bei diesem Prozess werden auch die Metadaten aus den Quell- und Zieldatenbanken gelesen. Nach einem erfolgreichen ersten Start können Sie schneller auf DMS Schema Conversion zugreifen.

Amazon beendet die Schemakonvertierungs-Instance, die Ihr Migrationsprojekt verwendet, innerhalb von drei Tagen nach Abschluss des Projekts. Sie können Ihr konvertiertes Schema und Ihren Bewertungsbericht aus dem Amazon-S3-Bucket abrufen, den Sie für DMS Schema Conversion verwenden.

Angeben von Migrationsprojekt-Einstellungen für DMS Schema Conversion

Nachdem Sie Ihr Migrationsprojekt erstellt und die Schemakonvertierung gestartet haben, können Sie Einstellungen für das Migrationsprojekt angeben. Sie können die Konvertierungseinstellungen ändern, um die Leistung des konvertierten Codes zu verbessern. Sie können auch die Ansicht der Schemakonvertierung anpassen.

Die Konvertierungseinstellungen hängen von Ihren Quell- und Zieldatenbank-Plattformen ab. Weitere Informationen erhalten Sie unter [Erstellen von Quelldatenanbietern](#) und [Erstellen von Zieldatenanbietern](#).

Verwenden Sie die Einstellungen für die Strukturansicht, um anzugeben, welche Schemas und Datenbanken in den Quell- und Zieldatenbank-Bereichen angezeigt werden sollen. Sie können leere Schemas, leere Datenbanken, Systemdatenbanken und benutzerdefinierte Datenbanken oder Schemas ausblenden.

So blenden Sie Datenbanken und Schemas in der Baumstrukturansicht aus

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt aus und dann auf der Registerkarte Schemakonvertierung die Option Schemakonvertierung starten aus.
4. Wählen Sie Settings (Einstellungen) aus. Die Seite Einstellungen wird geöffnet.
5. Führen Sie im Abschnitt Strukturansicht folgende Schritte aus:
 - Wählen Sie Leere Schemas ausblenden aus, um leere Schemas auszublenden.
 - Wählen Sie Leere Datenbanken ausblenden aus, um leere Datenbanken auszublenden.

- Wählen Sie für Systemdatenbanken oder Schemas Systemdatenbanken und Schemas dem Namen nach aus, um sie auszublenden.
- Geben Sie für Benutzerdefinierte Datenbanken oder Schemas die Namen benutzerdefinierter Datenbanken und Schemas ein, die ausgeblendet werden sollen. Wählen Sie Hinzufügen aus. Bei den Namen wird zwischen Groß- und Kleinschreibung unterschieden.

Wenn Sie mehrere Datenbanken oder Schemas hinzufügen möchten, trennen Sie die Namen durch ein Komma. Wenn Sie mehrere Objekte mit ähnlichen Namen hinzufügen möchten, verwenden Sie das Prozentzeichen (%) als Platzhalter. Dieser Platzhalter ersetzt eine beliebige Anzahl von Symbolen im Datenbank- oder Schemanamen.

Wiederholen Sie diese Schritte für die Abschnitte Quelle und Ziel.

6. Wählen Sie Anwenden und dann Schemakonvertierung aus.

Erstellung von Bewertungsberichten zur Datenbankmigration mit DMS Schema Conversion

Ein wichtiger Teil von DMS Schema Conversion ist der Bericht, der zur Unterstützung der Konvertierung Ihres Schemas generiert wird. Dieser Bewertungsbericht zur Datenbankmigration fasst alle Aufgaben zur Schemakonvertierung zusammen. Außerdem werden die Aktionspunkte für Schemata detailliert beschrieben, die nicht in die DB-Engine Ihrer Ziel-DB-Instance konvertiert werden können. Sie können den Bericht in der AWS DMS-Konsole anzeigen oder eine Kopie dieses Berichts als PDF- oder CSV-Dateien (durch Kommata getrennte Werte) ausführen.

Der Bericht zur Bewertung der Datenbankmigration enthält die folgenden Elemente:

- Kurze Zusammenfassung
- Empfehlungen, einschließlich Konvertierung von Server-Objekten, Backup-Vorschläge und verknüpfte Server-Änderungen

Wenn Sie Elemente haben, die DMS Schema Conversion nicht automatisch konvertieren kann, enthält der Bericht Schätzungen darüber, wie viel Aufwand erforderlich ist, um den entsprechenden Code für Ihre Ziel-DB-Instance zu schreiben.

Themen

- [Erstellen eines Berichts zur Bewertung der Datenbankmigration](#)
- [Anzeigen Ihres Bewertungsberichts zur Datenbankmigration](#)
- [Speichern Ihres Bewertungsberichts zur Datenbankmigration](#)

Erstellen eines Berichts zur Bewertung der Datenbankmigration

Nachdem Sie ein Migrationsprojekt erstellt haben, verwenden Sie das folgende Verfahren, um einen Bewertungsbericht zur Datenbankmigration zu erstellen.

So erstellen Sie einen Bewertungsbericht zur Datenbankmigration

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt und dann Schemakonvertierung aus.
4. Wählen Sie Schemakonvertierung starten aus. Die Seite Schemakonvertierung wird geöffnet.
5. Wählen Sie im Bereich „Quelldatenbank“ das Datenbankschema oder die Schemaelemente aus, die Sie bewerten möchten. Um mehrere Objekte in den Bericht aufzunehmen, stellen Sie sicher, dass Sie alle Elemente auswählen.
6. Nachdem Sie die Kontrollkästchen für alle Schemaobjekte aktiviert haben, die Sie bewerten möchten, müssen Sie den übergeordneten Knoten für die ausgewählten Objekte auswählen. Das Menü Aktionen im Bereich „Quelldatenbank“ ist jetzt verfügbar.
7. Wählen Sie Bewerten im Menü Aktionen. Ein Bestätigungsdialogfeld wird angezeigt.
8. Wählen Sie Bewerten in dem Dialogfeld aus, um Ihre Auswahl zu bestätigen.

Anzeigen Ihres Bewertungsberichts zur Datenbankmigration

Nachdem Sie einen Bewertungsbericht erstellt haben, fügt DMS Schema Conversion Informationen auf den folgenden Registerkarten hinzu:

- Übersicht
- Aktionselemente

Auf der Registerkarte Zusammenfassung wird die Anzahl der Elemente angezeigt, die DMS Schema Conversion automatisch konvertieren kann.

Auf der Registerkarte Aktionselemente werden Elemente angezeigt, die DMS Schema Conversion nicht automatisch konvertieren kann, sowie Empfehlungen zur Verwaltung dieser Elemente.

Zusammenfassung des Bewertungsberichts

Auf der Registerkarte Summary wird eine Zusammenfassung des Berichts zur Bewertung der Datenbankmigration angezeigt. Dies zeigt die Anzahl der Elemente an, die DMS Schema Conversion automatisch für Datenbankspeicherobjekte und Datenbankcodeobjekte konvertieren kann.

In den meisten Fällen kann DMS Schema Conversion nicht automatisch alle Schemaelemente in die Zieldatenbank-Engine konvertieren. Die Registerkarte Zusammenfassung enthält auch eine Schätzung des Aufwands, der erforderlich ist, Schemata in Ihrer Ziel-DB-Instance zu erstellen, die denen in Ihrer Quelldatenbank entsprechen.

Um die Konvertierungszusammenfassung für Datenbank-Speicherobjekte wie Tabellen, Sequenzen, Einschränkungen, Datentypen usw. anzuzeigen, wählen Sie Datenbank-Speicherobjekte.

Um die Konvertierungszusammenfassung für Datenbankcodeobjekte wie Prozeduren, Funktionen, Ansichten, Auslöser usw. anzuzeigen, wählen Sie Datenbankcode-Objekte.

Um den Umfang des Bewertungsberichts zu ändern, wählen Sie den gewünschten Knoten in der Quelldatenbankstruktur aus. DMS Schema Conversion aktualisiert die Zusammenfassung des Bewertungsberichts, so dass sie dem ausgewählten Umfang entspricht.

Aktionselemente im Bewertungsbericht

Die Registerkarte Aktionselemente enthält eine Liste von Elementen, die DMS Schema Conversion nicht automatisch in ein mit der Zieldatenbank-Engine kompatibles Format konvertieren kann. Für jedes Aktionselement stellt DMS Schema Conversion eine Beschreibung des Problems und die empfohlene Maßnahme bereit. DMS Schema Conversion gruppiert ähnliche Aktionselemente und zeigt deren Häufigkeit an.

Um den Code für das relationale Datenbankobjekt anzuzeigen, wählen Sie ein Aktionselement in der Liste aus.

Speichern Ihres Bewertungsberichts zur Datenbankmigration

Nachdem Sie einen Bewertungsbericht zur Datenbankmigration erstellt haben, können Sie eine Kopie dieses Berichts als PDF- oder CSV-Datei (Comma-Separated Value) speichern.

So speichern Sie einen Bewertungsbericht zur Datenbankmigration als PDF-Datei

1. Wählen Sie Exportieren und anschließend PDF. Prüfen Sie das Dialogfeld und wählen Sie Als PDF exportieren.
2. DMS Schema Conversion erstellt ein Archiv mit Ihrer PDF-Datei und speichert dieses Archiv in Ihrem Amazon-S3-Bucket. Um den Amazon-S3-Bucket zu ändern, bearbeiten Sie die Einstellungen für die Schemakonvertierung in Ihrem Instance-Profil.
3. Öffnen Sie die Bewertungsberichtsdatei in Ihrem Amazon-S3-Bucket.

So speichern Sie einen Bewertungsbericht zur Datenbankmigration als CSV-Dateien

1. Wählen Sie Exportieren und anschließend CSV. Prüfen Sie das Dialogfeld und wählen Sie Als CSV exportieren.
2. DMS Schema Conversion erstellt ein Archiv mit CSV-Dateien und speichert dieses Archiv in Ihrem Amazon-S3-Bucket. Um den Amazon-S3-Bucket zu ändern, bearbeiten Sie die Einstellungen für die Schemakonvertierung in Ihrem Instance-Profil.
3. Öffnen Sie die Bewertungsberichtsdateien in Ihrem Amazon-S3-Bucket.

Die PDF-Datei enthält sowohl die Zusammenfassung als auch Informationen zu den Aktionselementen.

Wenn Sie Ihren Bewertungsbericht als CSV exportieren, erstellt DMS Schema Conversion drei CSV-Dateien.

Die erste CSV-Datei enthält die folgenden Informationen über die Aktionselemente:

- Kategorie
- Vorkommen
- Aktionselement
- Betreff
- Gruppe
- Beschreibung
- Dokumentationsreferenzen
- Empfohlene Aktion
- Linien

- Position
- Quelle
- Ziel
- IP-Adresse und Port des Servers
- Datenbank
- Schema

Die zweite CSV-Datei enthält das `Action_Items_Summary`-Suffix in ihrem Namen und enthält die folgenden Informationen:

- Schema
- Aktionselement
- Anzahl der Vorkommen
- Lernkurvenaufwand, d. h. der Aufwand, der erforderlich ist, um ein Konzept für die Konvertierung der einzelnen Aktionselemente zu entwickeln
- Bemühungen, ein Vorkommen des Aktionselements zu konvertieren, wobei der Aufwand angegeben wird, der für die Konvertierung der einzelnen Aktionselemente erforderlich ist, wenn das entworfene Konzept verfolgt wird
- Aktionselementbeschreibung
- Empfohlene Aktion

Die Werte, die den Umfang der erforderlichen Bemühungen angeben, basieren auf einer gewichteten Skala, die von niedrig (am wenigsten) bis hoch (am meisten) reicht.

Die dritte CSV-Datei enthält das `Summary`-Suffix in ihrem Namen und enthält die folgenden Informationen:

- Kategorie
- Anzahl der Objekte
- Automatisch konvertierte Objekte
- Objekte mit einfachen Aktionen
- Objekte mit Aktionen mittlerer Komplexität
- Objekte mit komplexen Aktionen

- Gesamtzahl der Codezeilen

Verwenden von DMS Schema Conversion

DMS Schema Conversion konvertiert Ihre vorhandenen Datenbankschemas und die meisten Datenbank-Codeobjekte in ein mit der Zieldatenbank kompatibles Format.

DMS Schema Conversion automatisiert einen Großteil des Prozesses der Konvertierung Ihrer OLTP-Datenbankschemas (Online Transaction Processing, Online-Transaktionsverarbeitung) in Amazon RDS für MySQL oder RDS für PostgreSQL. Die Quell- und Zieldatenbank-Engines enthalten viele verschiedene Features und Funktionen und DMS Schema Conversion versucht, möglichst ein vergleichbares Schema zu erstellen. Für Datenbankobjekte, bei denen eine direkte Konvertierung nicht möglich ist, bietet DMS Schema Conversion eine Liste von möglichen Maßnahmen.

Gehen Sie wie folgt vor, um das Datenbankschema zu konvertieren:

- Bevor Sie Ihre Datenbankschemas konvertieren, richten Sie Transformationsregeln ein, die die Namen Ihrer Datenbankobjekte während der Konvertierung ändern.
- Erstellen Sie einen Bewertungsbericht zur Datenbankmigration, um die Komplexität der Migration abzuschätzen. Dieser Bericht enthält Einzelheiten zu den Schemaelementen, die DMS Schema Conversion nicht automatisch konvertieren kann.
- Konvertieren Sie Ihre Quelldatenbank-Speicher- und -Codeobjekte. DMS Schema Conversion erstellt eine lokale Version der konvertierten Datenbankobjekte. Sie können in Ihrem Migrationsprojekt auf diese konvertierten Objekte zugreifen.
- Speichern Sie den konvertierten Code in SQL-Dateien, um ihn zu überprüfen, zu bearbeiten oder die Aktionselemente zur Konvertierung anzugehen. Wenden Sie optional den konvertierten Code direkt auf Ihre Zieldatenbank an.

Um Data-Warehouse-Schemas zu konvertieren, verwenden Sie den Desktop AWS Schema Conversion Tool. Weitere Informationen finden Sie unter [Converting data warehouse schemas to Amazon Redshift](#) im Benutzerhandbuch zum AWS Schema Conversion Tool.

Themen

- [Einrichten von Transformationsregeln in DMS Schema Conversion](#)
- [Konvertieren von Datenbankschemas in DMS Schema Conversion](#)
- [Angaben von Einstellungen für die Schemakonvertierung für Migrationsprojekte](#)

- [Aktualisieren Ihrer Datenbankschemas in DMS Schema Conversion](#)
- [Speichern und Anwenden des konvertierten Codes in DMS Schema Conversion](#)

Einrichten von Transformationsregeln in DMS Schema Conversion

Bevor Sie Ihr Datenbankschema mit DMS Schema Conversion konvertieren, können Sie Transformationsregeln einrichten. Transformationsregeln können beispielsweise einen Objektnamen in Klein- oder Großbuchstaben ändern, ein Präfix oder Suffix hinzufügen oder entfernen und Objekte umbenennen. Nehmen wir beispielsweise an, dass Sie in Ihrem Quellschema eine Reihe von Tabellen mit dem Namen `test_TABLE_NAME` haben. Sie können eine Regel einrichten, die das Präfix `test_` im Zielschema in das Präfix `demo_` ändert.

Sie können Transformationsregeln für die folgenden Aufgaben erstellen:

- Ein Präfix hinzufügen, entfernen oder ersetzen
- Ein Suffix hinzufügen, entfernen oder ersetzen
- Den Datentyp einer Spalte ändern
- Den Objektnamen in Klein- oder Großbuchstaben ändern
- Umbenennen von Objekten

Sie können Transformationsregeln für die folgenden Objekte erstellen:

- Schema
- Tabelle
- Spalte

Erstellen von Transformationsregeln

DMS Schema Conversion speichert Transformationsregeln als Teil Ihres Migrationsprojekts. Sie können Transformationsregeln einrichten, wenn Sie Ihr Migrationsprojekt erstellen, oder sie später bearbeiten.

Sie können Ihrem Projekt mehrere Transformationsregeln hinzufügen. DMS Schema Conversion wendet Transformationsregeln während der Konvertierung in derselben Reihenfolge an, in der sie hinzugefügt wurden.

So erstellen Sie Transformationsregeln

1. Wählen Sie auf der Seite Migrationsprojekt erstellen die Option Transformationsregeln hinzufügen aus. Weitere Informationen finden Sie unter [Erstellen von Migrationsprojekten](#).
2. Wählen Sie unter Ziel der Regel den Typ der Datenbankobjekte aus, für die diese Regel gilt.
3. Wählen Sie für Quell-Schema die Option Schema eingeben aus. Geben Sie dann die Namen Ihrer Quellschemas, Tabellen und Spalten ein, für die diese Regel gilt. Sie können den exakten Namen eingeben und ein Objekt auswählen oder Sie können ein Muster angeben und damit mehrere Objekte auswählen. Verwenden Sie den Prozentsatz (%) als Platzhalter, um eine beliebige Anzahl von Symbolen im Datenbankobjektnamen zu ersetzen.
4. Wählen Sie unter Aktion die Aufgabe aus, die ausgeführt werden soll.
5. Geben Sie je nach Regeltyp einen oder zwei zusätzliche Werte ein. Wenn Sie beispielsweise ein Objekt umbenennen möchten, geben Sie den neuen Namen des Objekts ein. Wenn Sie ein Präfix ersetzen möchten, geben Sie das alte und das neue Präfix ein.
6. Wählen Sie Transformationsregeln hinzufügen aus, um eine weitere Transformationsregel hinzuzufügen.

Wenn Sie mit dem Hinzufügen von Regeln fertig sind, wählen Sie Migrationsprojekt erstellen aus.

Wenn Sie eine vorhandene Transformationsregel duplizieren möchten, wählen Sie Duplizieren aus. Wenn Sie eine vorhandene Transformationsregel bearbeiten möchten, wählen Sie die Regel in der Liste aus. Wenn Sie eine vorhandene Transformationsregel löschen möchten, wählen Sie Entfernen aus.

Bearbeiten von Transformationsregeln

Sie können in Ihrem Migrationsprojekt neue Transformationsregeln hinzufügen und vorhandene Transformationsregeln entfernen oder bearbeiten. Da DMS Schema Conversion die Transformationsregeln beim Start der Schemakonvertierung anwendet, müssen Sie die Schemakonvertierung schließen und erneut starten, nachdem Sie Ihre Regeln bearbeitet haben.

So bearbeiten Sie Transformationsregeln

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte und dann Ihr Migrationsprojekt aus.
3. Wählen Sie Schemakonvertierung und anschließend Schemakonvertierung schließen aus.

4. Nachdem die Schemakonvertierung AWS DMS geschlossen hat, wählen Sie Ändern, um Ihre Migrationsprojekteinstellungen zu bearbeiten.
5. Wählen Sie für Transformationsregeln eine der folgenden Aktionen:
 - Wählen Sie Duplizieren aus, um eine vorhandene Transformationsregel zu duplizieren und am Ende der Liste hinzuzufügen.
 - Wählen Sie Entfernen aus, um eine vorhandene Transformationsregel zu löschen.
 - Wählen Sie die vorhandene Transformationsregel aus, um sie zu bearbeiten.
6. Wählen Sie Änderungen speichern aus, wenn Sie mit dem Bearbeiten der Regeln fertig sind.
7. Wählen Sie Ihr Projekt in der Liste auf der Seite Migrationsprojekte aus. Wählen Sie Schemakonvertierung und anschließend Schemakonvertierung starten aus.

Konvertieren von Datenbankschemas in DMS Schema Conversion

Nachdem Sie das Migrationsprojekt erstellt und eine Verbindung mit Ihren Quell- und Zieldatenbanken hergestellt haben, können Sie Ihre Quelldatenbankobjekte in ein Format konvertieren, das mit Ihrer Zieldatenbank kompatibel ist. DMS Schema Conversion zeigt Ihr Quelldatenbankschema im linken Bereich in einer Strukturansicht an.

Jeder Knoten in der Datenbank-Strukturansicht wird per Lazy Loading abgerufen. Wenn Sie einen Knoten in der Strukturansicht auswählen, fordert DMS Schema Conversion die Schemainformationen aus der Quelldatenbank zu diesem Zeitpunkt an. Damit die Schemainformationen schneller geladen werden, wählen Sie Ihr Schema und dann Metadaten laden im Menü Aktionen aus. DMS Schema Conversion liest dann die Datenbank-Metadaten und speichert die Informationen in einem Amazon-S3-Bucket. Sie können die Datenbankobjekte jetzt schneller durchsuchen.

Sie können das gesamte Datenbankschema konvertieren oder ein beliebiges Schemaelement aus Ihrer Quelldatenbank zum Konvertieren auswählen. Falls das ausgewählte Schemaelement von einem übergeordneten Element abhängig ist, generiert DMS Schema Conversion auch das Schema für dieses übergeordnete Element. Wenn Sie beispielsweise eine Tabelle zum Konvertieren auswählen, erstellt DMS Schema Conversion die konvertierte Tabelle und das Datenbankschema, in dem sich die Tabelle befindet.

Konvertieren von Datenbankobjekten

Sie können DMS Schema Conversion verwenden, um ein ganzes Datenbankschema oder einzelne Datenbankschema-Objekte zu konvertieren.

So konvertieren Sie ein ganzes Datenbankschema

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt und dann Schemakonvertierung aus.
4. Wählen Sie Schemakonvertierung starten aus. Die Seite Schemakonvertierung wird geöffnet.
5. Aktivieren Sie im Bereich zur Quelldatenbank das Kontrollkästchen für den Schemanamen.
6. Wählen Sie dieses Schema im linken Bereich des Migrationsprojekts aus. DMS Schema Conversion hebt den Schemanamen blau hervor und aktiviert das Menü Aktionen.
7. Wählen Sie für Aktionen die Option Konvertieren aus. Daraufhin wird das Dialogfeld Konvertieren angezeigt.
8. Wählen Sie Konvertieren im Dialogfeld aus, um Ihre Auswahl zu bestätigen.

So konvertieren Sie Ihre Quelldatenbankobjekte

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt und dann Schemakonvertierung aus.
4. Wählen Sie Schemakonvertierung starten aus. Die Seite Schemakonvertierung wird geöffnet.
5. Wählen Sie im Bereich zur Quelldatenbank Ihre Quelldatenbankobjekte aus.
6. Nachdem Sie alle Kontrollkästchen für die Objekte aktiviert haben, die konvertiert werden sollen, wählen Sie den übergeordneten Knoten für alle ausgewählten Objekte im linken Bereich aus.

DMS Schema Conversion hebt den übergeordneten Knoten blau hervor und aktiviert das Menü Aktionen.
7. Wählen Sie für Aktionen die Option Konvertieren aus. Daraufhin wird das Dialogfeld Konvertieren angezeigt.
8. Wählen Sie Konvertieren im Dialogfeld aus, um Ihre Auswahl zu bestätigen.

Wenn Sie beispielsweise zwei von 10 Tabellen konvertieren möchten, aktivieren Sie die Kontrollkästchen für die beiden Tabellen, die Sie konvertieren möchten. Beachten Sie, dass das

Menü Aktionen inaktiv ist. Wenn Sie den Knoten Tabellen auswählen, hebt DMS Schema Conversion den Namen blau hervor und aktiviert das Menü Aktionen. Anschließend können Sie in diesem Menü die Option Konvertieren auswählen.

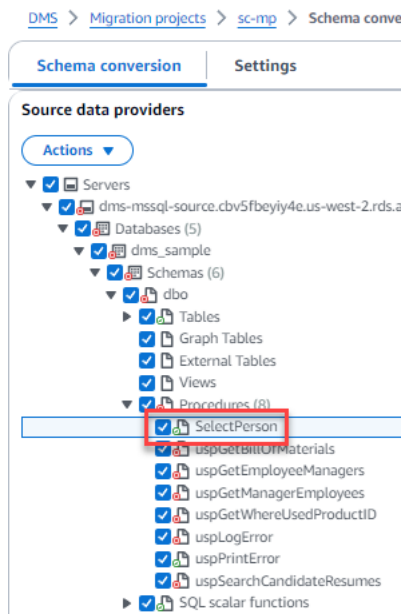
Wenn Sie zwei Tabellen und drei Verfahren konvertieren möchten, aktivieren Sie ebenfalls die Kontrollkästchen für die Objektnamen. Wählen Sie dann den Schemaknoten aus, um das Menü Aktionen zu aktivieren, und wählen Sie dann Schema konvertieren aus.

Bearbeiten und Speichern Ihres konvertierten SQL-Codes

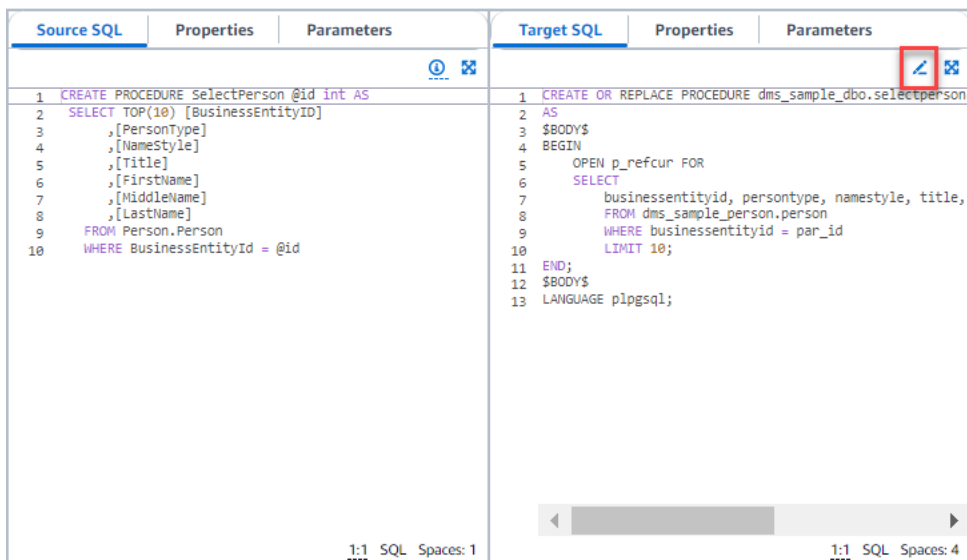
Auf der Seite Schemakonvertierung können Sie den verschlüsselten SQL-Code in Ihren Datenbankobjekten bearbeiten. Führen Sie die folgenden Schritte aus, um den konvertierten SQL-Code zu bearbeiten, die Änderungen zu übernehmen und sie anschließend zu speichern.

So können Sie konvertierten SQL-Code bearbeiten, Änderungen daran übernehmen und ihn speichern

1. Öffnen Sie auf der Seite Schemakonvertierung die Strukturansicht im Bereich Quelldatenanbieter, um ein Codeobjekt anzuzeigen.



2. Wählen Sie im Bereich Quelldatenanbieter die Option Aktionen, Konvertieren aus. Bestätigen Sie die Aktion.
3. Wenn die Konvertierung abgeschlossen ist, erweitern Sie bei Bedarf den mittleren Bereich, um das konvertierte SQL anzuzeigen. Um das konvertierte SQL zu bearbeiten, wählen Sie das Bearbeitungssymbol im Bereich Ziel-SQL.



4. Nachdem Sie die Ziel-SQL bearbeitet haben, bestätigen Sie Ihre Änderungen, indem Sie oben auf der Seite das Häkchen auswählen. Bestätigen Sie die Aktion.
5. Wählen Sie im Bereich Zieldatenanbieter die Option Aktionen, Änderungen anwenden aus. Bestätigen Sie die Aktion.
6. DMS schreibt die bearbeitete Prozedur in den Zieldatenspeicher.

Überprüfen von konvertierten Datenbankobjekten

Nachdem Sie Ihre Quelldatenbankobjekte konvertiert haben, können Sie im linken Bereich Ihres Projekts ein Objekt auswählen. Anschließend können Sie den Quell- und den konvertierten Code für dieses Objekt anzeigen. DMS Schema Conversion lädt den konvertierten Code für das im linken Bereich ausgewählte Objekt automatisch. Sie können auch die Eigenschaften oder Parameter des ausgewählten Objekts sehen.

DMS Schema Conversion speichert den konvertierten Code automatisch als Teil des Migrationsprojekts. Diese Codeänderungen werden nicht auf die Zieldatenbank angewendet. Weitere Informationen zum Anwenden von konvertiertem Code auf Ihre Zieldatenbank finden Sie unter [Anwenden des konvertierten Codes](#). Um den konvertierten Code aus dem Migrationsprojekt zu entfernen, wählen Sie Ihr Zielschema im rechten Bereich aus und wählen Sie dann die Option Aus Datenbank aktualisieren unter Aktionen.

Nachdem Sie Ihre Quelldatenbankobjekte konvertiert haben, können Sie die Konvertierungsübersicht und die Aktionselemente im unteren mittleren Bereich sehen. Sie erhalten die gleichen Informationen, wenn Sie einen Bewertungsbericht erstellen. Mit diesem Bewertungsbericht können Sie die

Schemaelemente ermitteln und auflösen, die DMS Schema Conversion nicht konvertieren kann. Sie können die Zusammenfassung des Bewertungsberichts und die Liste der Aktionselemente zur Konvertierung in CSV-Dateien speichern. Weitere Informationen finden Sie unter [Bewertungsberichte zur Datenbankmigration](#).

Angeben von Einstellungen für die Schemakonvertierung für Migrationsprojekte

Nachdem Sie ein Migrationsprojekt erstellt haben, können Sie die Konvertierungseinstellungen in DMS Schema Conversion angeben. Durch die Konfiguration der Einstellungen für die Schemakonvertierung wird die Leistung des konvertierten Codes verbessert.

So bearbeiten Sie die Konvertierungseinstellungen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt aus. Wählen Sie Schemakonvertierung und dann Schemakonvertierung starten aus.
4. Wählen Sie Settings (Einstellungen) aus. Die Seite Einstellungen wird geöffnet.
5. Ändern Sie die Einstellungen im Abschnitt Konvertierung.
6. Wählen Sie Anwenden und dann Schemakonvertierung aus.

Sie können die Anzahl der Kommentare mit Aktionselementen im konvertierten Code für alle Konvertierungspaare einschränken. Öffnen Sie zum Einschränken der Anzahl der Kommentare im konvertierten Code die Konvertierungseinstellungen im Migrationsprojekt.

Wählen Sie für Kommentare im konvertierten SQL-Code den Schweregrad der Aktionselemente aus. DMS Schema Conversion fügt dem konvertierten Code Kommentare zu Aktionselementen mit dem ausgewählten und höheren Schweregraden hinzu. Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus.

Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.

Andere Konvertierungseinstellungen unterscheiden sich je nach Kombination aus Quell- und Zieldatenbank.

Themen

- [Einstellungen für die Konvertierung von Oracle in MySQL](#)
- [Einstellungen für die Konvertierung von Oracle in PostgreSQL](#)
- [Einstellungen für die Konvertierung von SQL Server in MySQL](#)
- [Einstellungen für die Konvertierung von SQL Server in PostgreSQL](#)
- [Einstellungen für die Konvertierung von PostgreSQL in MySQL](#)

Einstellungen für die Konvertierung von Oracle in MySQL

Die Einstellungen für die Konvertierung von Oracle in MySQL in DMS Schema Conversion umfassen Folgendes:

- Ihre Oracle-Quelldatenbank kann die Pseudospalte ROWID verwenden. MySQL unterstützt keine ähnlichen Funktionen. DMS Schema Conversion kann die Pseudospalte ROWID im konvertierten Code emulieren. Aktivieren Sie dazu die Option Zeilen-ID generieren.

Wenn Ihr Oracle-Quellcode die Pseudospalte ROWID nicht verwendet, deaktivieren Sie die Option Zeilen-ID generieren. In diesem Fall arbeitet der konvertierte Code schneller.

- Der Oracle-Quellcode kann die Funktionen TO_CHAR, TO_DATE und TO_NUMBER mit Parametern enthalten, die MySQL nicht unterstützt. Standardmäßig emuliert DMS Schema Conversion die Verwendung dieser Parameter im konvertierten Code.

Sie können die nativen MySQL-Funktionen TO_CHAR, TO_DATE und TO_NUMBER verwenden, wenn Ihrem Oracle-Quellcode Parameter fehlen, die von MySQL nicht unterstützt werden. In diesem Fall arbeitet der konvertierte Code schneller. Wählen Sie dazu die folgenden Werte aus:

- Native MySQL-Funktion TO_CHAR verwenden
- Native MySQL-Funktion TO_DATE verwenden
- Native MySQL-Funktion TO_NUMBER verwenden
- Ihre Datenbank und Anwendungen können in verschiedenen Zeitzonen ausgeführt werden. Standardmäßig emuliert DMS Schema Conversion Zeitzonen im konvertierten Code. Sie benötigen diese Emulation jedoch nicht, wenn Ihre Datenbank und Anwendungen dieselbe Zeitzone verwenden. Wählen Sie in diesem Fall die Option Leistung des konvertierten Codes verbessern, wenn die Datenbank und die Anwendungen dieselbe Zeitzone verwenden.

Einstellungen für die Konvertierung von Oracle in PostgreSQL

Die Einstellungen für die Konvertierung von Oracle in PostgreSQL in DMS Schema Conversion umfassen Folgendes:

- AWS DMS kann materialisierte Oracle-Ansichten in Tabellen oder materialisierte Ansichten in PostgreSQL konvertieren. Wählen Sie für Materialisierte Ansichten aus, wie Ihre materialisierten Quellansichten konvertiert werden sollen.
- Ihre Oracle-Quelldatenbank kann die Pseudospalte ROWID verwenden. PostgreSQL unterstützt keine ähnlichen Funktionen. DMS Schema Conversion kann die Pseudospalte ROWID im konvertierten Code mit dem Datentyp `bigint` oder `character varying` emulieren. Wählen Sie dazu Datentyp `Bigint` zum Emulieren der Pseudospalte ROWID verwenden oder Datentyp `Character Varying` zum Emulieren der Pseudospalte ROWID verwenden für Zeilen-ID aus.

Wenn Ihr Oracle-Quellcode die Pseudospalte ROWID nicht verwendet, wählen Sie Nicht generieren aus. In diesem Fall arbeitet der konvertierte Code schneller.

- Der Oracle-Quellcode kann die Funktionen `TO_CHAR`, `TO_DATE` und `TO_NUMBER` mit Parametern enthalten, die PostgreSQL nicht unterstützt. Standardmäßig emuliert DMS Schema Conversion die Verwendung dieser Parameter im konvertierten Code.

Sie können die nativen PostgreSQL-Funktionen `TO_CHAR`, `TO_DATE` und `TO_NUMBER` verwenden, wenn Ihrem Oracle-Quellcode Parameter fehlen, die von PostgreSQL nicht unterstützt werden. In diesem Fall arbeitet der konvertierte Code schneller. Wählen Sie dazu die folgenden Werte aus:

- Native PostgreSQL-Funktion `TO_CHAR` verwenden
- Native PostgreSQL-Funktion `TO_DATE` verwenden
- Native PostgreSQL-Funktion `TO_NUMBER` verwenden
- Ihre Datenbank und Anwendungen können in verschiedenen Zeitzonen ausgeführt werden. Standardmäßig emuliert DMS Schema Conversion Zeitzonen im konvertierten Code. Sie benötigen diese Emulation jedoch nicht, wenn Ihre Datenbank und Anwendungen dieselbe Zeitzone verwenden. Wählen Sie in diesem Fall die Option Leistung des konvertierten Codes verbessern, wenn die Datenbank und die Anwendungen dieselbe Zeitzone verwenden.
- Damit weiterhin Sequenzen in Ihrem konvertierten Code verwendet werden, wählen Sie Konvertierte Sequenzen mit dem letzten quellseitig generierten Wert ausfüllen.
- In Einzelfällen speichert Ihre Oracle-Quelldatenbank möglicherweise nur Ganzzahlwerte in den Primär- oder Fremdschlüsselspalten des Datentyps `NUMBER`. In diesen Fällen AWS DMS kann diese Spalten in den `BIGINT` Datentyp konvertieren. Dieser Ansatz verbessert die Leistung

des konvertierten Codes. Wählen Sie dazu Primär- und Fremdschlüsselspalten des Datentyps NUMBER in den Datentyp BIGINT konvertieren. Stellen Sie sicher, dass Ihre Quelle keine Gleitkommawerte in diesen Spalten enthält, um Datenverlust zu vermeiden.

- Wählen Sie Nur aktive Auslöser und Einschränkungen konvertieren aus, damit deaktivierte Auslöser und Einschränkungen in Ihrem Quellcode übersprungen werden.
- Sie können DMS Schema Conversion verwenden, um Zeichenfolgenvariablen zu konvertieren, die als dynamisches SQL bezeichnet werden. Ihr Datenbankcode kann die Werte dieser Zeichenfolgenvariablen ändern. Um sicherzustellen, dass AWS DMS immer den neuesten Wert dieser Zeichenfolgenvariable konvertiert, wählen Sie Konvertieren des dynamischen SQL-Codes aus, der in aufgerufenen Routinen erstellt wird.
- PostgreSQL-Versionen 10 und früher unterstützen keine Verfahren. Wenn Sie mit der Verwendung von Prozeduren in PostgreSQL nicht vertraut sind, AWS DMS kann Oracle-Prozeduren in PostgreSQL-Funktionen konvertieren. Wählen Sie dazu Verfahren in Funktionen konvertieren aus.
- Wenn Sie zusätzliche Informationen zu den aufgetretenen Aktionselementen anzeigen möchten, können Sie dem Erweiterungspaket spezifische Funktionen hinzufügen. Wählen Sie dazu Erweiterungspaket-Funktionen hinzufügen, die benutzerdefinierte Ausnahmen auslösen. Wählen Sie anschließend Schweregrade zum Auslösen benutzerdefinierter Ausnahmen aus. Stellen Sie sicher, dass Sie nach dem Konvertieren Ihrer Quelldatenbankobjekte das Erweiterungspaket-Schema anwenden. Weitere Informationen zu Erweiterungspaketen finden Sie unter [Verwenden von Erweiterungspaketen](#).
- Ihre Oracle-Quelldatenbank kann Einschränkungen mit den automatisch generierten Namen enthalten. Wenn Ihr Quellcode diese Namen verwendet, stellen Sie sicher, dass Sie die Option Namen der vom System generierten Einschränkungen beibehalten auswählen. Wenn Ihr Quellcode diese Einschränkungen, aber nicht ihre Namen verwendet, deaktivieren Sie diese Option, um die Konvertierungsgeschwindigkeit zu erhöhen.
- Wenn Ihre Quell- und Zieldatenbanken in unterschiedlichen Zeitzonen ausgeführt werden, gibt die Funktion, die die integrierte Oracle-Funktion SYSDATE emuliert, andere Werte als die Quellfunktion zurück. Wählen Sie Zeitzone Ihrer Quelldatenbank festlegen aus, um sicherzustellen, dass die Quell- und die Zielfunktion die gleichen Werte zurückgeben.
- Sie können die Funktionen der Erweiterung orafce im konvertierten Code verwenden. Wählen Sie dazu für In Oracle integrierte Routinen die Funktionen aus, die Sie verwenden möchten. Weitere Informationen zu orafce finden Sie unter [orafce](#) auf GitHub.

Einstellungen für die Konvertierung von SQL Server in MySQL

Die Einstellungen für die Konvertierung von SQL Server in MySQL in DMS Schema Conversion umfassen Folgendes:

- Ihre SQL-Server-Quelldatenbank kann die Ausgabe von EXEC in einer Tabelle speichern. DMS Schema Conversion erstellt temporäre Tabellen und ein zusätzliches Verfahren, um dieses Feature zu emulieren. Wählen Sie Zusätzliche Routinen für die Verarbeitung offener Datensätze erstellen aus, um diese Emulation zu verwenden.

Einstellungen für die Konvertierung von SQL Server in PostgreSQL

Die Einstellungen für die Konvertierung von SQL Server in PostgreSQL in DMS Schema Conversion umfassen Folgendes:

- In SQL Server können Sie Indizes mit dem gleichen Namen in verschiedenen Tabellen verwenden. In PostgreSQL müssen jedoch alle im Schema verwendeten Indexnamen eindeutig sein. Wählen Sie Eindeutige Namen für Indizes generieren aus, um sicherzustellen, dass DMS Schema Conversion für alle Indizes eindeutige Namen generiert.
- PostgreSQL-Versionen 10 und früher unterstützen keine Verfahren. Wenn Sie mit der Verwendung von Prozeduren in PostgreSQL nicht vertraut sind, AWS DMS kann SQL Server-Prozeduren in PostgreSQL-Funktionen konvertieren. Wählen Sie dazu Verfahren in Funktionen konvertieren aus.
- Ihre SQL-Server-Quelldatenbank kann die Ausgabe von EXEC in einer Tabelle speichern. DMS Schema Conversion erstellt temporäre Tabellen und ein zusätzliches Verfahren, um dieses Feature zu emulieren. Wählen Sie Zusätzliche Routinen für die Verarbeitung offener Datensätze erstellen aus, um diese Emulation zu verwenden.
- Sie können die Vorlage definieren, die für die Schemanamen im konvertierten Code verwendet werden soll. Wählen Sie für Schema-Namen eine der folgenden Optionen aus:
 - DB – Verwendet den SQL-Server-Datenbanknamen als Schemanamen in PostgreSQL.
 - SCHEMA – Verwendet den SQL-Server-Schemanamen als Schemanamen in PostgreSQL.
 - DB_SCHEMA – Verwendet eine Kombination aus SQL-Server-Datenbank- und -Schemanamen als Schemanamen in PostgreSQL.
- Sie können die Schreibweise der Quellobjektnamen beibehalten. Wählen Sie Objektnamen in der gleichen Schreibweise beibehalten, um zu verhindern, dass Objektnamen in Kleinbuchstaben umgewandelt werden. Diese Option gilt nur, wenn Sie die Option zur Berücksichtigung von Groß- und Kleinschreibung in der Zieldatenbank aktivieren.

- Sie können die Parameternamen aus der Quelldatenbank beibehalten. DMS Schema Conversion kann den Namen von Parametern im konvertierten Code doppelte Anführungszeichen hinzufügen. Wählen Sie dazu die Option Ursprüngliche Parameternamen beibehalten aus.
- Sie können eine Länge von Routineparametern aus Ihrer Quelldatenbank behalten. DMS Schema Conversion erstellt Domains und verwendet sie, um eine Länge von Routineparametern anzugeben. Wählen Sie dazu die Option Parameterlängen beibehalten aus.

Einstellungen für die Konvertierung von PostgreSQL in MySQL

Die PostgreSQL-MySQL-Konvertierungseinstellungen in DMS Schema Conversion umfassen Folgendes:

- Kommentare im konvertierten SQL-Code: Diese Einstellung enthält Kommentare im konvertierten Code für die Aktionselemente des ausgewählten Schweregrads und höher. Diese Einstellung unterstützt die folgenden Werte:
 - Nur Fehler
 - Errors and warnings (Fehler und Warnungen)
 - Alle Nachrichten

Aktualisieren Ihrer Datenbankschemas in DMS Schema Conversion

Nachdem Sie ein Migrationsprojekt erstellt haben, speichert DMS Schema Conversion die Informationen zu Ihren Quell- und Zielschemas in diesem Projekt. DMS Schema Conversion verwendet Lazy Loading, um Metadaten nur dann zu laden, wenn sie benötigt werden, z. B. wenn Sie einen Knoten in Ihrer Datenbank-Strukturansicht auswählen. Sie können Eager Loading verwenden, um die Schemainformationen schneller zu laden. Wählen Sie dazu Ihr Schema und dann Metadaten laden unter Aktionen aus.

Nachdem Sie das Objekt automatisch oder manuell in Ihr Migrationsprojekt geladen haben, verwendet DMS Schema Conversion Lazy Loading nicht erneut. Wenn Sie also Objekte wie Tabellen und Verfahren in Ihrer Datenbank ändern, müssen Sie sie unbedingt in Ihrem Migrationsprojekt aktualisieren.

Um Schemas aus der Datenbank zu aktualisieren, wählen Sie die Objekte, die aktualisiert werden sollen, und anschließend Aus Datenbank aktualisieren unter Aktionen aus. Sie können Datenbankobjekte in Ihren Quell- und Zieldatenbankschemas aktualisieren:

- Quelle – Wenn Sie das Schema der Quelldatenbank aktualisieren, wählen Sie Aus Datenbank aktualisieren aus, um das Schema im Projekt durch das neueste Schema von der Quelldatenbank zu ersetzen.
- Ziel – Wenn Sie das Schema der Zieldatenbank aktualisieren, ersetzt DMS Schema Conversion das Schema im Projekt durch das neueste Schema von der Zieldatenbank. DMS Schema Conversion ersetzt den konvertierten Code durch den Code aus der Zieldatenbank. Stellen Sie sicher, dass Sie den konvertierten Code auf die Zieldatenbank angewendet haben, bevor Sie Aus Datenbank aktualisieren auswählen. Konvertieren Sie andernfalls Ihr Quelldatenbankschema erneut.

Speichern und Anwenden des konvertierten Codes in DMS Schema Conversion

Nach dem Konvertieren Ihrer Quelldatenbankobjekte wendet DMS Schema Conversion den konvertierten Code nicht sofort auf Ihre Zieldatenbank an. Stattdessen speichert DMS Schema Conversion den konvertierten Code im Projekt, bis Sie bereit sind, ihn auf die Zieldatenbank anzuwenden.

Bevor Sie den konvertierten Code anwenden, können Sie den Code Ihrer Quelldatenbank aktualisieren und die aktualisierten Objekte erneut konvertieren, um die vorhandenen Aktionselemente zu berücksichtigen. Weitere Informationen zu Elementen, die DMS Schema Conversion nicht automatisch konvertieren kann, finden Sie unter [Erstellung von Bewertungsberichten zur Datenbankmigration mit DMS Schema Conversion](#). Weitere Informationen zum Aktualisieren Ihrer Quelldatenbankobjekte im Migrationsprojekt für DMS Schema Conversion finden Sie unter [Aktualisieren Ihrer Datenbankschemas](#).

Anstatt den konvertierten Code in DMS Schema Conversion direkt auf Ihre Datenbank anzuwenden, können Sie den Code als SQL-Skript in einer Datei speichern. Sie können diese SQL-Skripts überprüfen, bei Bedarf bearbeiten und dann manuell auf Ihre Zieldatenbank anwenden.

Speichern des konvertierten Codes in einer SQL-Datei

Sie können das konvertierte Schema als SQL-Skripts in einer Textdatei speichern. Sie können den konvertierten Code ändern, um Aktionselemente zu berücksichtigen, die DMS Schema Conversion nicht automatisch konvertieren kann. Anschließend können Sie die aktualisierten SQL-Skripts auf der Zieldatenbank ausführen und den konvertierten Code auf die Zieldatenbank anwenden.

So speichern Sie Ihr konvertiertes Schema als SQL-Skripts

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt und dann Schemakonvertierung aus.
4. Wählen Sie Schemakonvertierung starten aus. Die Seite Schemakonvertierung wird geöffnet.
5. Wählen Sie im rechten Bereich das Zieldatenbankschema oder die konvertierten Objekte aus, die Sie speichern möchten. Stellen Sie sicher, dass DMS Schema Conversion den übergeordneten Knoten blau hervorhebt und das Menü Aktionen für die Zieldatenbank aktiviert.
6. Wählen Sie Als SQL speichern für Aktionen aus. Das Dialogfeld Speichern wird angezeigt.
7. Wählen Sie Als SQL speichern aus, um Ihre Auswahl zu bestätigen.

DMS Schema Conversion erstellt ein Archiv mit SQL-Dateien und speichert dieses Archiv in Ihrem Amazon-S3-Bucket.

8. (Optional) Ändern Sie den S3-Bucket für das Archiv, indem Sie die Einstellungen für die Schemakonvertierung in Ihrem Instance-Profil bearbeiten.
9. Öffnen Sie die SQL-Skripts über Ihren S3-Bucket.

Anwenden des konvertierten Codes

Wenn Sie bereit sind, den konvertierten Code auf die Zieldatenbank anzuwenden, wählen Sie im rechten Projektbereich die Datenbankobjekte aus. Sie können Änderungen auf ein ganzes Datenbankschema oder auf ausgewählte Datenbankschema-Objekte anwenden.

Nachdem Sie die Datenbankobjekte ausgewählt haben, hebt DMS Schema Conversion den Namen des ausgewählten Knotens oder des übergeordneten Knotens blau hervor. Anschließend wird das Menü Aktionen aktiviert. Wählen Sie Änderungen anwenden für Aktionen aus. Wählen Sie im daraufhin angezeigten Dialogfeld Anwenden aus, um Ihre Auswahl zu bestätigen und den konvertierten Code auf die Zieldatenbank anzuwenden.

Anwenden des Erweiterungspaket-Schemas

Wenn Sie Ihr konvertiertes Schema zum ersten Mal auf die Zieldatenbank anwenden, wendet DMS Schema Conversion möglicherweise auch das Erweiterungspaket-Schema an. Das Erweiterungspaket-Schema emuliert Systemfunktionen der Quelldatenbank, die zum Ausführen des konvertierten Codes für die Zieldatenbank benötigt werden. Wenn der konvertierte Code die

Funktionen des Erweiterungspakets verwendet, stellen Sie sicher, dass Sie das Erweiterungspaket-Schema anwenden.

Wählen Sie Änderungen anwenden unter Aktionen aus, um das Erweiterungspaket manuell auf die Zieldatenbank anzuwenden. Wählen Sie im daraufhin angezeigten Dialogfeld Bestätigen aus, um das Erweiterungspaket auf die Zieldatenbank anzuwenden.

Es wird empfohlen, das Erweiterungspaket-Schema nicht zu ändern, um unerwartete Ergebnisse im konvertierten Code zu vermeiden.

Weitere Informationen finden Sie unter [Verwenden von Erweiterungspaketen bei DMS Schema Conversion](#).

Verwenden von Erweiterungspaketen bei DMS Schema Conversion

Ein Erweiterungspaket in DMS Schema Conversion ist ein Zusatzmodul, das Funktionen der Quelldatenbank emuliert, die in der Zieldatenbank nicht unterstützt werden. Verwenden Sie ein Erweiterungspaket, um sicherzustellen, dass der konvertierte Code dieselben Ergebnisse liefert wie der Quellcode. Bevor Sie ein Erweiterungspaket installieren können, müssen Sie Ihre Datenbankschemata konvertieren.

Jedes Erweiterungspaket enthält ein Datenbankschema. Dieses Schema beinhaltet SQL-Funktionen, Prozeduren, Tabellen und Ansichten zur Emulation bestimmter OLTP-Objekte (Online Transaction Processing) oder nicht unterstützter integrierter Funktionen aus der Quelldatenbank.

Wenn Sie Ihre Quelldatenbank konvertieren, fügt DMS Schema Conversion ein zusätzliches Schema zur Zieldatenbank hinzu. Über dieses Schema werden SQL-Systemfunktionen der Quelldatenbank implementiert, die zum Schreiben des konvertierten Schemas in die Zieldatenbank benötigt werden. Dieses zusätzliche Schema wird als Erweiterungspaketschema bezeichnet.

Das Erweiterungspaketschema wird wie folgt entsprechend Ihrer Quelldatenbank benannt:

- Microsoft SQL Server – `aws_sqlserver_ext`
- Oracle – `aws_oracle_ext`

Sie können Erweiterungspakete auf zwei Arten anwenden:

- DMS Schema Conversion kann automatisch ein Erweiterungspaket anwenden, wenn Sie den konvertierten Code anwenden. DMS Schema Conversion wendet das Erweiterungspaket an, bevor es die anderen Schema-Objekte anwendet.

- Sie können ein Erweiterungspaket manuell anwenden. Wählen Sie dazu das Erweiterungspaket-Schema in Ihrer Zieldatenbankstruktur aus und klicken Sie dann auf Anwenden und dann auf Erweiterungspaket anwenden.

Migrieren von Datenbanken zu ihren Amazon-RDS-Entsprechungen mit AWS DMS

Homogene Datenmigrationen in AWS Database Migration Service (AWS DMS) vereinfachen die Migration von selbstverwalteten, On-Premises-Datenbanken zu ihren Amazon Relational Database Service (Amazon RDS)-Entsprechungen. Sie können beispielsweise homogene Datenmigrationen verwenden, um eine On-Premises-PostgreSQL-Datenbank zu Amazon RDS für PostgreSQL oder Aurora PostgreSQL zu migrieren. AWS DMS verwendet für homogene Datenmigrationen native Datenbank-Tools, um einfache und leistungsstarke Like-to-Like-Migrationen zu ermöglichen.

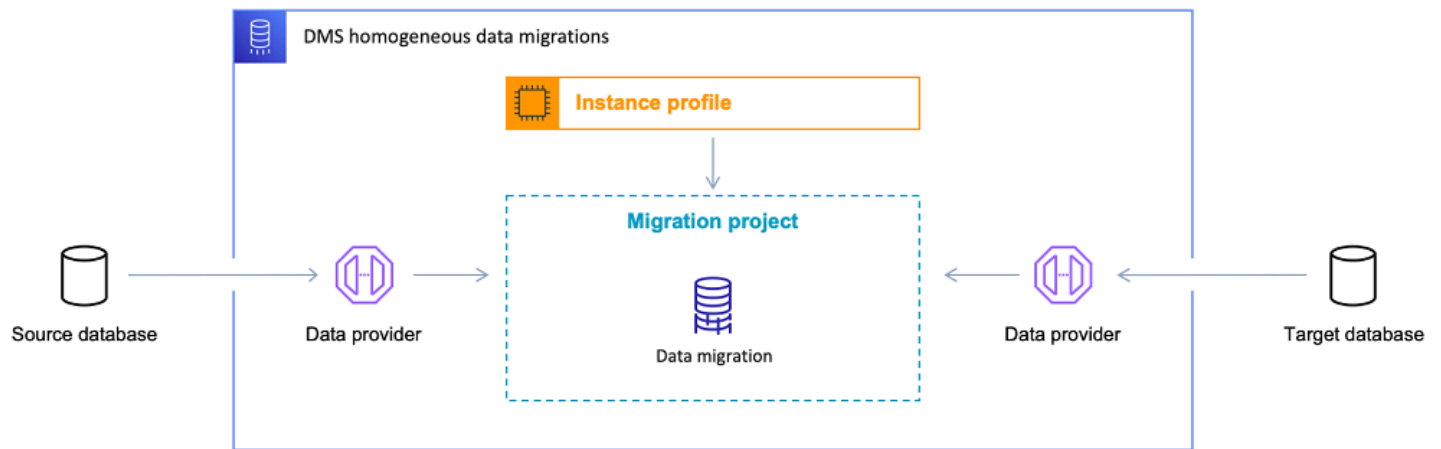
Homogene Datenmigrationen sind Serverless, was bedeutet, dass AWS DMS die Ressourcen, die für Ihre Migration benötigt werden, automatisch skaliert. Mit homogenen Datenmigrationen können Sie Daten, Tabellenpartitionen, Datentypen und sekundäre Objekte wie Funktionen, gespeicherte Prozeduren usw. migrieren.

Allgemein werden homogene Datenmigrationen mit Instance-Profilen, Datenanbietern und Migrationsprojekten durchgeführt. Wenn Sie ein Migrationsprojekt mit kompatiblen Quell- und Zieldatenanbietern desselben Typs erstellen, stellt AWS DMS eine Serverless-Umgebung bereit, in der Ihre Datenmigration ausgeführt wird. Als Nächstes stellt AWS DMS eine Verbindung zum Quelldatenanbieter her, liest die Quelldaten, speichert die Dateien auf der Festplatte und stellt die Daten mithilfe systemeigener Datenbank-Tools wieder her. Weitere Informationen zu Instance-Profilen, Datenanbietern und Migrationsprojekten finden Sie unter [Arbeit mit Datenanbietern, Instanzprofilen und Migrationsprojekten in AWS DMS](#).

Eine Liste der unterstützten Quelldatenbanken finden Sie unter [Quellen für homogene DMS-Datenmigrationen](#).

Eine Liste der unterstützten Zieldatenbanken finden Sie unter [Ziele für homogene DMS-Datenmigrationen](#).

Das folgende Diagramm veranschaulicht, wie homogene Datenmigrationen funktionieren.



Die folgenden Abschnitte enthalten Informationen zur Verwendung homogener Datenmigrationen.

Themen

- [Unterstützte AWS-Regionen](#)
- [Features](#)
- [Einschränkungen für homogene Datenmigrationen](#)
- [Überblick über den homogenen Datenmigrationsprozess in AWS DMS](#)
- [Einrichtung homogener Datenmigrationen in AWS DMS](#)
- [Erstellen von Quelldatenanbietern für homogene Datenmigrationen in AWS DMS](#)
- [Erstellung von Zieldatenanbietern für homogene Datenmigrationen in AWS DMS](#)
- [Durchführung homogener Datenmigrationen in AWS DMS](#)
- [Problembehandlung bei homogenen Datenmigrationen in AWS DMS](#)

Unterstützte AWS-Regionen

Sie können homogene Datenmigrationen in den folgenden AWS-Regionen ausführen.

Name der Region	Region
USA Ost (Nord-Virginia)	us-east-1
USA Ost (Ohio)	us-east-2
USA West (Oregon)	us-west-2

Name der Region	Region
Asien-Pazifik (Tokio)	ap-northeast-1
Asien-Pazifik (Singapur)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Europa (Frankfurt)	eu-central-1
Europa (Stockholm)	eu-north-1
Europa (Irland)	eu-west-1

Features

Homogene Datenmigrationen bieten die folgenden Features:

- AWS DMS verwaltet automatisch die Computing- und Speicherressourcen in der AWS Cloud, die für homogene Datenmigrationen erforderlich sind. AWS DMS stellt diese Ressourcen in einer Serverless-Umgebung bereit, wenn Sie eine Datenmigration starten.
- AWS DMS verwendet native Datenbank-Tools, um eine vollautomatische Migration zwischen Datenbanken desselben Typs zu initiieren.
- Sie können homogene Datenmigrationen verwenden, um Ihre Daten sowie die sekundären Objekte wie Partitionen, Funktionen, gespeicherte Prozeduren usw. zu migrieren.
- Sie können homogene Datenmigrationen in den folgenden drei Migrationsmodi ausführen: Volllast, fortlaufende Replikation und Volllast mit fortlaufender Replikation.
- Für homogene Datenmigrationen können Sie On-Premises-Amazon-EC2- und Amazon-RDS-Datenbanken als Quelle verwenden. Sie können Amazon RDS oder Amazon Aurora als Migrationsziel für homogene Datenmigrationen wählen.

Einschränkungen für homogene Datenmigrationen

Bei der Verwendung homogener Datenmigrationen gelten die folgenden Einschränkungen:

- Homogene Datenmigrationen unterstützen keine Auswahlregeln. Auch können Sie keine Transformationsregeln verwenden, um den Datentyp von Spalten zu ändern, Objekte von einem Schema in ein anderes zu verschieben und Objektnamen zu bearbeiten.
- Homogene Datenmigrationen bieten kein integriertes Tool zur Datenvalidierung.
- Wenn Sie homogene Datenmigrationen mit PostgreSQL verwenden, migriert AWS DMS Ansichten als Tabellen in Ihre Zieldatenbank.
- Homogene Datenmigrationen erfassen keine Änderungen auf Schemaebene während einer laufenden Datenreplikation. Wenn Sie in Ihrer Quelldatenbank eine neue Tabelle erstellen, kann AWS DMS diese Tabelle nicht migrieren. Um diese neue Tabelle zu migrieren, starten Sie Ihre Datenmigration neu.
- Sie können keine homogenen Datenmigrationen in AWS DMS verwenden, um Daten von einer höheren Datenbankversion zu einer niedrigeren Datenbankversion zu migrieren.
- Sie können keine homogenen Datenmigrationen in der CLI oder API verwenden.
- Homogene Datenmigrationen unterstützen den Aufbau einer Verbindung mit Datenbank-Instances in sekundären VPC-CIDR-Bereichen nicht.
- Sie können den 8081-Port nicht für homogene Migrationen von Ihren Datenanbietern verwenden.

Überblick über den homogenen Datenmigrationsprozess in AWS DMS

Sie können homogene Datenmigrationen in AWS DMS verwenden, um Daten zwischen zwei Datenbanken desselben Typs zu migrieren. Verwenden Sie den folgenden Workflow, um eine Datenmigration zu erstellen und auszuführen.

1. Erstellen Sie die erforderliche AWS Identity and Access Management (IAM)-Richtlinie und -Rolle. Weitere Informationen finden Sie unter [Erstellen von IAM-Ressourcen](#).
2. Konfigurieren Sie Ihre Quell- und Zieldatenbanken und erstellen Sie Datenbankbenutzer mit den Mindestberechtigungen, die für homogene Datenmigrationen in AWS DMS erforderlich sind. Weitere Informationen finden Sie unter [Erstellen von Quelldatenanbietern](#) und [Erstellen von Zieldatenanbietern](#).
3. Speichern Sie Ihre Quell- und Zieldatenbankanmeldedaten in AWS Secrets Manager. Weitere Informationen finden Sie unter [Schritt 1: Erstellen des Secrets](#) im AWS Secrets Manager-Benutzerhandbuch.

4. Erstellen Sie eine Subnetzgruppe, ein Instance-Profil und Datenanbieter in der AWS DMS-Konsole. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#), [Erstellen von Instance-Profilen](#) und [Erstellen von Datenanbietern](#).
5. Erstellen Sie ein Migrationsprojekt, indem Sie die Ressourcen verwenden, die Sie im vorherigen Schritt erstellt haben. Weitere Informationen finden Sie unter [Erstellen von Migrationsprojekten](#).
6. Erstellen, konfigurieren und starten Sie eine Datenmigration. Weitere Informationen finden Sie unter [Erstellen einer Datenmigration](#).
7. Nachdem Sie den Volllast-Vorgang oder die fortlaufende Replikation abgeschlossen haben, können Sie zur Verwendung Ihrer neuen Zieldatenbank übergehen.
8. Bereinigen Ihrer Ressourcen Amazon beendet Ihre Datenmigration in Ihrem Migrationsprojekt innerhalb von drei Tagen nach Abschluss der Migration. Sie müssen jedoch Ressourcen wie das Instance-Profil, die Datenanbieter, die IAM-Richtlinie und -Rolle sowie die Secrets in AWS Secrets Manager manuell löschen.

Weitere Informationen zu homogenen Datenmigrationen finden Sie in AWS DMS, der schrittweisen Anleitung zur Migration von [Migrationen von PostgreSQL zu Amazon RDS für PostgreSQL](#).

[Dieses Video](#) stellt die Benutzeroberfläche für homogene Datenmigrationen in AWS DMS vor und hilft Ihnen, sich mit diesem Feature vertraut zu machen.

Einrichtung homogener Datenmigrationen in AWS DMS

Führen Sie die folgenden erforderlichen Aufgaben aus, um homogene Datenmigrationen in AWS DMS einzurichten.

Themen

- [Erstellen der erforderlichen IAM-Ressourcen für homogene Datenmigrationen in AWS DMS](#)
- [Einrichtung eines Netzwerks für homogene Datenmigrationen in AWS DMS](#)

Erstellen der erforderlichen IAM-Ressourcen für homogene Datenmigrationen in AWS DMS

Um homogene Datenmigrationen durchzuführen, müssen Sie eine IAM-Richtlinie und eine IAM-Rolle in Ihrem Konto erstellen, um mit anderen AWS-Services interagieren zu können. In diesem Abschnitt erstellen Sie diese erforderlichen IAM-Ressourcen.

Themen

- [Erstellen einer IAM-Richtlinie für homogene Datenmigrationen in AWS DMS](#)
- [Erstellen einer IAM-Rolle für homogene Datenmigrationen in AWS DMS](#)

Erstellen einer IAM-Richtlinie für homogene Datenmigrationen in AWS DMS

Damit Sie auf Ihre Datenbanken zugreifen und Daten migrieren können, erstellt AWS DMS eine Serverless-Umgebung für homogene Datenmigrationen. In dieser Umgebung benötigt AWS DMS den Zugriff auf VPC-Peering, Routing-Tabellen, Sicherheitsgruppen und andere AWS-Ressourcen. AWS DMS speichert außerdem Protokolle, Metriken und den Fortschritt für jede Datenmigration in Amazon CloudWatch. Um ein Datenmigrationsprojekt zu erstellen, benötigt AWS DMS den Zugriff auf diese Services.

In diesem Schritt erstellen Sie eine IAM-Richtlinie, die AWS DMS den Zugriff auf Amazon-EC2- und CloudWatch-Ressourcen ermöglicht. Anschließend erstellen Sie eine IAM-Rolle und fügen dieser diese Richtlinie an.

So erstellen Sie eine IAM-Richtlinie für homogene Datenmigrationen in AWS DMS

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Policies.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte JSON aus.
5. Fügen Sie den folgenden JSON-Code in den Editor ein.

```
{
  "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. Wählen Sie Weiter: Tags und Weiter: Prüfen aus.

7. Geben Sie unter Name* **HomogeneousDataMigrationsPolicy** ein und wählen Sie dann Richtlinie erstellen aus.

Erstellen einer IAM-Rolle für homogene Datenmigrationen in AWS DMS

In diesem Schritt erstellen Sie eine IAM-Rolle, die AWS DMS den Zugriff auf AWS Secrets Manager, Amazon EC2 und CloudWatch ermöglicht.

So erstellen Sie eine IAM-Rolle für homogene Datenmigrationen in AWS DMS

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Roles.
3. Wählen Sie Create role (Rolle erstellen) aus.
4. Wählen Sie auf der Seite Vertrauenswürdige Entität auswählen unter Typ der vertrauenswürdigen Entität die Option AWS-Service aus. Unter Anwendungsfälle für andere AWS-Services wählen Sie DMS.
5. Markieren Sie das Kontrollkästchen DMS und wählen Sie Weiter.
6. Wählen Sie auf der Seite Berechtigungen hinzufügen die Option HomogeneousDataMigrationsPolicy aus, die Sie zuvor erstellt haben. Wählen Sie außerdem SecretsManagerReadWrite. Wählen Sie Weiter.
7. Geben Sie auf der Seite Rollename festlegen und überprüfen **HomogeneousDataMigrationsRole** für Rollename ein und wählen Sie Rolle erstellen aus.
8. Geben Sie auf der Seite Rollen **HomogeneousDataMigrationsRole** für Rollename ein. Wählen Sie HomogeneousDataMigrationsRole aus.
9. Wählen Sie auf der Seite HomogeneousDataMigrationsRole die Registerkarte Vertrauensbeziehungen aus. Wählen Sie Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.
10. Fügen Sie auf der Seite Vertrauensrichtlinie bearbeiten den folgenden JSON-Code in den Editor ein und ersetzen Sie den vorhandenen Text.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
```

```
    "Principal": {
      "Service": [
        "dms-data-migrations.amazonaws.com",
        "dms.your_region.amazonaws.com"
      ]
    },
    "Action": "sts:AssumeRole"
  }
]
```

Ersetzen Sie im vorherigen Beispiel *your_region* durch den Namen Ihrer AWS-Region.

Die vorherige ressourcenbasierte Richtlinie gewährt AWS DMS-Service-Prinzipalen Berechtigungen zur Ausführung von Aufgaben gemäß der von AWS verwalteten Richtlinie `SecretsManagerReadWrite` und der kundenverwalteten Richtlinie `HomogeneousDataMigrationsPolicy`.

11. Wählen Sie Update policy.

Einrichtung eines Netzwerks für homogene Datenmigrationen in AWS DMS

AWS DMS erstellt eine Serverless-Umgebung für homogene Datenmigrationen in einer Virtual Private Cloud (VPC) basierend auf dem Amazon-VPC-Service. Wenn Sie Ihr Instance-Profil erstellen, geben Sie die VPC an, die Sie verwenden möchten. Sie können Ihre Standard-VPC für Ihr Konto und Ihre AWS-Region verwenden oder eine neue VPC erstellen.

AWS DMS stellt für jede Datenmigration eine VPC-Peering-Verbindung mit der VPC her, die Sie für Ihr Instance-Profil verwenden. Als Nächstes fügt AWS DMS der Sicherheitsgruppe, die Ihrem Instance-Profil zugeordnet ist, den CIDR-Block hinzu. Da AWS DMS Ihrem Instance-Profil eine öffentliche IP-Adresse anfügt, haben alle Ihre Datenmigrationen, die dasselbe Instance-Profil verwenden, dieselbe öffentliche IP-Adresse. Wenn Ihre Datenmigration angehalten wird oder fehlschlägt, wird die VPC-Peering-Verbindung von AWS DMS gelöscht.

Um zu vermeiden, dass sich der CIDR-Block mit der VPC Ihres Instance-Profiles VPC überschneidet, verwendet AWS DMS das /24-Präfix aus einem der folgenden CIDR-Blöcke: `10.0.0.0/8`, `172.16.0.0/12` und `192.168.0.0/16`. Wenn Sie beispielsweise drei Datenmigrationen parallel ausführen, verwendet AWS DMS die folgenden CIDR-Blöcke, um eine VPC-Peering-Verbindung herzustellen.

- 192.168.0.0/24 – für die erste Datenmigration
- 192.168.1.0/24 – für die zweite Datenmigration
- 192.168.2.0/24 – für die dritte Datenmigration

Sie können verschiedene Netzwerkkonfigurationen verwenden, um die Interaktion zwischen Ihren Quell- und Zieldatenbanken und AWS DMS einzurichten. Für die fortlaufende Datenreplikation müssen Sie außerdem die Interaktion zwischen Ihren Quell- und Zieldatenbanken einrichten. Diese Konfigurationen hängen vom Standort Ihres Quelldatenanbieters und Ihren Netzwerkeinstellungen ab. Die folgenden Abschnitte enthalten Beschreibungen gängiger Netzwerkkonfigurationen.

Themen

- [Verwendung einer einzigen VPC für Quell- und Zieldatenanbieter](#)
- [Verwendung verschiedener VPCs für Quell- und Zieldatenanbieter](#)
- [Verwenden eines On-Premises-Quelldatenanbieters](#)
- [Konfiguration der fortlaufenden Datenreplikation](#)

Verwendung einer einzigen VPC für Quell- und Zieldatenanbieter

In dieser Konfiguration stellt AWS DMS eine Verbindung zu Ihren Quell- und Zieldatenanbietern innerhalb des privaten Netzwerks her.

So konfigurieren Sie ein Netzwerk, wenn sich Ihre Quell- und Zieldatenanbieter in derselben VPC befinden

1. Erstellen Sie die Subnetzgruppe in der AWS DMS-Konsole mit der VPC und den Subnetzen, die Ihr Quell- und Zieldatenanbieter verwenden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).
2. Erstellen Sie das Instance-Profil in der AWS DMS-Konsole mit der VPC und der Subnetzgruppe, die Sie erstellt haben. Wählen Sie außerdem VPC-Sicherheitsgruppen aus, die Ihr Quell- und Zieldatenanbieter verwenden. Weitere Informationen finden Sie unter [Erstellen von Instance-Profilen](#).

Diese Konfiguration erfordert nicht, dass Sie die öffentliche IP-Adresse für Datenmigrationen verwenden.

Verwendung verschiedener VPCs für Quell- und Zieldatenanbieter

In dieser Konfiguration stellt AWS DMS eine Verbindung zu Ihren Quell- und Zieldatenanbietern innerhalb eines privaten Netzwerks her. Für einen anderen Datenanbieter verwendet AWS DMS ein öffentliches Netzwerk. Je nachdem, welchen Datenanbieter Sie in derselben VPC wie Ihr Instance-Profil haben, wählen Sie eine der folgenden Konfigurationen.

So konfigurieren Sie ein privates Netzwerk für Ihren Quelldatenanbieter und ein öffentliches Netzwerk für Ihren Zieldatenanbieter

1. Erstellen Sie die Subnetzgruppe in der AWS DMS-Konsole mit der VPC und den Subnetzen, die Ihr Quelldatenanbieter verwendet. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).
2. Erstellen Sie das Instance-Profil in der AWS DMS-Konsole mit der VPC und der Subnetzgruppe, die Sie erstellt haben. Wählen Sie außerdem VPC-Sicherheitsgruppen aus, die Ihr Quelldatenanbieter verwendet. Weitere Informationen finden Sie unter [Erstellen von Instance-Profilen](#).
3. Öffnen Sie Ihr Migrationsprojekt. Wählen Sie auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Notieren Sie sich die öffentliche IP-Adresse unter Konnektivität und Sicherheit auf der Registerkarte Details.
4. Erlauben Sie den Zugriff über die öffentliche IP-Adresse Ihrer Datenmigration in der Sicherheitsgruppe Ihrer Zieldatenbank. Weitere Informationen finden Sie unter [Zugriffskontrolle mit Sicherheitsgruppen](#) im Amazon-Relational-Database-Service-Benutzerhandbuch.

So konfigurieren Sie ein öffentliches Netzwerk für Ihren Quelldatenanbieter und ein privates Netzwerk für Ihren Zieldatenanbieter

1. Erstellen Sie die Subnetzgruppe in der AWS DMS-Konsole mit der VPC und den Subnetzen, die Ihr Zieldatenanbieter verwendet. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).
2. Erstellen Sie das Instance-Profil in der AWS DMS-Konsole mit der VPC und der Subnetzgruppe, die Sie erstellt haben. Wählen Sie außerdem VPC-Sicherheitsgruppen aus, die Ihr Zieldatenanbieter verwendet. Weitere Informationen finden Sie unter [Erstellen von Instance-Profilen](#).

3. Öffnen Sie Ihr Migrationsprojekt. Wählen Sie auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Notieren Sie sich die öffentliche IP-Adresse unter Konnektivität und Sicherheit auf der Registerkarte Details.
4. Erlauben Sie den Zugriff über die öffentliche IP-Adresse Ihrer Datenmigration in der Sicherheitsgruppe Ihrer Quelldatenbank. Weitere Informationen finden Sie unter [Zugriffskontrolle mit Sicherheitsgruppen](#) im Amazon-Relational-Database-Service-Benutzerhandbuch.

Verwenden eines On-Premises-Quelldatenanbieters

In dieser Konfiguration stellt AWS DMS eine Verbindung zu Ihrem Quelldatenanbieter innerhalb eines öffentlichen Netzwerks her. AWS DMS verwendet ein privates Netzwerk, um eine Verbindung zu Ihrem Zieldatenanbieter herzustellen.

So konfigurieren Sie ein Netzwerk für Ihren On-Premises-Quelldatenanbieter

1. Erstellen Sie die Subnetzgruppe in der AWS DMS-Konsole mit der VPC und den Subnetzen, die Ihr Zieldatenanbieter verwendet. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).
2. Erstellen Sie das Instance-Profil in der AWS DMS-Konsole mit der VPC und der Subnetzgruppe, die Sie erstellt haben. Wählen Sie außerdem VPC-Sicherheitsgruppen aus, die Ihr Zieldatenanbieter verwendet. Weitere Informationen finden Sie unter [Erstellen von Instance-Profilen](#).
3. Öffnen Sie Ihr Migrationsprojekt. Wählen Sie auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Notieren Sie sich die öffentliche IP-Adresse unter Konnektivität und Sicherheit auf der Registerkarte Details.
4. Erlauben Sie den Zugriff über auf Ihre Quelldatenbank über die öffentliche IP-Adresse Ihrer Datenmigration in AWS DMS.

AWS DMS erstellt Regeln für eingehenden oder ausgehenden Datenverkehr in VPC-Sicherheitsgruppen. Stellen Sie sicher, dass Sie diese Regeln nicht löschen, da dies dazu führen kann, dass Ihre Datenmigration fehlschlägt. Sie können Ihre eigenen Regeln in VPC-Sicherheitsgruppen konfigurieren. Wir empfehlen, den Regeln Beschreibungen hinzuzufügen, damit Sie sie verwalten können.

Konfiguration der fortlaufenden Datenreplikation

Um Datenmigrationen des Typs Volllast und Change Data Capture (CDC) oder Change Data Capture (CDC) auszuführen, müssen Sie die Verbindung zwischen Ihrer Quell- und Zieldatenbank zulassen.

So konfigurieren Sie eine Verbindung zwischen Ihren öffentlich zugänglichen Quell- und Zieldatenbanken

1. Notieren Sie sich die öffentlichen IP-Adressen Ihrer Quell- und Zieldatenbanken.
2. Erlauben Sie den Zugriff auf Ihre Quelldatenbank über die öffentliche IP-Adresse Ihrer Zieldatenbank.
3. Erlauben Sie den Zugriff auf Ihre Zieldatenbank über die öffentliche IP-Adresse Ihrer Quelldatenbank.

So konfigurieren Sie eine Verbindung zwischen Ihren privat in einer einzigen VPC zugänglichen Quell- und Zieldatenbanken

1. Notieren Sie sich die privaten IP-Adressen Ihrer Quell- und Zieldatenbanken.

Important

Wenn sich Ihre Quell- und Zieldatenbanken in unterschiedlichen VPCs oder in unterschiedlichen Netzwerken befinden, können Sie nur öffentliche IP-Adressen für Ihre Quell- und Zieldatenbanken verwenden. Sie können nur öffentliche Hostnamen oder IP-Adressen in Datenanbietern verwenden.

2. Erlauben Sie den Zugriff auf Ihre Quelldatenbank über die private IP-Adresse Ihrer Zieldatenbank.
3. Erlauben Sie den Zugriff auf Ihre Zieldatenbank über die private IP-Adresse Ihrer Quelldatenbank.

Erstellen von Quelldatenanbietern für homogene Datenmigrationen in AWS DMS

Sie können MySQL-kompatible, PostgreSQL- und MongoDB-kompatible Datenbanken als Quelldatenanbieter für [Homogene Datenbankmigrationen](#) in verwenden AWS DMS.

Informationen zu unterstützten Datenbankversionen finden Sie unter [Quelldatenanbieter für homogene DMS-Datenmigrationen](#).

Ihr Quelldatenanbieter kann eine On-Premises-, Amazon-EC2- oder Amazon-RDS-Datenbank sein.

Themen

- [Verwenden einer MySQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)
- [Verwenden einer PostgreSQL-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)
- [Verwenden einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)

Verwenden einer MySQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

Sie können eine MySQL-kompatible Datenbank (MySQL oder MariaDB) als Quelle für [Homogene Datenbankmigrationen](#) in AWS DMS verwenden. In diesem Fall kann Ihr Quelldatenanbieter eine On-Premises-, Amazon-EC2-, Amazon-RDS-Datenbank oder MariaDB-Datenbank sein.

Um homogene Datenmigrationen durchzuführen, müssen Sie einen Datenbankbenutzer mit den SELECT-Berechtigungen für alle Quelltabellen und sekundären Objekte für die Replikation verwenden. Für Change-Data-Capture (CDC)-Aufgaben muss dieser Benutzer auch über die REPLICATION CLIENT (BINLOG MONITOR für MariaDB-Versionen nach 10.5.2) und REPLICATION SLAVE-Berechtigungen verfügen. Für eine Datenmigration mit Volllast benötigen Sie diese beiden Berechtigungen nicht.

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer mit den erforderlichen Berechtigungen in Ihrer MySQL-Datenbank zu erstellen. Führen Sie die GRANT Abfragen für alle Datenbanken aus, die Sie zu migrieren 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'@'%';
```

Ersetzen Sie im vorherigen Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen. Wenn Ihre Quell-MySQL-Datenbankversion niedriger als 8.0 ist, können Sie den GRANT BACKUP_ADMIN-Befehl überspringen.

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer mit den erforderlichen Berechtigungen in Ihrer MariaDB-Datenbank zu erstellen. Führen Sie die GRANT-Abfragen für alle Datenbanken aus, die Sie zu migrieren AWS.

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';  
GRANT SELECT, RELOAD, LOCK TABLES, REPLICATION SLAVE, BINLOG MONITOR, SHOW VIEW ON *.*  
TO 'your_user'@'%';
```

Ersetzen Sie im vorherigen Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

In den folgenden Abschnitten werden spezifische Konfigurationsvoraussetzungen für selbstverwaltete und von AWS verwaltete MySQL-Datenbanken beschrieben.

Themen

- [Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen](#)
- [Verwenden einer von verwalteten MySQL AWS-kompatiblen Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)
- [Einschränkungen bei der Verwendung einer MySQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen](#)

Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen

In diesem Abschnitt wird beschrieben, wie Sie Ihre MySQL-kompatiblen Datenbanken konfigurieren, die On-Premises oder auf Amazon-EC2-Instances gehostet werden.

Prüfen Sie Version Ihrer MySQL- oder MariaDB-Quelldatenbank. Stellen Sie sicher, dass Ihre MySQL- oder MariaDB-Quelldatenbankversion AWS DMS unterstützt, wie unter [beschrieben Quellen für homogene DMS-Datenmigrationen](#).

Um CDC zu verwenden, stellen Sie sicher, dass die binäre Protokollierung aktiviert ist. Um die binäre Protokollierung zu aktivieren, müssen die folgenden Parameter in der `my.ini`- (Windows) oder `my.cnf`-Datei (UNIX) Ihrer MySQL- oder MariaDB-Datenbank konfiguriert werden.

Parameter	Wert
<code>server-id</code>	Legen Sie diesen Parameter auf einen Wert von 1 oder größer fest.
<code>log-bin</code>	Legen Sie den Pfad zur binären Protokolldatei fest, z. B. <code>log-bin=E:\MySQL_Logs\BinLog</code> . Schließen Sie nicht die Dateierweiterung ein.
<code>binlog_format</code>	Legen Sie diesen Parameter auf <code>ROW</code> fest. Wir empfehlen, diese Einstellung während der Replikation zu verwenden, da es in gewissen Fällen zu Inkonsistenzen bei der Replikation von Daten auf das Ziel kommen kann, wenn <code>binlog_format</code> auf <code>STATEMENT</code> gesetzt ist. Die Datenbank-Engine schreibt auch ähnlich inkonsistente Daten in das Ziel, wenn <code>binlog_format</code> auf <code>MIXED</code> gesetzt ist, da die Datenbank-Engine automatisch zur <code>STATEMENT</code> -basierten Protokollierung wechselt.
<code>expire_logs_days</code>	Legen Sie diesen Parameter auf einen Wert von 1 oder größer fest. Um zu verhindern, dass zu viel Speicherplatz verwendet wird, empfehlen wir, nicht den Standardwert 0 zu verwenden.
<code>binlog_checksum</code>	Legen Sie diesen Parameter auf <code>NONE</code> fest.
<code>binlog_row_image</code>	Legen Sie diesen Parameter auf <code>FULL</code> fest.
<code>log_slave_updates</code>	Setzen Sie diesen Parameter auf <code>TRUE</code> , wenn Sie ein MySQL- oder MariaDB-Replikat als Quelle verwenden.

Verwenden einer von verwalteten MySQL AWS-kompatiblen Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

In diesem Abschnitt wird beschrieben, wie Sie Ihre Amazon-RDS-für-MySQL- und Amazon-RDS-für-MariaDB-Datenbank-Instances konfigurieren.

Wenn Sie eine von verwaltete MySQL AWS- oder MariaDB-Datenbank als Quelle für homogene Datenmigrationen in verwenden AWS DMS, stellen Sie sicher, dass Sie die folgenden Voraussetzungen für CDC erfüllen:

- Aktivieren Sie automatische Backups auf Instance-Ebene, um Binärprotokolle für RDS für MySQL und MariaDB zu aktivieren. Um Binärprotokolle für einen Aurora-MySQL-Cluster zu aktivieren, ändern Sie die Variable `binlog_format` in der Parametergruppe. Sie müssen keine automatischen Backups für einen Aurora-MySQL-Cluster aktivieren.

Stellen Sie den Parameter `binlog_format` auf `ROW` ein.

Weitere Informationen zum Aktivieren von automatischen Backups finden Sie unter [Aktivieren von automatisierten Backups](#) im Benutzerhandbuch für Amazon RDS.

Weitere Informationen zum Einrichten der Binärprotokollierung für eine Amazon-RDS-für-MySQL- oder MariaDB-Datenbank finden Sie unter [Einstellen des Binärprotokollierungsformats](#) im Benutzerhandbuch für Amazon RDS.

Weitere Informationen zum Einrichten der Binärprotokollierung für einen Aurora-MySQL-Cluster finden Sie unter [Wie aktiviere ich die Binärprotokollierung für meinen Amazon-Aurora-MySQL-Cluster?](#)

- Stellen Sie sicher, dass die Binärprotokolle für verfügbar sind AWS DMS. Da AWS-verwaltete MySQL- und MariaDB-Datenbanken die Binärprotokolle so schnell wie möglich bereinigen, sollten Sie die Dauer erhöhen, in der die Protokolle verfügbar bleiben. Um z. B. die Aufbewahrungszeit der Protokolle auf 24 Stunden zu verlängern, führen Sie den folgenden Befehl aus.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Stellen Sie den Parameter `binlog_row_image` auf `Full` ein.
- Stellen Sie den Parameter `binlog_checksum` auf `NONE` ein.
- Wenn Sie ein Lesereplikat von Amazon RDS MySQL oder MariaDB als Quelle verwenden, aktivieren Sie für das Lesereplikat Backups und stellen Sie sicher, dass der Parameter `log_slave_updates` auf `TRUE` gesetzt ist.

Einschränkungen bei der Verwendung einer MySQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen

Bei der Verwendung einer MySQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen gelten die folgenden Einschränkungen:

- MariaDB-Objekte wie Sequenzen werden bei homogenen Migrationsaufgaben nicht unterstützt.
- Die Migration von MariaDB zu Amazon RDS MySQL/Aurora MySQL schlägt möglicherweise aufgrund inkompatibler Objektunterschiede fehl.
- Der Benutzername, den Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Er darf 2 bis 64 Zeichen lang sein.
 - Er darf keine Leerzeichen enthalten.
 - Er kann die folgenden Zeichen enthalten: a-z, A-Z, 0-9, Unterstrich (_).
 - Er muss mit a-z oder A-Z beginnen.
- Das Passwort, das Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Es darf 1 bis 128 Zeichen lang sein.
 - Es darf keines der folgenden Zeichen enthalten: einfaches Anführungszeichen ('), doppeltes Anführungszeichen ("), Semikolon (;) oder Leerzeichen.

Verwenden einer PostgreSQL-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

Sie können eine PostgreSQL-Datenbank als Quelle für [Homogene Datenbankmigrationen](#) in AWS DMS verwenden. In diesem Fall kann Ihr Quelldatenanbieter eine On-Premises-, Amazon-EC2- oder RDS-für-PostgreSQL-Datenbank sein.

Um homogene Datenmigrationen auszuführen, erteilen Sie Superuser-Berechtigungen für den Datenbankbenutzer, den Sie in AWS DMS für Ihre PostgreSQL-Quelldatenbank angegeben haben. Das Benutzerkonto erfordert Superuser-Berechtigungen, um auf replikationsspezifische Funktionen in der Quelle zugreifen zu können. Für eine vollständige Datenmigration benötigt Ihr Datenbankbenutzer SELECT-Berechtigungen für Tabellen, um sie zu migrieren.

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer mit den erforderlichen Berechtigungen in Ihrer PostgreSQL-Quelldatenbank zu erstellen. Führen Sie die GRANT Abfrage für alle Datenbanken aus, die Sie zu migrieren 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;
```

Ersetzen Sie im vorherigen Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

In den folgenden Abschnitten werden spezifische Konfigurationsvoraussetzungen für selbstverwaltete und von AWS verwaltete PostgreSQL-Datenbanken beschrieben.

Themen

- [Verwenden einer selbstverwalteten PostgreSQL-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)
- [Verwenden einer von verwalteten PostgreSQL AWS-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)
- [Einschränkungen bei der Verwendung einer PostgreSQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen](#)

Verwenden einer selbstverwalteten PostgreSQL-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

In diesem Abschnitt wird beschrieben, wie Sie Ihre PostgreSQL-Datenbanken konfigurieren, die On-Premises oder auf Amazon-EC2-Instances gehostet werden.

Überprüfen Sie die Version Ihrer PostgreSQL-Quelldatenbank. Stellen Sie sicher, dass Ihre PostgreSQL-Quelldatenbankversion AWS DMS unterstützt, wie unter beschrieben [Quellen für homogene DMS-Datenmigrationen](#).

Homogene Datenmigrationen unterstützen die Erfassung von Datenänderungen (CDC) mithilfe logischer Replikation. Legen Sie die folgenden Parameter und Werte in der Konfigurationsdatei `postgresql.conf` fest, um die logische Replikation einer selbstverwalteten PostgreSQL-Quelldatenbank zu aktivieren.

- Setzen Sie `wal_level` auf `logical`.

- Legen Sie für `max_replication_slots` einen Wert größer als 1 fest.

Legen Sie den Wert `max_replication_slots` gemäß der Anzahl der Aufgaben fest, die Sie ausführen möchten. Wenn Sie beispielsweise fünf Aufgaben ausführen möchten, legen Sie mindestens fünf Slots fest.. Slots werden automatisch geöffnet, sobald eine Aufgabe gestartet wird und bleiben geöffnet, selbst wenn die Aufgabe nicht mehr ausgeführt wird. Löschen Sie offene Slots unbedingt manuell.

- Legen Sie für `max_wal_senders` einen Wert größer als 1 fest.

Der Parameter `max_wal_senders` legt die Anzahl der Aufgaben fest, die gleichzeitig ausgeführt werden können.

- Der Parameter `wal_sender_timeout` beendet Replikationsverbindungen, die länger als die angegebene Anzahl von Millisekunden inaktiv sind. Der Standardwert ist 60 000 Millisekunden (60 Sekunden). Wenn Sie den Wert auf 0 (Null) festlegen, wird der Timeout-Mechanismus deaktiviert. Dies ist eine gültige Einstellung für DMS.

Einige Parameter sind statisch und können nur beim Serverstart festgelegt werden. Alle Änderungen an deren Einträgen in der Konfigurationsdatei werden ignoriert, bis der Server neu gestartet wird. Weitere Informationen finden Sie in der [PostgreSQL-Dokumentation](#).

Verwenden einer von verwalteten PostgreSQL AWS-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

In diesem Abschnitt wird beschrieben, wie Sie Ihre Amazon-RDS-für-PostgreSQL-Datenbank-Instances konfigurieren.

Verwenden Sie das AWS Hauptbenutzerkonto für die PostgreSQL-DB-Instance als Benutzerkonto für den PostgreSQL-Quelldatenanbieter für homogene Datenmigrationen in AWS DMS. Das Masterbenutzerkonto hat die erforderlichen Rollen für die CDC-Einrichtung. Wenn Sie ein anderes Konto als das Hauptbenutzerkonto verwenden, muss das Konto über die `rds_superuser`-Rolle und die `rds_replication`-Rolle verfügen. Die Rolle `rds_replication` erteilt Berechtigungen zur Verwaltung von logischen Slots und zum Streamen von Daten mithilfe von logischen Slots.

Verwenden Sie das folgende Codebeispiel: Gewähren der Rollen `rds_superuser` und `rds_replication`.

```
GRANT rds_superuser to your_user;  
GRANT rds_replication to your_user;
```

Ersetzen Sie im vorherigen Beispiel *your_user* durch den Namen Ihres Datenbankbenutzers.

Um die logische Replikation zu aktivieren, setzen Sie den `rds.logical_replication`-Parameter in Ihrer DB-Parametergruppe auf 1. Für das Wirksamwerden dieses statischen Parameters ist ein Neustart der DB-Instance erforderlich.

Einschränkungen bei der Verwendung einer PostgreSQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen

Bei der Verwendung einer PostgreSQL-kompatiblen Datenbank als Quelle für homogene Datenmigrationen gelten die folgenden Einschränkungen:

- Der Benutzername, den Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Er darf 2 bis 64 Zeichen lang sein.
 - Er darf keine Leerzeichen enthalten.
 - Er kann die folgenden Zeichen enthalten: a-z, A-Z, 0-9, Unterstrich (_).
 - Er muss mit a-z oder A-Z beginnen.
- Das Passwort, das Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Es darf 1 bis 128 Zeichen lang sein.
 - Es darf keines der folgenden Zeichen enthalten: einfaches Anführungszeichen ('), doppeltes Anführungszeichen ("), Semikolon (;) oder Leerzeichen.

Verwenden einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

Sie können eine MongoDB-kompatible Datenbank als Quelle für homogene Datenmigrationen in verwenden AWS DMS. In diesem Fall kann Ihr Quelldatenanbieter eine On-Premises-, Amazon EC2-für-MongoDB-Datenbank oder Amazon DocumentDB-Datenbank (mit MongoDB-Kompatibilität) sein.

Informationen zu unterstützten Datenbankversionen finden Sie unter [Quelldatenanbieter für homogene DMS-Datenmigrationen](#).

In den folgenden Abschnitten werden spezifische Konfigurationsvoraussetzungen für selbstverwaltete MongoDB-Datenbanken und -verwaltete Amazon DocumentDB AWS-Datenbanken beschrieben.

Themen

- [Verwenden einer selbstverwalteten MongoDB-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)
- [Verwenden einer Amazon DocumentDB-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS](#)
- [Funktionen für die Verwendung einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen](#)
- [Einschränkungen bei der Verwendung einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen](#)
- [Bewährte Methoden für die Verwendung einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen](#)

Verwenden einer selbstverwalteten MongoDB-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

In diesem Abschnitt wird beschrieben, wie Sie Ihre MongoDB-Datenbanken konfigurieren, die On-Premises oder auf Amazon EC2 gehostet werden.

Überprüfen Sie die Version Ihrer MongoDB-Quelldatenbank. Stellen Sie sicher, dass Ihre MongoDB-Quelldatenbankversion AWS DMS unterstützt, wie unter beschrieben [Quelldatenanbieter für homogene DMS-Datenmigrationen](#).

Um homogene Datenmigrationen mit einer MongoDB-Quelle auszuführen, können Sie entweder ein Benutzerkonto mit Root-Berechtigungen oder einen Benutzer mit Berechtigungen nur für die zu migrierende Datenbank erstellen. Weitere Informationen zur Benutzererstellung finden Sie unter [Erforderliche Berechtigungen für die Verwendung von MongoDB als Quelle für AWS DMS](#).

Um die laufende Replikation oder CDC mit MongoDB zu verwenden, AWS DMS benötigt Zugriff auf das MongoDB-Operationsprotokoll (Oplog). Weitere Informationen finden Sie unter [Konfigurieren eines MongoDB-Replikatsatzes für CDC](#).

Informationen zu MongoDB-Authentifizierungsmethoden finden Sie unter [Sicherheitsanforderungen für die Verwendung von MongoDB als Quelle für AWS DMS](#).

Für MongoDB als Quelle unterstützen homogene Datenmigrationen alle Datentypen, die Amazon DocumentDB unterstützt.

Für MongoDB als Quelle müssen Sie Benutzeranmeldeinformationen in Secrets Manager im Klartext bereitstellen, indem Sie den Secret-Typ *Andere Art von Secrets* verwenden. Weitere Informationen finden Sie unter [Verwenden von Secrets für den Zugriff auf AWS Database Migration Service - Endpunkte](#).

Das folgende Codebeispiel zeigt, wie Datenbankgeheimnisse mit Klartext gespeichert werden.

```
{
  "username": "dbuser",
  "password": "dbpassword"
}
```

Verwenden einer Amazon DocumentDB-Datenbank als Quelle für homogene Datenmigrationen in AWS DMS

In diesem Abschnitt wird beschrieben, wie Sie Ihre Amazon DocumentDB-Datenbank-Instances für die Verwendung als Quelle für homogene Datenmigrationen konfigurieren.

Verwenden Sie den Hauptbenutzernamen für die Amazon DocumentDB-Instance als Benutzerkonto für den MongoDB-kompatiblen Quelldatenanbieter für homogene Datenmigrationen in AWS DMS. Das Masterbenutzerkonto hat die erforderlichen Rollen für die CDC-Einrichtung. Wenn Sie ein anderes Konto als das Hauptbenutzerkonto verwenden, muss das Konto über die Stammrolle verfügen. Weitere Informationen zur Benutzererstellung als Stammkonto finden Sie unter [Festlegen von Berechtigungen zur Verwendung von Amazon DocumentDB als Quelle](#).

Um die logische Replikation zu aktivieren, legen Sie den `change_stream_log_retention_duration` Parameter in Ihrer Datenbankparametergruppe auf eine für Ihre Transaktions-Workload geeignete Einstellung fest. Um diesen statischen Parameter zu ändern, müssen Sie Ihre DB-Instance neu starten, damit sie wirksam wird. Bevor Sie die Datenmigration für alle Aufgabentypen starten, einschließlich Nur Volllast, aktivieren Sie Amazon DocumentDB-Änderungsstreams für alle Sammlungen innerhalb einer bestimmten Datenbank oder nur für ausgewählte Sammlungen. Weitere Informationen zum Aktivieren von Änderungsstreams für Amazon DocumentDB finden Sie unter [Aktivieren von Änderungsstreams](#) im Amazon DocumentDB-Entwicklerhandbuch.

Note

AWS DMS verwendet den Amazon DocumentDB-Änderungsstream, um Änderungen während der laufenden Replikation zu erfassen. Wenn Amazon DocumentDB die Datensätze

aus dem Änderungsstream löscht, bevor DMS sie liest, schlagen Ihre Aufgaben fehl. Wir empfehlen, den `change_stream_log_retention_duration` Parameter so einzustellen, dass Änderungen mindestens 24 Stunden lang beibehalten werden.

Um Amazon DocumentDB für die homogene Datenmigration zu verwenden, speichern Sie Benutzeranmeldeinformationen in Secrets Manager unter Anmeldeinformationen für die Amazon DocumentDB-Datenbank .

Funktionen für die Verwendung einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen

- Sie können alle sekundären Indizes migrieren, die Amazon DocumentDB während der Vollastphase unterstützt.
- AWS DMS migriert Sammlungen parallel. homogene Datenmigrationen berechnen Segmente zur Laufzeit basierend auf der durchschnittlichen Größe jedes Dokuments in der Sammlung, um maximale Leistung zu erzielen.
- DMS kann die sekundären Indizes replizieren, die Sie in der CDC-Phase erstellen. DMS unterstützt diese Funktion in MongoDB Version 6.0.
- DMS unterstützt Dokumente mit einer Verschachtelungsebene von mehr als 97.

Einschränkungen bei der Verwendung einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen

- Dokumente dürfen keine Feldnamen mit einem \$ Präfix haben.
- AWS DMS unterstützt keine Zeitreihenerfassungsmigration.
- AWS DMS unterstützt während der CDC-Phase keine `create-drop`, - oder `-rename collectionDDL`-Ereignisse.
- AWS DMS unterstützt keine inkonsistenten Datentypen in der Sammlung für das `_id` Feld . Die folgende nicht unterstützte Sammlung hat beispielsweise mehrere Datentypen für das `_id` Feld .

```
rs0 [direct: primary] test> db.collection1.aggregate([
...   {
...     $group: {
...       _id: { $type: "$_id" },
...       count: { $sum: 1 }
...     }
...   }
... ])
```

```
...    }  
...    }  
...  ])  
[ { _id: 'string', count: 6136 }, { _id: 'objectId', count: 848033 } ]
```

- Für reine CDC-Aufgaben unterstützt AWS DMS nur den `immediate` Startmodus.
- AWS DMS unterstützt keine Dokumente mit ungültigen UTF8-Zeichen.
- AWS DMS unterstützt keine Sharded-Sammlungen.

Bewährte Methoden für die Verwendung einer MongoDB-kompatiblen Datenbank als Quelle für homogene Datenmigrationen

- Für mehrere große Datenbanken und Sammlungen, die auf derselben MongoDB-Instance gehostet werden, empfehlen wir Ihnen, Auswahlregeln für jede Datenbank und Sammlung zu verwenden, um die Aufgabe auf mehrere Datenmigrationsaufgaben und Projekte aufzuteilen. Sie können Ihre Datenbank- und Sammlungsbereiche auf maximale Leistung einstellen.

Erstellung von Zieldatenanbietern für homogene Datenmigrationen in AWS DMS

Sie können MySQL-kompatible, PostgreSQL- und Amazon DocumentDB DocumentDB-Datenbanken als Zieldatenanbieter für homogene Datenmigrationen in verwenden. AWS DMS

Informationen zu unterstützten [Zieldatenanbieter für homogene DMS-Datenmigrationen](#) Datenbankversionen finden Sie unter.

Ihr Ziel-Datenanbieter kann eine Amazon-RDS-DB-Instance oder ein Amazon-Aurora-DB-Cluster sein. Beachten Sie, dass die Datenbankversion Ihres Zieldatenproviders der Datenbankversion Ihres Quelldatenproviders entsprechen oder höher sein muss.

Themen

- [Verwendung einer MySQL-kompatiblen Datenbank als Ziel für homogene Datenmigrationen in AWS DMS](#)
- [Verwendung einer PostgreSQL-Datenbank als Ziel für homogene Datenmigrationen in AWS DMS](#)
- [Verwendung einer Amazon DocumentDB DocumentDB-Datenbank als Ziel für homogene Datenmigrationen in AWS DMS](#)

Verwendung einer MySQL-kompatiblen Datenbank als Ziel für homogene Datenmigrationen in AWS DMS

Sie können eine MySQL-kompatible Datenbank als Migrationsziel für homogene Datenmigrationen in AWS DMS verwenden.

AWS DMS erfordert bestimmte Berechtigungen, um Daten in Ihre Amazon RDS for MySQL- oder MariaDB- oder Amazon Aurora MySQL-Zieldatenbank zu migrieren. Verwenden Sie das folgende Skript, um einen Datenbankbenutzer mit den erforderlichen Berechtigungen in Ihrer MySQL-Zieldatenbank zu erstellen.

```
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'@'%' ;
```

Ersetzen Sie im vorherigen Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Verwenden Sie das folgende Skript, um einen Datenbankbenutzer mit den erforderlichen Berechtigungen in Ihrer MariaDB-Datenbank zu erstellen. Führen Sie die GRANT-Abfragen für alle Datenbanken aus, zu denen Sie migrieren. 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'@'%' ;
```

Ersetzen Sie im vorherigen Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Note

Wenn Sie in Amazon RDS die automatische Sicherung für eine MySQL/Maria-Datenbank-Instance aktivieren, aktivieren Sie auch die Binärprotokollierung. Wenn diese Einstellungen aktiviert sind, kann Ihre Datenmigrationsaufgabe beim Erstellen sekundärer Objekte wie Funktionen, Prozeduren und Auslöser in der Zieldatenbank mit dem folgenden

Fehler fehlschlagen. Wenn in Ihrer Zieldatenbank die Binärprotokollierung aktiviert ist, legen Sie vor dem Starten der Aufgabe in der Datenbankparametergruppe den Wert `log_bin_trust_function_creators` auf `true` fest.

```
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)
```

Einschränkungen bei der Verwendung einer MySQL-kompatiblen Datenbank als Ziel für homogene Datenmigrationen

Bei der Verwendung einer MySQL-kompatiblen Datenbank als Ziel für homogene Datenmigrationen gelten die folgenden Einschränkungen:

- Der Benutzername, den Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Er darf 2 bis 64 Zeichen lang sein.
 - Er darf keine Leerzeichen enthalten.
 - Er kann die folgenden Zeichen enthalten: a-z, A-Z, 0-9, Unterstrich (`_`).
 - Er muss mit a-z oder A-Z beginnen.
- Das Passwort, das Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Es darf 1 bis 128 Zeichen lang sein.
 - Es darf keines der folgenden Zeichen enthalten: einfaches Anführungszeichen (`'`), doppeltes Anführungszeichen (`"`), Semikolon (`;`) oder Leerzeichen.

Verwendung einer PostgreSQL-Datenbank als Ziel für homogene Datenmigrationen in AWS DMS

Sie können eine PostgreSQL-Datenbank als Migrationsziel für homogene Datenmigrationen in AWS DMS verwenden.

AWS DMS erfordert bestimmte Berechtigungen, um Daten in Ihre Amazon RDS for PostgreSQL- oder Amazon Aurora PostgreSQL-Zieldatenbank zu migrieren. Verwenden Sie das folgende

Skript, um einen Datenbankbenutzer mit den erforderlichen Berechtigungen in Ihrer PostgreSQL-Zieldatenbank zu erstellen.

```
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;
```

Ersetzen Sie im vorherigen Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Um die logische Replikation für Ihr RDS-für-PostgreSQL-Ziel zu aktivieren, setzen Sie den `rds.logical_replication`-Parameter in Ihrer DB-Parametergruppe auf 1. Für das Wirksamwerden dieses statischen Parameters ist ein Neustart der DB-Instance oder des DB-Clusters erforderlich. Einige Parameter sind statisch und können nur beim Serverstart festgelegt werden. AWS DMS ignoriert Änderungen an ihren Einträgen in der DB-Parametergruppe, bis Sie den Server neu starten.

PostgreSQL verwendet Auslöser, um Fremdschlüsselbeschränkungen zu implementieren. AWS DMS lädt während der Vollladephase jede Tabelle einzeln. Wir empfehlen, dass Sie Fremdschlüsseleinschränkungen für Ihre Zieldatenbank während des Volllast-Vorgangs deaktivieren. Verwenden Sie dazu eines der folgenden Verfahren.

- Deaktivieren Sie vorübergehend alle Auslöser für Ihrer Instance und beenden Sie den Volllast-Vorgang.
- Ändern Sie den Wert des `session_replication_role`-Parameters zu PostgreSQL.

Ein Auslöser kann sich zu jedem beliebigen Zeitpunkt in einem der folgenden Zustände befinden: `origin`, `replica`, `always` und `disabled`. Wenn Sie den `session_replication_role`-Parameter auf `replica` setzen, sind nur Auslöser im `replica`-Status aktiv. Andernfalls bleiben die Auslöser inaktiv.

Einschränkungen bei der Verwendung einer PostgreSQL-kompatiblen Datenbank als Ziel für homogene Datenmigrationen

Bei der Verwendung einer PostgreSQL-kompatiblen Datenbank als Ziel für homogene Datenmigrationen gelten die folgenden Einschränkungen:

- Der Benutzername, den Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Er darf 2 bis 64 Zeichen lang sein.
 - Er darf keine Leerzeichen enthalten.
 - Er kann die folgenden Zeichen enthalten: a-z, A-Z, 0-9, Unterstrich (_).
 - Er muss mit a-z oder A-Z beginnen.
- Das Passwort, das Sie für die Verbindung mit Ihrer Datenquelle verwenden, hat die folgenden Einschränkungen:
 - Es darf 1 bis 128 Zeichen lang sein.
 - Es darf keines der folgenden Zeichen enthalten: einfaches Anführungszeichen ('), doppeltes Anführungszeichen ("), Semikolon (;) oder Leerzeichen.

Verwendung einer Amazon DocumentDB DocumentDB-Datenbank als Ziel für homogene Datenmigrationen in AWS DMS

Sie können eine Amazon DocumentDB DocumentDB-Datenbank (mit MongoDB-Kompatibilität) und einen DocumentDB Elastic-Cluster als Migrationsziel für homogene Datenmigrationen in verwenden. AWS DMS

Um homogene Datenmigrationen für ein Amazon DocumentDB DocumentDB-Ziel durchzuführen, können Sie entweder ein Benutzerkonto mit Administratorrechten oder einen Benutzer mit Lese-/Schreibberechtigungen nur für die zu migrierende Datenbank erstellen.

Homogene Datenmigrationen unterstützen alle BSON-Datentypen, die Amazon DocumentDB unterstützt. Eine Liste dieser Datentypen finden Sie unter [Datentypen](#) im Amazon DocumentDB Developer Guide.

Um die Shard-Funktionen des DocumentDB Elastic-Clusters für die Migration von Sammlungen ohne Sharded aus der Quelle zu verwenden, erstellen Sie vor dem Start der Datenmigrationsaufgabe eine zu migrierende Shard-Sammlung. Weitere Informationen zur Shard-Erfassung in einem

Amazon DocumentDB Elastic-Cluster finden Sie unter [Schritt 5: Ihre Sammlung teilen](#) im Amazon DocumentDB DocumentDB-Entwicklerhandbuch.

AWS DMS unterstützt für ein Amazon DocumentDB DocumentDB-Ziel die Modi `none` oder `require SSL`.

Durchführung homogener Datenmigrationen in AWS DMS

Sie können [Homogene Datenbankmigrationen](#) in verwenden AWS DMS, um Daten von Ihrer Quelldatenbank zur entsprechenden Engine auf Amazon Relational Database Service (Amazon RDS), Amazon Aurora oder Amazon DocumentDB zu migrieren. AWS DMS automatisiert den Datenmigrationsprozess mithilfe nativer Datenbanktools in Ihren Quell- und Zieldatenbanken.

Nachdem Sie ein Instance-Profil und kompatible Datenanbieter für homogene Datenmigrationen erstellt haben, erstellen Sie ein Migrationsprojekt. Weitere Informationen finden Sie unter [Erstellen von Migrationsprojekten](#).

In den folgenden Abschnitten wird beschrieben, wie Sie homogene Datenmigrationen erstellen, konfigurieren und ausführen.

Themen

- [Erstellen einer Datenmigration in AWS DMS](#)
- [Auswahlregeln für homogene Datenmigrationen](#)
- [Verwaltung von Datenmigrationen in AWS DMS](#)
- [Überwachung von Datenmigrationen in AWS DMS](#)
- [Status homogener Datenmigrationen in AWS DMS](#)
- [Migrieren von Daten aus MySQL-Datenbanken mit homogenen Datenmigrationen in AWS DMS](#)
- [Migrieren von Daten aus PostgreSQL-Datenbanken mit homogenen Datenmigrationen in AWS DMS](#)
- [Migrieren von Daten aus MongoDB-Datenbanken mit homogenen Datenmigrationen in AWS DMS](#)

Erstellen einer Datenmigration in AWS DMS

Nachdem Sie ein Migrationsprojekt mit kompatiblen Datenanbietern desselben Typs erstellt haben, können Sie dieses Projekt für homogene Datenmigrationen verwenden. Weitere Informationen finden Sie unter [Erstellen von Migrationsprojekten](#).

Um mit der Verwendung homogener Datenmigrationen zu beginnen, erstellen Sie eine neue Datenmigration. Sie können mehrere homogene Datenmigrationen verschiedener Typen in einem einzigen Migrationsprojekt erstellen.

AWS DMS hat die maximale Anzahl von homogenen Datenmigrationen, die Sie für Ihr AWS-Konto erstellen können. Im folgenden Abschnitt finden Sie Informationen zu AWS DMS-Servicekontingenten [Kontingente für AWS Database Migration Service](#).

Bevor Sie eine Datenmigration erstellen, stellen Sie sicher, dass Sie die erforderlichen Ressourcen wie Ihre Quell- und Zieldatenbanken, eine IAM-Richtlinie und -Rolle, ein Instance-Profil und Datenanbieter einrichten. Weitere Informationen finden Sie unter [Erstellen von IAM-Ressourcen](#), [Erstellen von Instance-Profilen](#) und [Erstellen von Datenanbietern](#).

Außerdem sollten Sie keine homogenen Datenmigrationen verwenden, um Daten von einer höheren Datenbankversion zu einer niedrigeren Datenbankversion zu migrieren. Überprüfen Sie die Versionen der Datenbanken, die Sie für Quell- und Zieldatenanbieter verwenden, und aktualisieren Sie bei Bedarf Ihre Zieldatenbankversion.

So erstellen Sie eine Datenmigration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt und auf der Registerkarte Datenmigrationen Datenmigration erstellen.
4. Geben Sie unter Name einen Namen für Ihre Datenmigration ein. Stellen Sie sicher, dass Sie einen eindeutigen Namen für Ihre Datenmigration verwenden, damit Sie sie leicht identifizieren können.
5. Wählen Sie unter Replikationstyp den Typ der Datenmigration aus, den Sie konfigurieren möchten. Wählen Sie eine der folgenden Optionen aus:
 - Volllast – Migriert Ihre vorhandenen Quelldaten.
 - Volllast und Erfassung von Änderungsdaten (CDC) – migriert vorhandene Daten und fortlaufende Änderungen.
 - Change Data Capture (CDC) – Repliziert fortlaufende Änderungen.

6. Aktivieren Sie das Kontrollkästchen für CloudWatch Protokolle aktivieren, um Datenmigrationsprotokolle in Amazon zu speichern CloudWatch. Sollten Sie diese Option nicht wählen, können Sie die Protokolldateien nicht sehen, wenn Ihre Datenmigration fehlschlägt.
7. (Optional) Erweitern Sie den Abschnitt Advanced Settings (Erweiterte Einstellungen). Geben Sie unter Anzahl der Jobs die Anzahl der parallelen Threads ein, mit denen AWS DMS Ihre Quelldaten zum Ziel migrieren kann.
8. Wählen Sie unter IAM-Servicerolle die zuvor erstellte IAM-Rolle aus. Weitere Informationen finden Sie unter [Erstellen einer IAM-Rolle für homogene Datenmigrationen in AWS DMS](#).
9. Konfigurieren Sie den Startmodus für Datenmigrationen des Typs Change Data Capture (CDC). Wählen Sie eine der folgenden Optionen aus:

- Sofort – Startet die fortlaufende Replikation, wenn Sie Ihre Datenmigration starten.
- Verwenden eines systemeigenen Startpunkts – Startet die fortlaufende Replikation vom angegebenen Punkt aus.

Geben Sie für PostgreSQL-Datenbanken den Namen des logischen Replikationsslots unter Slot-Name und die Sequenznummer des Transaktionsprotokolls unter Systemeigener Startpunkt ein.

Geben Sie für MySQL-Datenbanken die Sequenznummer des Transaktionsprotokolls unter Protokoll-Sequenznummer (LSN) ein.

10. Konfigurieren Sie den Stoppmodus für Datenmigrationen des Typs Change Data Capture (CDC) oder Volllast und Change Data Capture (CDC). Wählen Sie eine der folgenden Optionen aus:
 - CDC nicht anhalten – AWS DMS setzt die fortlaufende Replikation fort, bis Sie Ihre Datenmigration beenden.
 - Verwendung eines Serverzeitpunkts – AWS DMS hält die fortlaufende Replikation zum angegebenen Zeitpunkt an.

Wenn Sie diese Option wählen, geben Sie für Datum und Uhrzeit des Anhaltens das Datum und die Uhrzeit ein, zu dem die fort laufende Replikation automatisch angehalten werden soll.

11. Wählen Sie Datenmigration erstellen aus.

AWS DMS erstellt Ihre Datenmigration und fügt sie der Liste auf der Registerkarte Datenmigrationen in Ihrem Migrationsprojekt hinzu. Hier können Sie den Status Ihrer Datenmigration sehen. Weitere Informationen finden Sie unter [Migrationsstatus](#).

⚠ Important

AWS DMS löscht bei Datenmigrationen des Typs Volllast und Volllast und Change Data Capture (CDC) alle Daten, Tabellen und anderen Datenbankobjekte in Ihrer Zieldatenbank. Stellen Sie sicher, dass Sie eine Sicherungskopie Ihrer Zieldatenbank haben.

Nachdem AWS DMS Ihre Datenmigration erstellt hat, wird der Status dieser Datenmigration auf Bereit gesetzt. Um Ihre Daten zu migrieren, müssen Sie die Datenmigration manuell starten. Wählen Sie dazu Ihre Datenmigration aus der Liste. Wählen Sie dann für Aktionen die Option Start. Weitere Informationen finden Sie unter [Verwaltung von Datenmigrationen](#).

Der erste Start einer homogenen Datenmigration erfordert einige Einstellungen. AWS DMS erstellt eine Serverless-Umgebung für Ihre Datenmigration. Dieser Vorgang kann bis zu 15 Minuten dauern. Nachdem Sie Ihre Datenmigration angehalten und neu gestartet haben, erstellt AWS DMS die Umgebung nicht erneut, so dass Sie schneller auf Ihre Datenmigration zugreifen können.

Auswahlregeln für homogene Datenmigrationen

Sie können Auswahlregeln verwenden, um das Schema, die Tabellen oder beide auszuwählen, die Sie in Ihre Replikation aufnehmen möchten.

ℹ Note

AWS DMS unterstützt nur Auswahlregeln für homogene Datenmigrationen, wenn eine MongoDB-kompatible Datenbank als Quelle verwendet wird.

Wählen Sie beim Erstellen der Datenmigrationsaufgabe Auswahlregel hinzufügen aus.

Geben Sie für die Regeleinstellungen die folgenden Werte an:

- Schema : Wählen Sie Schema eingeben aus.
- Schemaname: Geben Sie den Namen des Schemas an, das Sie replizieren möchten, oder verwenden Sie ihn % als Platzhalter.
- Tabellenname : : Geben Sie den Namen der Tabelle an, die Sie replizieren möchten, oder verwenden Sie % ihn als Platzhalter.

Standardmäßig ist die einzige Regelaktion, die DMS unterstützt `Include`, und das einzige Platzhalterzeichen, das DMS unterstützt, ist `%`.

Example Migrieren aller Tabellen in einem Schema

Im folgenden Beispiel werden alle Tabellen aus einem Schema mit dem Namen `dmsst` in Ihrer Quelle zu Ihrem Zielendpunkt migriert.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-action": "include",
      "object-locator": {
        "schema-name": "dmsst",
        "table-name": "%"
      },
      "filters": [],
      "rule-id": "1",
      "rule-name": "1"
    }
  ]
}
```

Example Migrieren einiger Tabellen in einem Schema

Im folgenden Beispiel werden alle Tabellen mit einem Namen, der mit `collectionTest` beginnt, von einem Schema mit dem Namen `dmsst` in Ihrer Quelle zu Ihrem Zielendpunkt migriert.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-action": "include",
      "object-locator": {
        "schema-name": "dmsst",
        "table-name": "collectionTest%"
      },
      "filters": [],
      "rule-id": "1",
      "rule-name": "1"
    }
  ]
}
```



```
}
```

Example Migrieren bestimmter Tabellen aus mehreren Schemata

Im folgenden Beispiel werden einige der Tabellen aus mehreren Schemata namens `dmsst` und `Test` in Ihrer Quelle auf Ihren Zielendpunkt migriert.

```
{
  "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"
    }
  ]
}
```

Verwaltung von Datenmigrationen in AWS DMS

Nachdem Sie eine Datenmigration erstellt haben, beginnt AWS DMS nicht automatisch mit der Migration von Daten. Sie starten eine Datenmigration bei Bedarf manuell.

Bevor Sie eine Datenmigration starten, können Sie alle Einstellungen Ihrer Datenmigration ändern. Nachdem Sie Ihre Datenmigration gestartet haben, können Sie den Replikationstyp nicht mehr ändern. Um einen anderen Replikationstyp zu verwenden, erstellen Sie eine neue Datenmigration.

So starten Sie eine Datenmigration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt aus. Wählen Sie auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Die Seite Übersicht für Ihre Datenmigration wird geöffnet.
4. Wählen Sie für Actions (Aktionen) die Option Start.

Danach erstellt AWS DMS eine Serverless-Umgebung für Ihre Datenmigration. Dieser Vorgang kann bis zu 15 Minuten dauern.

Nachdem Sie eine Datenmigration gestartet haben, setzt AWS DMS deren Status auf Wird gestartet. Der nächste Status, den AWS DMS für Ihre Datenmigration verwendet, hängt von der Art der Replikation ab, die Sie in den Datenmigrationseinstellungen auswählen. Weitere Informationen finden Sie unter [Migrationsstatus](#).

So ändern Sie eine Datenmigration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt aus. Wählen Sie auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Die Seite Übersicht für Ihre Datenmigration wird geöffnet.
4. Wählen Sie Ändern aus.
5. Konfigurieren Sie die Einstellungen für Ihre Datenmigration.

Important

Wenn Sie Ihre Datenmigration gestartet haben, können Sie den Replikationstyp nicht mehr ändern.

6. Um Ihre Datenmigrationsprotokolle in Amazon anzuzeigen CloudWatch, aktivieren Sie das Kontrollkästchen für CloudWatch Protokolle aktivieren.
7. Wählen Sie Änderungen speichern aus.

Nach dem Start einer Datenmigration durch AWS DMS können Sie sie anhalten. Wählen Sie dazu auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Wählen Sie dann für Aktionen die Option Stopp.

Nachdem Sie eine Datenmigration angehalten haben, setzt AWS DMS deren Status auf Wird angehalten. Als Nächstes setzt AWS DMS den Status dieser Datenmigration auf Angehalten. Nach dem Anhalten einer Datenmigration durch AWS DMS können Sie Ihre Datenmigration ändern, fortsetzen, neu starten oder löschen.

Um die Datenreplikation fortzusetzen, wählen Sie auf der Registerkarte Datenmigrationen die Datenmigration aus, die Sie angehalten haben. Wählen Sie als Nächstes für Aktionen die Option Verarbeitung fortsetzen aus.

Um die Datenladung fortzusetzen, wählen Sie auf der Registerkarte Datenmigrationen die Datenmigration aus, die Sie angehalten haben. Wählen Sie als Nächstes für Aktionen die Option Neu starten aus. AWS DMS löscht alle Daten aus Ihrer Zieldatenbank und startet die Datenmigration von vorn.

Sie können eine Datenmigration löschen, die Sie angehalten oder nicht gestartet haben. Wählen Sie zum Löschen einer Datenmigration auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Wählen Sie dann unter Aktionen die Option Löschen. Um Ihr Migrationsprojekt zu löschen, beenden und löschen Sie alle Datenmigrationen.

Überwachung von Datenmigrationen in AWS DMS

Nachdem Sie Ihre homogene Datenmigration gestartet haben, können Sie deren Status und Fortschritt überwachen. Die Durchführung von Datenmigrationen großer Datenmengen mit Hunderten von Gigabyte dauert mehrere Stunden. Überwachen Sie den Fortschritt Ihrer Datenmigration regelmäßig, um die Zuverlässigkeit, Verfügbarkeit und hohe Leistung Ihrer Datenmigration aufrechtzuerhalten.

So prüfen Sie den Status und den Fortschritt Ihrer Datenmigration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt aus und navigieren Sie zur Registerkarte Datenmigrationen.
4. Informationen zu Ihrer Datenmigration finden Sie in der Spalte Status. Weitere Informationen zu diesen Werten in dieser Spalte finden Sie unter [Migrationsstatus](#).

5. Bei einer fortlaufenden Datenmigration wird in der Spalte Migrationsfortschritt der Prozentsatz der migrierten Daten angezeigt.

So prüfen Sie die Details Ihrer Datenmigration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt aus. Wählen Sie auf der Registerkarte Datenmigrationen Ihre Datenmigration aus.
4. Auf der Registerkarte Details können Sie den Fortschritt der Migration sehen. Insbesondere können Sie die folgenden Metriken sehen.
 - Öffentliche IP-Adresse – Die öffentliche IP-Adresse Ihrer Datenmigration. Sie benötigen diesen Wert, um ein Netzwerk zu konfigurieren. Weitere Informationen finden Sie unter [Einrichten eines Netzwerks](#).
 - Geladene Tabellen – Die Anzahl der erfolgreich geladenen Tabellen.
 - Tabellen, die geladen werden – Die Anzahl der Tabellen, die gerade geladen werden.
 - Tabellen in der Warteschlange – Die Anzahl der Tabellen, die derzeit darauf warten, geladen zu werden.
 - Fehlerhafte Tabellen – Die Anzahl der Tabellen, die nicht geladen werden konnten.
 - Verstrichene Zeit – Die Zeit, die nach dem Start Ihrer Datenmigration vergangen ist.
 - CDC-Latenz – Die durchschnittliche Zeit, die zwischen dem Zeitpunkt, an dem eine Änderung an einer Quelltablette erfolgt, und dem Zeitpunkt, an dem AWS DMS diese Änderung auf die Zieltabelle anwendet, vergeht.
 - Migration gestartet – Der Zeitpunkt, zu dem Sie diese Datenmigration gestartet haben.
 - Migration angehalten – Der Zeitpunkt, zu dem Sie diese Datenmigration angehalten haben.
5. Um die Protokolldateien für Ihre Datenmigration anzuzeigen, wählen Sie CloudWatch Protokolle anzeigen unter Einstellungen für homogene Datenmigration aus. Sie können CloudWatch Protokolle aktivieren, wenn Sie eine Datenmigration erstellen oder ändern. Weitere Informationen finden Sie unter [Erstellen einer Datenmigration](#) und [Verwaltung von Datenmigrationen](#).

Sie können Amazon- CloudWatch Alarme oder -Ereignisse verwenden, um Ihre Datenmigration genau zu verfolgen. Weitere Informationen finden Sie unter [Was sind Amazon CloudWatch, Amazon](#)

[CloudWatch Events und Amazon CloudWatch Logs?](#) im Amazon- CloudWatch Benutzerhandbuch.

Beachten Sie, dass für die Nutzung von Amazon eine Gebühr erhoben wird CloudWatch.

Für homogene Datenmigrationen AWS DMS enthält die folgenden Metriken in Amazon CloudWatch.

Kennzahl	Beschreibung
OverallCDCLatency	<p>Die Gesamtlatenz während der CDC-Phase.</p> <p>Für MySQL-Datenbanken zeigt diese Metrik die Anzahl der Sekunden, die zwischen der Änderung im Quell-Binärprotokoll und der Replikation dieser Änderung vergehen.</p> <p>Für PostgreSQL-Datenbanken zeigt diese Metrik die Anzahl der Sekunden, die zwischen <code>last_msg_receipt_time</code> und <code>last_msg_send_time</code> aus der <code>pg_stat_subscription</code> - Ansicht vergehen.</p> <p>Einheiten: Sekunden</p>
StorageConsumption	<p>Der Speicherplatz, den Ihre Datenmigration verbraucht.</p> <p>Einheiten: Byte</p>


Status homogener Datenmigrationen in AWS DMS

AWS DMS zeigt für jede Datenmigration, die Sie ausführen, den Status in der AWS DMS-Konsole an. Die folgende Liste enthält die möglichen Status:

- **Creating** – AWS DMS erstellt die Datenmigration.
- **Ready** – Die Datenmigration ist startbereit.
- **Starting** – AWS DMS erstellt die Serverless-Umgebung für Ihre Datenmigration. Dieser Vorgang kann bis zu 15 Minuten dauern.
- **Load running** – AWS DMS führt die Volllastmigration durch.
- **Load complete, replication ongoing** – AWS DMS hat den Volllast-Vorgang abgeschlossen und repliziert jetzt fortlaufende Änderungen. AWS DMS verwendet diesen Status nur für Datenmigrationen des Typs „Volllast und Change Data Capture (CDC)“.

- **Replication ongoing** – AWS DMS repliziert fortlaufende Änderungen. AWS DMS verwendet diesen Status nur für Datenmigrationen des Typs „Change Data Capture (CDC)“.
- **Reloading target** – AWS DMS startet eine Datenmigration neu und führt den angegebenen Migrationstyp durch.
- **Stopping** – AWS DMS hält die Datenmigration an. AWS DMS legt diesen Status fest, nachdem Sie im Menü Aktionen ausgewählt haben, dass die Datenmigration angehalten werden soll.
- **Stopped** – AWS DMS hat die Datenmigration angehalten.
- **Failed** – Die Datenmigration ist fehlgeschlagen. Weitere Informationen finden Sie in den Protokolldateien.

Wählen Sie zur Anzeige der Protokolldateien auf der Registerkarte Datenmigrationen Ihre Datenmigration aus. Wählen Sie als Nächstes CloudWatch Protokolle anzeigen unter Einstellungen für homogene Datenmigration aus.

 **Important**

Sie können Protokolldateien anzeigen, wenn Sie bei der Erstellung Ihrer Datenmigration das Kontrollkästchen CloudWatch Protokolle aktivieren aktivieren aktivieren.

- **Deleting** – AWS DMS löscht die Datenmigration. AWS DMS legt diesen Status fest, nachdem Sie im Menü Aktionen ausgewählt haben, dass die Datenmigration gelöscht werden soll.

Migrieren von Daten aus MySQL-Datenbanken mit homogenen Datenmigrationen in AWS DMS

Sie können [Homogene Datenbankmigrationen](#) verwenden, um eine selbstverwaltete MySQL-Datenbank zu RDS für MySQL oder Aurora MySQL zu migrieren. AWS DMS erstellt eine Serverless-Umgebung für Ihre Datenmigration. Für verschiedene Arten von Datenmigrationen verwendet AWS DMS verschiedene native MySQL-Datenbanktools.

AWS DMS verwendet bei homogenen Datenmigrationen des Typs Volllast mydumper, um Daten aus Ihrer Quelldatenbank zu lesen und sie auf der Festplatte zu speichern, die an die Serverless-Umgebung angeschlossen ist. Nachdem AWS DMS alle Ihre Quelldaten gelesen hat, verwendet es myloader in der Zieldatenbank, um Ihre Daten wiederherzustellen.

AWS DMS verwendet bei homogenen Datenmigrationen des Typs Volllast und Change Data Capture (CDC) mydumper, um Daten aus Ihrer Quelldatenbank zu lesen und sie auf der Festplatte

zu speichern, die an die Serverless-Umgebung angeschlossen ist. Nachdem AWS DMS alle Ihre Quelldaten gelesen hat, verwendet es myloader in der Zieldatenbank, um Ihre Daten wiederherzustellen. Nach Abschluss des Vollladevorgangs durch AWS DMS wird die Binärprotokoll-Replikation eingerichtet, wobei die Binlog-Position auf den Beginn des Volllast-Vorgangs gesetzt wird. Um Dateninkonsistenzen zu vermeiden, setzen Sie die Anzahl der Jobs auf 1, um den konsistenten Zustand vorhandener Daten zu erfassen. Weitere Informationen finden Sie unter [Erstellen einer Datenmigration](#).

Für homogene Datenmigrationen des Typs Change Data Capture (CDC) erfordert AWS DMS den systemeigenen CDC-Startpunkt, um die Replikation zu starten. Wenn Sie den systemeigenen CDC-Startpunkt angeben, erfasst AWS DMS die Änderungen von diesem Punkt aus. Wählen Sie alternativ in den Datenmigrationseinstellungen die Option Sofort aus, um den Startpunkt für die Replikation automatisch zu erfassen, wenn die eigentliche Datenmigration beginnt.

Note

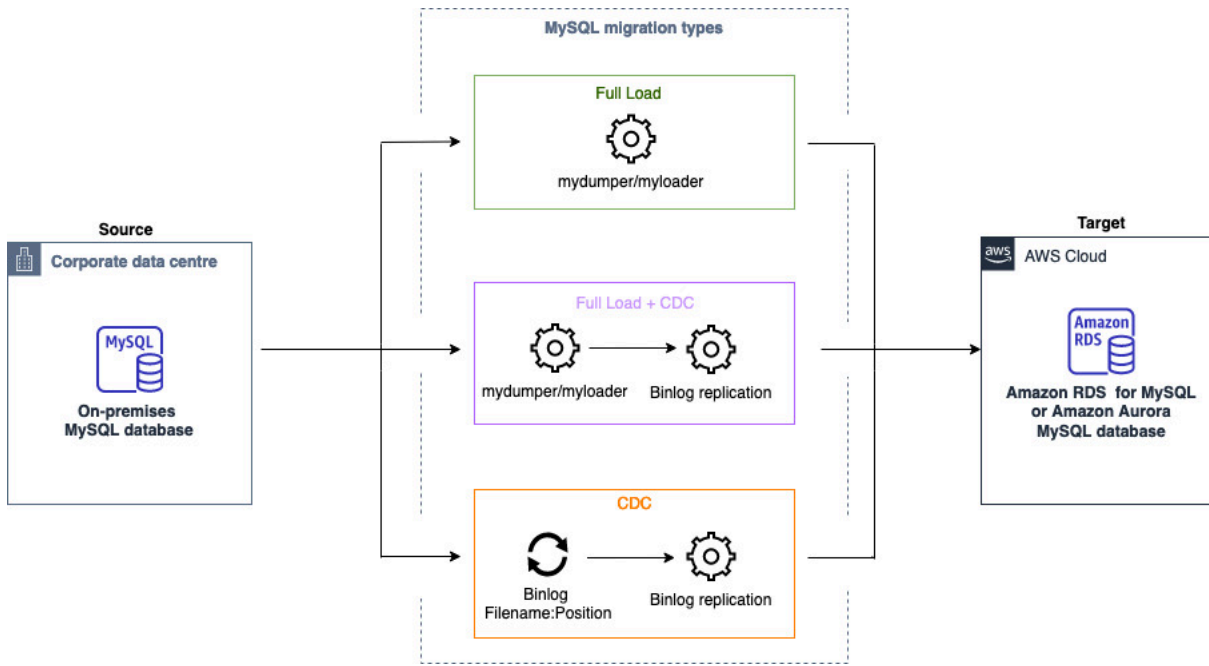
Damit eine reine CDC-Migration ordnungsgemäß funktioniert, müssen alle Quelldatenbankschemata und -objekte bereits in der Zieldatenbank vorhanden sein. Das Ziel kann jedoch Objekte enthalten, die in der Quelle nicht vorhanden sind.

Sie können das folgende Codebeispiel verwenden, um die aktuelle Protokoll-Sequenznummer (LSN) in Ihrer MySQL-Datenbank abzurufen.

```
show master status
```

Diese Abfrage gibt einen Binärprotokoll-Dateinamen und die Position zurück. Der systemeigene CDC-Startpunkt ist eine Kombination aus dem Namen der Binärprotokolldatei und der Position. Beispiel: `mysql-bin-changelog.000024:373` In diesem Beispiel ist `mysql-bin-changelog.000024` der Binärprotokoll-Dateiname und 373 die Position, an der AWS DMS mit der Erfassung der Änderungen beginnt.

Das folgende Diagramm zeigt den Prozess der Verwendung homogener Datenmigrationen in AWS DMS für die Migration einer MySQL-Datenbank zu RDS für MySQL oder Aurora MySQL.



Migrieren von Daten aus PostgreSQL-Datenbanken mit homogenen Datenmigrationen in AWS DMS


Sie können [Homogene Datenbankmigrationen](#) verwenden, um eine selbstverwaltete PostgreSQL-Datenbank zu RDS für MySQL oder Aurora MySQL zu migrieren. AWS DMS erstellt eine Serverless-Umgebung für Ihre Datenmigration. Für verschiedene Arten von Datenmigrationen verwendet AWS DMS verschiedene native PostgreSQL-Datenbanktools.

AWS DMS verwendet bei homogenen Datenmigrationen des Typs Volllast `pg_dump`, um Daten aus Ihrer Quelldatenbank zu lesen und sie auf der Festplatte zu speichern, die an die Serverless-Umgebung angeschlossen ist. Nachdem AWS DMS alle Ihre Quelldaten gelesen hat, verwendet es `pg_restore` in der Zieldatenbank, um Ihre Daten wiederherzustellen.

AWS DMS verwendet bei homogenen Datenmigrationen des Typs Volllast und Change Data Capture (CDC) `pg_dump`, um Daten aus Ihrer Quelldatenbank zu lesen und sie auf der Festplatte zu speichern, die an die Serverless-Umgebung angeschlossen ist. Nachdem AWS DMS alle Ihre Quelldaten gelesen hat, verwendet es `pg_restore` in der Zieldatenbank, um Ihre Daten wiederherzustellen. Nach Abschluss des Volllast-Vorgangs wechselt AWS DMS automatisch zu einem Publisher- und Subscriber-Modell für die logische Replikation. Bei diesem Modell abonnieren ein oder mehrere Subscriber eine oder mehrere Veröffentlichungen auf einem Publisher-Knoten.

Bei homogenen Datenmigrationen des Typs Change Data Capture (CDC) erfordert AWS DMS den systemeigenen Startpunkt, um die Replikation zu starten. Wenn Sie den systemeigenen Startpunkt

angeben, erfasst AWS DMS die Änderungen von diesem Punkt aus. Wählen Sie alternativ in den Datenmigrationseinstellungen die Option Sofort aus, um den Startpunkt für die Replikation automatisch zu erfassen, wenn die eigentliche Datenmigration beginnt.

 Note

Damit eine reine CDC-Migration ordnungsgemäß funktioniert, müssen alle Quelldatenbankschemata und -objekte bereits in der Zieldatenbank vorhanden sein. Das Ziel kann jedoch Objekte enthalten, die in der Quelle nicht vorhanden sind.

Sie können das folgende Codebeispiel verwenden, um den systemeigenen Startpunkt in Ihrer PostgreSQL-Datenbank abzurufen.

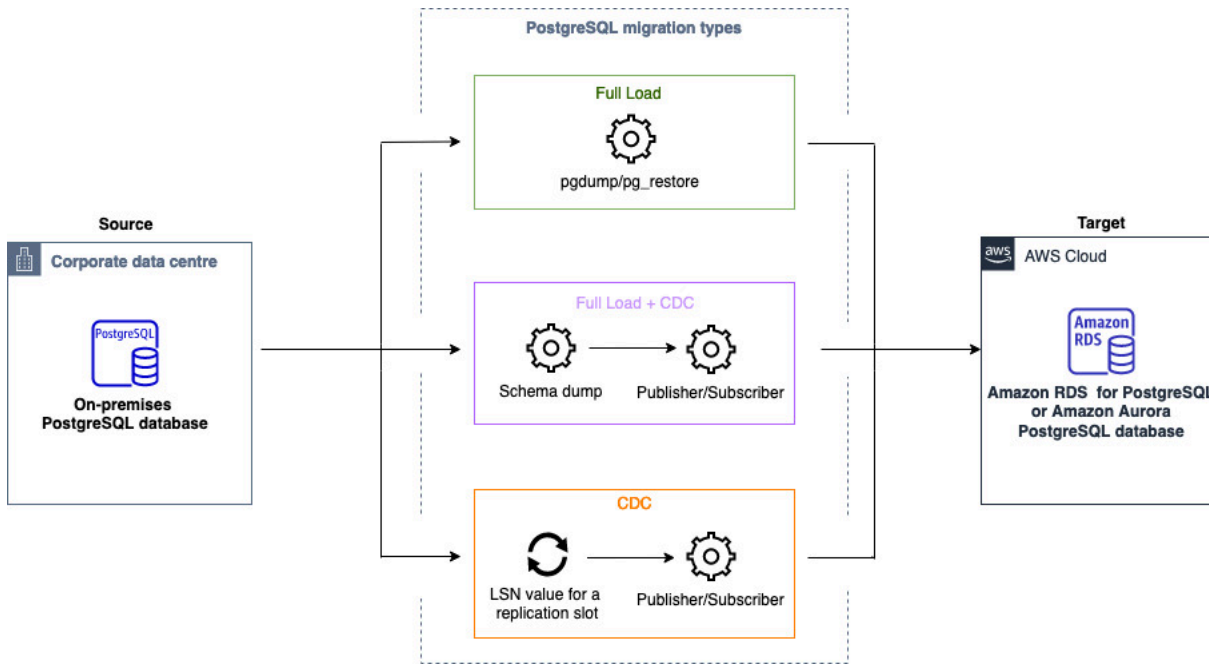
```
select confirmed_flush_lsn from pg_replication_slots where
slot_name='migrate_to_target';
```

Diese Abfrage verwendet die `pg_replication_slots`-Ansicht in Ihrer PostgreSQL-Datenbank, um den LSN-Wert (Log Sequence Number) zu erfassen.

Nachdem AWS DMS den Status Ihrer homogenen PostgreSQL-Datenmigration auf Angehalten, Fehlgeschlagen oder Gelöscht gesetzt hat, werden Publisher und Replikation nicht gelöscht. Wenn Sie die Migration nicht fortsetzen möchten, löschen Sie den Replikationsslot und den Publisher mithilfe des folgenden Befehls.

```
SELECT pg_drop_replication_slot('migration_subscriber_{ARN}');
DROP PUBLICATION publication_{ARN};
```

Das folgende Diagramm zeigt den Prozess der Verwendung homogener Datenmigrationen in AWS DMS für die Migration einer PostgreSQL-Datenbank zu RDS für MySQL oder Aurora MySQL.



Migrieren von Daten aus MongoDB-Datenbanken mit homogenen Datenmigrationen in AWS DMS

Sie können verwenden [Homogene Datenbankmigrationen](#), um eine selbstverwaltete MongoDB-Datenbank zu Amazon DocumentDB zu migrieren. AWS DMS erstellt eine Serverless-Umgebung für Ihre Datenmigration. Für verschiedene Arten von Datenmigrationen AWS DMS verwendet verschiedene native MongoDB-Datenbanktools.

Bei homogenen Datenmigrationen des Typs Volllast AWS DMS verwendet `mongodump` um Daten aus Ihrer Quelldatenbank zu lesen und auf der Festplatte zu speichern, die an die Serverless-Umgebung angeschlossen ist. Nachdem alle Ihre Quelldaten AWS DMS gelesen hat, verwendet es `mongorestore` in der Zieldatenbank, um Ihre Daten wiederherzustellen.

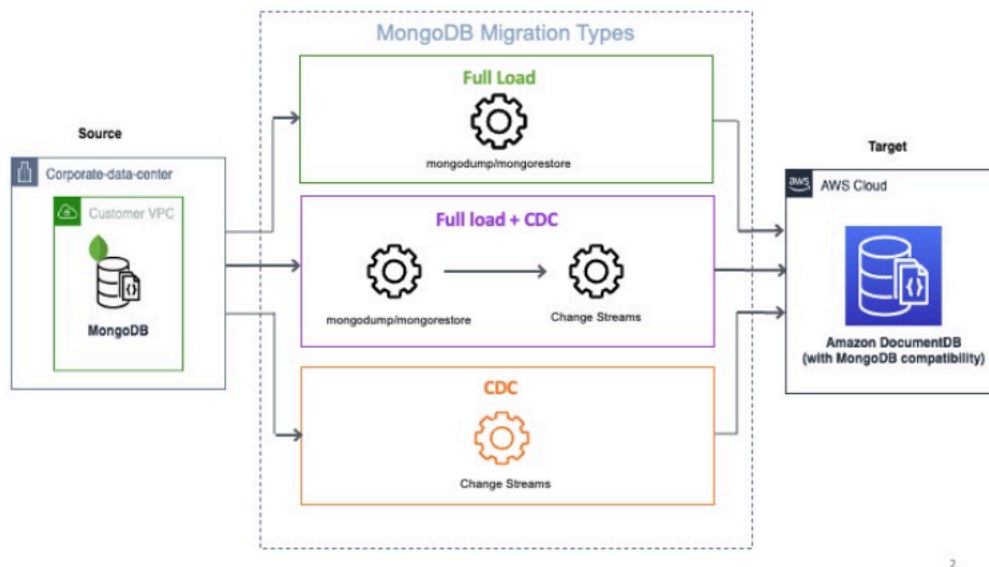
Für homogene Datenmigrationen des Typs Volllast und Erfassung von Datenänderungen (CDC) AWS DMS verwendet `mongodump` um Daten aus Ihrer Quelldatenbank zu lesen und auf der Festplatte zu speichern, die an die Serverless-Umgebung angeschlossen ist. Nachdem alle Ihre Quelldaten AWS DMS gelesen hat, verwendet es `mongorestore` in der Zieldatenbank, um Ihre Daten wiederherzustellen. Nach Abschluss des Volllast-Vorgangs wechselt AWS DMS automatisch zu einem Publisher- und Subscriber-Modell für die logische Replikation. In diesem Modell empfehlen wir, die Größe des Oplogs so zu dimensionieren, dass Änderungen mindestens 24 Stunden lang beibehalten werden.

Wählen Sie für homogene Datenmigrationen des Typs Change Data Capture (CDC) **immediately** in den Datenmigrationseinstellungen aus, um den Startpunkt für die Replikation automatisch zu erfassen, wenn die eigentliche Datenmigration beginnt.

Note

Für jede neue oder umbenannte Sammlung müssen Sie eine neue Datenmigrationsaufgabe für diese Sammlungen als homogene Datenmigrationen erstellen. Für eine MongoDB-kompatible Quelle unterstützt AWS DMS keine `-create`, `-rename` und `-drop collection` Operationen.

Das folgende Diagramm zeigt den Prozess der Verwendung homogener Datenmigrationen in , AWS DMS um eine MongoDB-Datenbank zu Amazon DocumentDB zu migrieren.



Problembehandlung bei homogenen Datenmigrationen in AWS DMS

In der folgenden Liste finden Sie Maßnahmen, die Sie ergreifen können, wenn Sie auf Probleme mit homogenen Datenmigrationen in AWS DMS stoßen.

Themen

- [Ich kann keine homogene Datenmigration in AWS DMS erstellen](#)
- [Ich kann keine homogene Datenmigration in AWS DMS starten](#)

- [Ich kann keine Verbindung zur Zieldatenbank herstellen, wenn ich eine Datenmigration in AWS DMS durchführe](#)
- [AWS DMS migriert Ansichten als Tabellen in PostgreSQL](#)

Ich kann keine homogene Datenmigration in AWS DMS erstellen

Wenn Sie eine Fehlermeldung erhalten, die besagt, dass AWS DMS keine Verbindung zu Ihren Datenanbietern herstellen kann, nachdem Sie Datenmigration erstellen ausgewählt haben, stellen Sie sicher, dass Sie die erforderliche IAM-Rolle konfiguriert haben. Weitere Informationen finden Sie unter [Erstellen einer IAM-Rolle](#).

Wenn Sie die IAM-Rolle konfiguriert haben und immer noch diese Fehlermeldung erhalten, fügen Sie diese IAM-Rolle Ihrem Hauptbenutzer in der AWS KMS-Schlüsselkonfiguration hinzu. Weitere Informationen finden Sie unter [Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben](#) im AWS Key Management Service-Entwicklerhandbuch.

Ich kann keine homogene Datenmigration in AWS DMS starten

Wenn Sie den Failed-Status erhalten, wenn Sie eine Datenmigration in Ihrem Migrationsprojekt starten, überprüfen Sie die Versionen Ihrer Quell- und Zieldatenanbieter. Führen Sie dazu die `SELECT VERSION();`-Abfrage in Ihrer MySQL- oder PostgreSQL-Datenbank aus. Achten Sie darauf, die unterstützte Datenbankversion zu verwenden.

Eine Liste der unterstützten Quelldatenbanken finden Sie unter [Quellen für homogene DMS-Datenmigrationen](#).

Eine Liste der unterstützten Zieldatenbanken finden Sie unter [Ziele für homogene DMS-Datenmigrationen](#).

Wenn Sie eine nicht unterstützte Datenbankversion verwenden, aktualisieren Sie Ihre Quell- oder Zieldatenbank und versuchen Sie es erneut.

Überprüfen Sie die Fehlermeldung für Ihre Datenmigration in der AWS DMS-Konsole. Öffnen Sie dazu Ihr Migrationsprojekt und wählen Sie Ihre Datenmigration aus. Überprüfen Sie auf der Registerkarte Details die Option Letzte Fehlermeldung unter Allgemein.

Analysieren Sie abschließend das CloudWatch-Protokoll. Öffnen Sie dazu Ihr Migrationsprojekt und wählen Sie Ihre Datenmigration aus. Wählen Sie auf der Registerkarte Details die Option CloudWatch-Protokolle anzeigen aus.

Ich kann keine Verbindung zur Zieldatenbank herstellen, wenn ich eine Datenmigration in AWS DMS durchführe

Wenn Sie die Fehlermeldung Verbindung zum Ziel nicht möglich erhalten, führen Sie die folgenden Aktionen durch.

1. Stellen Sie sicher, dass die Sicherheitsgruppe, die an Ihre Quell- und Zieldatenbanken angefügt ist, eine Regel für eingehenden und ausgehenden Datenverkehr enthält. Weitere Informationen finden Sie unter [Konfiguration der fortlaufenden Datenreplikation](#).
2. Überprüfen Sie die Zugriffssteuerungsliste (ACL) und die Routing-Tabellenregeln.
3. Auf Ihre Datenbank muss von der VPC aus zugegriffen werden können, die Sie erstellt haben. Fügen Sie öffentliche IP-Adressen zu VPC-Sicherheitsgruppen hinzu und lassen Sie Eingabeverbindungen in Ihrer Firewall zu.
4. Wählen Sie auf der Registerkarte Datenmigrationen Ihres Migrationsprojekts Ihre Datenmigration aus. Notieren Sie sich die öffentliche IP-Adresse unter Konnektivität und Sicherheit auf der Registerkarte Details. Erlauben Sie als Nächstes den Zugriff über die öffentliche IP-Adresse Ihrer Datenmigration in Ihren Quell- und Zieldatenbanken.
5. Stellen Sie für die fortlaufende Datenreplikation sicher, dass Ihre Quell- und Zieldatenbanken miteinander kommunizieren können.

Weitere Informationen finden Sie unter [Kontrollieren des Datenverkehrs zu Ressourcen mithilfe von Sicherheitsgruppen](#) im Benutzerhandbuch von Amazon Virtual Private Cloud.

AWS DMS migriert Ansichten als Tabellen in PostgreSQL

Eine homogene Datenmigration unterstützt nicht die Migration von Ansichten als Ansichten in PostgreSQL. Für PostgreSQL migriert AWS DMS Ansichten als Tabellen.

Arbeit mit Datenanbietern, Instanzprofilen und Migrationsprojekten in AWS DMS

Wenn Sie DMS-Schemakonvertierung und homogene Datenmigrationen in verwenden AWS Database Migration Service, arbeiten Sie mit Migrationsprojekten. AWS DMS -Migrationsprojekte wiederum verwenden Subnetzgruppen, Instance-Profile und Datenanbieter.

Ein Subnetz ist ein Bereich an IP-Adressen in Ihrer VPC. Eine Replikationssubnetzgruppe umfasst Subnetze aus verschiedenen Availability Zones, die Ihr Instanzprofil verwenden kann. Beachten Sie, dass eine Replikationssubnetzgruppe eine DMS-Ressource ist und sich von Subnetzgruppen unterscheidet, die Amazon VPC und Amazon RDS verwenden.

Ein Instance-Profil spezifiziert Netzwerk- und Sicherheitseinstellungen für die Serverless-Umgebung, in der Ihr Migrationsprojekt ausgeführt wird.

Ein Datenanbieter speichert einen Datenspeichertyp und die Standortinformationen zu Ihrer Datenbank. Nachdem Sie Ihrem Migrationsprojekt einen Datenanbieter hinzugefügt haben, geben Sie die Datenbankanmeldedaten von ein. AWS Secrets Manager AWS DMS verwendet diese Informationen, um eine Verbindung zu Ihrer Datenbank herzustellen.

Nachdem Sie Datenanbieter, Ihr Instanzprofil und andere AWS Ressourcen erstellt haben, können Sie ein Migrationsprojekt erstellen. Ein Migrationsprojekt beschreibt Ihr Instanzprofil, Quell- und Zieldatenanbieter sowie geheime Informationen von AWS Secrets Manager. Sie können mehrere Migrationsprojekte für verschiedene Quell- und Zieldatenanbieter erstellen.

Sie erledigen den Großteil Ihrer Arbeit im Migrationsprojekt. Bei DMS Schema Conversion verwenden Sie ein Migrationsprojekt, um die Objekte Ihres Quelldatenanbieters zu bewerten und sie in ein mit der Zieldatenbank kompatibles Format zu konvertieren. Anschließend können Sie den konvertierten Code auf Ihren Zieldatenanbieter anwenden oder ihn als SQL-Skript speichern. Bei homogenen Datenmigrationen verwenden Sie ein Migrationsprojekt, um Daten aus Ihrer Quelldatenbank in eine Zieldatenbank desselben Typs in der AWS Cloud zu migrieren.

Migrationsprojekte in AWS DMS sind nur serverlos. AWS DMS stellt automatisch die Cloud-Ressourcen für Ihre Migrationsprojekte bereit.

AWS DMS hat die maximale Anzahl von Instanzprofilen, Datenanbietern und Migrationsprojekten, die Sie für Ihre erstellen können AWS-Konto. Im folgenden Abschnitt finden Sie Informationen zu AWS DMS -Servicekontingenten [Kontingente für AWS Database Migration Service](#).

Themen

- [Eine Subnetzgruppe für ein AWS DMS Migrationsprojekt erstellen](#)
- [Instanzprofile erstellen für AWS Database Migration Service](#)
- [Datenanbieter erstellen in AWS Database Migration Service](#)
- [Migrationsprojekte erstellen in AWS Database Migration Service](#)
- [Verwaltung von Migrationsprojekten in AWS Database Migration Service](#)

Eine Subnetzgruppe für ein AWS DMS Migrationsprojekt erstellen

Bevor Sie ein Instance-Profil erstellen, konfigurieren Sie eine Subnetzgruppe für Ihr Instance-Profil.

So erstellen Sie eine Subnetzgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Subnetzgruppen und anschließend Subnetzgruppe erstellen aus.
3. Geben Sie für Name einen eindeutigen Namen Ihrer Subnetzgruppe ein.
4. Geben Sie unter Beschreibung eine kurze Beschreibung Ihrer Subnetzgruppe ein.
5. Wählen Sie für VPC eine VPC aus, die über mindestens ein Subnetz in mindestens zwei Availability Zones verfügt.
6. Wählen Sie unter Subnetze hinzufügen die Subnetze aus, die in die Subnetzgruppe aufgenommen werden sollen. Sie müssen Subnetze in mindestens zwei Availability Zones auswählen.

Um eine Verbindung zu Amazon-RDS-Datenbanken herzustellen, fügen Sie Ihrer Subnetzgruppe öffentliche Subnetze hinzu. Um eine Verbindung zu On-Premises-Datenbanken herzustellen, fügen Sie Ihrer Subnetzgruppe private Subnetze hinzu.

7. Wählen Sie Create subnet group (Subnetz-Grupp erstellen) aus.

Instanzprofile erstellen für AWS Database Migration Service

Sie können mehrere Instanzprofile in der AWS DMS Konsole erstellen. Stellen Sie sicher, dass Sie ein Instance-Profil auswählen, das für jedes Migrationsprojekt verwendet werden soll, das Sie in AWS DMS erstellen.

So erstellen Sie ein Instance-Profil

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Instance-Profile aus.
3. Wählen Sie Instance-Profil erstellen aus.
4. Geben Sie auf der Seite Instance-Profil erstellen einen beschreibenden Wert für den Namen Ihres Instance-Profiles ein.
5. Wählen Sie als Netzwerktyp den Dual-Stack-Modus, um ein Instance-Profil zu erstellen, das IPv4- und IPv6-Adressierung unterstützt. Behalten Sie die Standardoption bei, um ein Instance-Profil zu erstellen, das nur IPv4-Adressierung unterstützt.
6. Wählen Sie als Nächstes Virtual Private Cloud (VPC), um Ihre Instance des ausgewählten Netzwerktyps auszuführen. Wählen Sie dann eine Subnetzgruppe und VPC-Sicherheitsgruppen für Ihr Instance-Profil aus.

Um eine Verbindung zu Amazon-RDS-Datenbanken herzustellen, verwenden Sie eine Subnetzgruppe, die öffentliche Subnetze enthält. Um eine Verbindung zu On-Premises-Datenbanken herzustellen, verwenden Sie eine Subnetzgruppe, die private Subnetze enthält. Stellen Sie sicher, dass Sie Ihr Netzwerk so konfiguriert haben, dass es über die öffentliche IP-Adresse des NAT-Gateways auf Ihre lokale Quelldatenbank zugreifen AWS DMS kann. Weitere Informationen finden Sie unter [Erstellen einer VPC basierend auf Amazon VPC](#).

7. (Optional) Wenn Sie ein Migrationsprojekt für DMS Schema Conversion erstellen, wählen Sie anschließend für Einstellungen zur Schemakonvertierung – optional einen Amazon-S3-Bucket zum Speichern von Informationen aus Ihrem Migrationsprojekt aus. Wählen Sie dann die AWS Identity and Access Management (IAM-) Rolle aus, die den Zugriff auf diesen Amazon S3 S3-Bucket ermöglicht. Weitere Informationen finden Sie unter [Erstellen eines Amazon-S3-Buckets](#).
8. Wählen Sie Instance-Profil erstellen aus.

Nachdem Sie Ihr Instance-Profil erstellt haben, können Sie es ändern oder löschen.

So ändern Sie ein Instance-Profil

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Instance profiles (Instance-Profile) aus. Die Seite Instance-Profile wird geöffnet.
3. Wählen Sie Ihr Instance-Profil und dann Ändern.

4. Aktualisieren Sie den Namen Ihres Instance-Profiles und bearbeiten Sie die VPC- oder Amazon-S3-Bucket-Einstellungen.
5. Wählen Sie Änderungen speichern aus.

So löschen Sie ein Instance-Profil

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Instance profiles (Instance-Profile) aus. Die Seite Instance-Profile wird geöffnet.
3. Wählen Sie Ihr Instance-Profil und dann Löschen.
4. Wählen Sie zur Bestätigung Ihrer Auswahl Löschen aus.

Datenanbieter erstellen in AWS Database Migration Service

Sie können Datenanbieter erstellen und sie in AWS DMS Migrationsprojekten verwenden. Ihr Datenanbieter kann eine selbstverwaltete Engine sein, die On-Premises oder auf einer Amazon-EC2-Instance ausgeführt wird. Ihr Datenanbieter kann auch eine vollständig verwaltete Engine sein, z. B. Amazon Relational Database Service (Amazon RDS) oder Amazon Aurora.

Für jede Datenbank können Sie einen einzelnen Datenanbieter erstellen. Sie können einen einzelnen Datenanbieter in mehreren Migrationsprojekten verwenden.

Bevor Sie ein Migrationsprojekt erstellen, vergewissern Sie sich, dass Sie mindestens zwei Datenanbieter erstellt haben. Einer Ihrer Datenanbieter muss sich in einem AWS-Service befinden. Sie können AWS DMS nicht verwenden, um Ihre Schemata zu konvertieren oder Ihre Daten in eine On-Premises-Datenbank zu migrieren.

Das folgende Verfahren zeigt Ihnen, wie Sie Datenprovider im AWS DMS Konsolenassistenten erstellen.

So erstellen Sie einen Datenanbieter

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie dann die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Datenanbieter aus. Die Seite Datenanbieter wird geöffnet.
3. Wählen Sie Datenanbieter erstellen aus. In der folgenden Tabelle sind die Einstellungen beschrieben.

Option	Aktion
Konfiguration	Wählen Sie aus, ob Sie die Informationen über Ihren Datenanbieter manuell eingeben oder die Amazon-RDS-DB-Instance verwenden möchten.
Name	Geben Sie einen Namen für Ihren Datenanbieter ein. Stellen Sie sicher, dass Sie einen eindeutigen Namen für Ihren Datenanbieter verwenden, damit Sie ihn leicht identifizieren können.
Engine type (Engine-Typ)	Wählen Sie den Typ der Datenbank-Engine für Ihren Datenanbieter aus.
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Datenbankservers ein. Der Servername für einen für eine homogene Replikation verwendeten Datenanbieter muss mit einem alphanumerischen Zeichen beginnen und darf nur alphanumerische Zeichen, Bindestriche (-), Punkte (.) oder Unterstriche (_) enthalten.
Port	Geben Sie den Port ein, der für die Verbindung zu Ihrem Datenbankserver verwendet wird.
Service-ID (SID) oder Servicename	Geben Sie die Oracle-System-ID (SID) ein. Um die Oracle-SID abzurufen, senden Sie die folgende Abfrage an Ihre Oracle-Datenbank: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre>
Datenbankname	Geben Sie den Namen der Datenbank für diesen Datenanbieter ein. Der Datenbankname für einen Datenprovider, der für eine homogene Replikation verwendet wird, kann bis zu 63 Zeichen lang sein und darf keine Leerzeichen enthalten.

Option	Aktion
Secure Socket Layer (SSL)-Modus	Wählen Sie einen SSL-Modus, wenn Sie die Verbindungsverschlüsselung für diesen Datenanbieter aktivieren möchten. Je nach ausgewähltem Modus müssen Sie möglicherweise Zertifikat- und Serverzertifikatinformationen angeben. Weitere Informationen finden Sie unter SSL verwenden mit AWS Database Migration Service .
Authentifizierungsmodus	Für eine MongoDB-Quelle der Authentifizierungsmodus, der zur Authentifizierung der Endpunktverbindung AWS DMS verwendet wird.
Authentifizierungsquelle	Für eine MongoDB-Quelle der Name der MongoDB-Datenbank, die zur Überprüfung Ihrer Anmeldeinformationen für die Authentifizierung verwendet werden soll.
Authentifizierungsmechanismus	Für eine MongoDB-Quelle die Authentifizierungsmethode, die MongoDB verwendet, um das Passwort zu verschlüsseln.

4. Wählen Sie Datenanbieter erstellen aus.

Nachdem Sie einen Datenanbieter erstellt haben, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Datenbankverbindung in AWS Secrets Manager hinzufügen.

Migrationsprojekte erstellen in AWS Database Migration Service

Bevor Sie ein Migrationsprojekt in erstellen AWS DMS, stellen Sie sicher, dass Sie die folgenden Ressourcen erstellen:

- Datenanbieter, die Ihre Quell- und Zieldatenbanken beschreiben
- Geheimnisse mit Datenbankmeldedaten, gespeichert in AWS Secrets Manager
- Die AWS Identity and Access Management (IAM) -Rolle, die den Zugriff auf Secrets Manager ermöglicht
- Ein Instance-Profil, das Netzwerk- und Sicherheitseinstellungen beinhaltet

So erstellen Sie ein Migrationsprojekt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Migrationsprojekt erstellen aus. In der folgenden Tabelle sind die Einstellungen beschrieben.

Option	Aktion
Name	Geben Sie einen Namen für Ihr Migrationsprojekt ein. Stellen Sie sicher, dass Sie einen eindeutigen Namen für Ihr Migrationsprojekt verwenden, damit Sie es leicht identifizieren können.
Instance-Profil	Wählen Sie Ihr Instance-Profil aus, das Sie für Ihr Migration sprojekt verwenden möchten.
Quelle	Wählen Sie Durchsuchen und wählen Sie dann Ihren Quelldatenanbieter aus.
Secret-ID	Wählen Sie den Amazon-Ressourcennamen (ARN) Ihres Secrets in Secrets Manager, in dem Ihre Quelldatenbank-Anmeldeinformationen gespeichert sind.
IAM role (IAM-Rolle)	Wählen Sie eine IAM-Rolle, um Zugriff auf Ihre Quelldate nbank-Anmeldeinformationen in Secrets Manager zu gewähren.
Ziel	Wählen Sie Durchsuchen und wählen Sie dann Ihren Quelldatenanbieter aus.
Secret-ID	Wählen Sie den ARN Ihres Secrets in Secrets Manager, in dem Ihre Quelldatenbank-Anmeldeinformationen gespeichert sind.

Option	Aktion
IAM role (IAM-Rolle)	Wählen Sie eine IAM-Rolle, um Zugriff auf Ihre Zieldatenbank-Anmeldeinformationen in Secrets Manager zu gewähren.
Transformationsregeln	(Optional) Wenn Sie ein Migrationsprojekt für DMS Schema Conversion erstellen, wählen Sie Transformationsregel hinzufügen, um Transformationsregeln einzurichten. Transformationsregeln ermöglichen Ihnen, die Objektnamen gemäß der von Ihnen angegebenen Regel zu ändern. Weitere Informationen finden Sie unter Einrichten von Transformationsregeln .

4. Wählen Sie Migrationsprojekt erstellen aus.

Nachdem Sie Ihr Migrationsprojekt AWS DMS erstellt haben, können Sie dieses Projekt für die DMS-Schemakonvertierung oder homogene Datenmigrationen verwenden. Um mit der Arbeit an Ihrem Migrationsprojekt zu beginnen, wählen Sie auf der Seite Migrationsprojekte Ihr Projekt aus der Liste aus.

Verwaltung von Migrationsprojekten in AWS Database Migration Service

Nachdem Sie Ihr Migrationsprojekt erstellt haben, können Sie es ändern oder löschen. Um beispielsweise den Quell- oder Zieldatenanbieter ändern zu können, ändern Sie Ihr Migrationsprojekt.

Sie können Ihr Migrationsprojekt erst ändern oder löschen, nachdem Sie die Schemakonvertierungs- oder Datenmigrationsvorgänge abgeschlossen haben. Wählen Sie dazu Ihr Migrationsprojekt aus der Liste aus und wählen Sie Schemakonvertierung oder Datenmigrationen aus. Wählen Sie anschließend Schemakonvertierung schließen für DMS Schema Conversion und bestätigen Sie Ihre Auswahl. Wählen Sie für homogene Datenmigrationen Ihre Datenmigration und dann im Menü Aktionen die Option Stopp aus. Nachdem Sie Ihr Migrationsprojekt bearbeitet haben, können Sie die Schemakonvertierung starten oder Ihre Datenmigration erneut starten.

So ändern Sie ein Migrationsprojekt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt und dann Ändern aus.
4. Aktualisieren Sie den Namen Ihres Projekts, bearbeiten Sie das Instance-Profil oder ändern Sie die Quell- und Zieldatenanbieter. Optional können Sie Migrationsregeln hinzufügen oder bearbeiten, die die Objektnamen während der Konvertierung ändern.
5. Wählen Sie Änderungen speichern aus.

So löschen Sie ein Migrationsprojekt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Migrationsprojekte aus. Die Seite Migrationsprojekte wird geöffnet.
3. Wählen Sie Ihr Migrationsprojekt und dann Löschen aus.
4. Wählen Sie zur Bestätigung Ihrer Auswahl Löschen aus.

Bewährte Methoden für AWS Database Migration Service

Um AWS Database Migration Service (AWS DMS) möglichst effektiv zu nutzen, schauen Sie sich die Empfehlungen in diesem Abschnitt zu den effizientesten Datenmigrationsmethoden an.

Themen

- [Planen der Migration für AWS Database Migration Service](#)
- [Konvertieren des Schemas](#)
- [Überprüfung der öffentlichen Dokumentation von AWS DMS](#)
- [Ausführen eines Machbarkeitsnachweises](#)
- [Verbessern der Leistung einer AWS DMS-Migration](#)
- [Verwenden Ihres eigenen Vor-Ort-Nameservers](#)
- [Migrieren von Binary Large Objects \(LOBs\)](#)
- [Verbessern der Leistung beim Migrieren umfangreicher Tabellen mit Zeilenfilterung](#)
- [Fortlaufende Replikation](#)
- [Reduzieren der Last Ihrer Quelldatenbank](#)
- [Reduzierung von Engpässen in Ihrer Zieldatenbank](#)
- [Verwendung der Datenvalidierung während der Migration](#)
- [Überwachen Ihrer AWS DMS-Aufgaben mithilfe von Metriken](#)
- [Ereignisse und Benachrichtigungen](#)
- [Verwenden des Aufgabenprotokolls für die Behebung von Migrationsproblemen](#)
- [Problembehandlung bei Replikationsaufgaben mit Time Travel](#)
- [Ändern des Benutzers und des Schemas für ein Oracle-Ziel](#)
- [Ändern von Tabellen- und Index-Tabellenräumen für ein Oracle-Ziel](#)
- [Aktualisieren der Engine-Version einer Replikations-Instance](#)
- [Grundlegendes zu Ihren Migrationskosten](#)

Planen der Migration für AWS Database Migration Service

Beim Planen einer Datenbankmigration mit AWS Database Migration Service sollten Sie Folgendes beachten:

- Sie müssen ein Netzwerk konfigurieren, um Ihre Quell- und Zieldatenbanken mit einer AWS DMS-Replikations-Instance zu verbinden. Dies kann so einfach sein wie die Verbindung zweier AWS-Ressourcen in derselben Virtual Private Cloud (VPC), in der sich Ihre Replikations-Instance befindet. Dies kann bis hin zu komplexeren Konfigurationen reichen, z. B. der Verbindung einer On-Premises-Datenbank mit einer Amazon RDS-DB-Instance über ein Virtual Private Network (VPN). Weitere Informationen finden Sie unter [Netzwerkkonfigurationen für die Datenbankmigration](#).
- Quell- und Ziel-Endpunkte – Sie müssen wissen, welche Informationen und Tabellen in der Quelldatenbank zur Zieldatenbank migriert werden müssen. AWS DMS unterstützt eine grundlegende Schemamigration, einschließlich der Erstellung von Tabellen und primären Schlüsseln. Allerdings erstellt AWS DMS nicht automatisch sekundäre Indizes, Fremdschlüssel, Benutzerkonten und dergleichen in der Zieldatenbank. Je nach Quell- und Ziel-Datenbank-Engine müssen Sie zusätzliche Protokollierung einrichten oder andere Einstellungen für eine Quell- oder Zieldatenbank ändern müssen. Weitere Informationen finden Sie unter [Quellen für die Datenmigration](#) und [Ziele für die Datenmigration](#).
- Schema/Code-Migration – AWS DMS führt keine Schema- oder Code-Konvertierung durch. Sie können Ihr Schema mithilfe von Tools wie Oracle SQL Developer, MySQL Workbench oder pgAdmin III konvertieren. Wenn Sie ein vorhandenes Schema in eine andere Datenbank-Engine konvertieren möchten, können Sie AWS Schema Conversion Tool (AWS SCT) verwenden. Dieses Tool kann ein Zielschema erstellen, aber auch ein ganzes Schema generieren und erstellen, wie Tabellen, Indizes, Ansichten und so weiter. Sie können das Tool auch verwenden, um PL/SQL oder TSQL in PostgreSQL und andere Formate umzuwandeln. Weitere Informationen zu AWS SCT finden Sie im [AWS SCT-Benutzerhandbuch](#).
- Nicht unterstützte Datentypen – Einige Quelldatentypen müssen in die äquivalenten Datentypen für die Zieldatenbank konvertiert werden. Weitere Informationen über unterstützte Datentypen finden Sie im Quell- oder Zielabschnitt für Ihren Datenspeicher.
- Ergebnisse des Diagnoseunterstützungsskripts – Wenn Sie Ihre Migration planen, empfehlen wir Ihnen, Diagnoseunterstützungsskripts auszuführen. Anhand der Ergebnisse dieser Skripts finden Sie erweiterte Informationen zu potenziellen Migrationsfehlern.

Wenn ein Unterstützungsskript für Ihre Datenbank verfügbar ist, laden Sie es über den Link im entsprechenden Skriptthema im folgenden Abschnitt herunter. Nachdem Sie das Skript überprüft haben, können Sie es gemäß dem im Skriptthema beschriebenen Verfahren in Ihrer lokalen Umgebung ausführen. Wenn die Skriptausführung abgeschlossen ist, können Sie die Ergebnisse überprüfen. Wir empfehlen, diese Skripts als ersten Schritt bei der Problembehandlung auszuführen. Die Ergebnisse können bei der Arbeit mit einem AWS Support-Team nützlich sein. Weitere Informationen finden Sie unter [Arbeiten mit Diagnoseunterstützungsskripts in AWS DMS](#).

- **Bewertungen vor der Migration** – Bei einer Bewertung vor der Migration werden bestimmte Komponenten einer Datenbankmigrationsaufgabe bewertet, um Probleme zu identifizieren, die verhindern könnten, dass eine Migrationsaufgabe wie erwartet ausgeführt wird. Mithilfe dieser Bewertung können Sie potenzielle Probleme identifizieren, bevor Sie eine neue oder geänderte Aufgabe ausführen. Weitere Informationen zur Arbeit mit Bewertungen vor der Migration finden Sie unter [Aktivieren und Verwenden von Vormigrationsbewertungen für eine Aufgabe](#).

Konvertieren des Schemas

AWS DMS führt keine Schema- oder Codekonvertierung durch. Wenn Sie ein vorhandenes Schema zu einer anderen Datenbank-Engine konvertieren möchten, können Sie AWS SCT verwenden. AWS SCT konvertiert Ihre Quellobjekte, Tabellen, Indizes, Ansichten, Auslöser und andere Systemobjekte in das Zielformat der Datendefinitionssprache (DDL). Sie können mithilfe von AWS SCT auch einen Großteil des Anwendungscodes, wie PL/SQL oder TSQL, in die äquivalente Zielsprache umwandeln.

Sie erhalten AWS SCT als kostenlosen Download von AWS. Weitere Informationen zu AWS SCT finden Sie im [AWS SCT-Benutzerhandbuch](#).

Wenn sich Ihre Quell- und Zielendpunkte in derselben Datenbank-Engine befinden, können Sie Tools wie Oracle SQL Developer, MySQL Workbench oder PgAdmin4 verwenden, um Ihr Schema zu verschieben.

Überprüfung der öffentlichen Dokumentation von AWS DMS

Wir empfehlen Ihnen dringend, vor der ersten Migration die öffentlichen AWS DMS-Dokumentationsseiten für Ihre Quell- und Zielendpunkte durchzugehen. Diese Dokumentation kann Ihnen helfen, die Voraussetzungen für die Migration zu ermitteln und die aktuellen Einschränkungen zu verstehen, bevor Sie beginnen. Weitere Informationen finden Sie unter [Arbeiten mit AWS-DMS-Endpunkten](#).

Während der Migration kann Ihnen die öffentliche Dokumentation bei der Behebung von Problemen mit AWS DMS helfen. Die Seiten zur Fehlerbehebung in der Dokumentation können Ihnen helfen, häufig auftretende Probleme bei der Verwendung von AWS DMS und ausgewählten Endpunktdatenbanken zu beheben. Weitere Informationen finden Sie unter [Fehlerbehebung für Migrationsaufgaben in AWS Database Migration Service](#).

Ausführen eines Machbarkeitsnachweises

Um Probleme mit Ihrer Umgebung in den frühen Phasen Ihrer Datenbankmigration besser erkennen zu können, empfehlen wir Ihnen, eine kleine Testmigration durchzuführen. Auf diese Weise können Sie auch einen realistischeren Zeitplan für die Migration festlegen. Darüber hinaus müssen Sie möglicherweise eine umfassende Testmigration durchführen, um zu messen, ob AWS DMS den Durchsatz Ihrer Datenbank über Ihr Netzwerk bewältigen kann. Während dieser Zeit empfehlen wir, Ihre anfängliche Volllast und die laufende Replikation zu vergleichen und zu optimieren. Auf diese Weise können Sie Ihre Netzwerklatenz besser verstehen und die Gesamtleistung einschätzen.

An dieser Stelle haben Sie auch die Möglichkeit, Ihr Datenprofil und die Größe Ihrer Datenbank zu verstehen, einschließlich der folgenden Informationen:

- Wie viele Tabellen groß, mittel und klein sind.
- Wie AWS DMS mit Datentyp- und Zeichensatzkonvertierungen umgeht.
- Wie viele Tabellen Large Object (LOB)-Spalten haben.
- Wie lange es dauert, eine Testmigration durchzuführen.

Verbessern der Leistung einer AWS DMS-Migration

Die Leistung Ihrer AWS DMS-Migration wird von verschiedenen Faktoren beeinflusst:

- Ressourcenverfügbarkeit an der Quelle.
- Verfügbarer Netzwerkdurchsatz.
- Ressourcenkapazität des Replikationsservers.
- Fähigkeit des Ziels, Änderungen aufzunehmen.
- Art und Verteilung der Quelldaten.
- Anzahl der zu migrierenden Objekte.

Es ist möglich, durch Beachtung einiger oder aller der nachstehend erwähnten bewährten Methoden die Leistung zu verbessern. Ob Sie eine dieser Methoden anwenden können, hängt von Ihrem speziellen Anwendungsfall ab. Im Folgenden finden Sie einige Einschränkungen:

Bereitstellung eines geeigneten Replikationsservers

AWS DMS ist ein verwalteter Service, der auf einer Amazon-EC2-Instance ausgeführt wird. Dieser Service stellt eine Verbindung zur Quelldatenbank her, liest die Quelldaten, formatiert die Daten für die Verwendung durch die Zieldatenbank und lädt die Daten dann in die Zieldatenbank.

Der Großteil dieser Verarbeitung erfolgt im Arbeitsspeicher. Allerdings werden umfangreiche Transaktionen ggf. auf Festplatte gepuffert. Zwischengespeicherte Transaktionen und Protokolldateien werden ebenfalls auf Festplatte geschrieben. In den folgenden Abschnitten finden Sie Informationen dazu, was Sie bei der Auswahl Ihres Replikationsservers beachten sollten.

CPU

AWS DMS ist für heterogene Migrationen konzipiert, unterstützt aber auch homogene Migrationen. Um eine homogene Migration durchzuführen, konvertieren Sie zunächst jeden Quelldatentyp in den entsprechenden AWS DMS-Datentyp. Konvertieren Sie dann jeden AWS DMS-Datentyp in den Zieldatentyp. Verweise auf diese Konvertierungen für jede Datenbank-Engine finden Sie im AWS DMS-Benutzerhandbuch.

Damit AWS DMS diese Konvertierungen optimal durchführen kann, muss die CPU zum Zeitpunkt der Konvertierungen verfügbar sein. Eine Überlastung der CPU und ein Mangel an CPU-Ressourcen können zu langsamen Migrationen führen, was auch weitere Nebenwirkungen haben kann.

Replikations-Instance-Klasse

Einige der kleineren Instance-Klassen reichen für das Testen des Service oder für kleine Migrationen aus. Wenn Ihre Migration eine große Anzahl von Tabellen beinhaltet, oder Sie mehrere gleichzeitige Replikationsaufgaben ausführen möchten, sollten Sie in Betracht ziehen, eine der größeren Instances zu verwenden. Eine größere Instance kann sinnvoll sein, da der Service eine beträchtliche Menge an Arbeitsspeicher und CPU verbraucht.

T2-Instances wurden entwickelt, um eine moderate Basisleistung und eine signifikant höhere Leistung bereitzustellen, je nach den Anforderungen Ihres Workloads. Sie eignen sich für Workloads, die die volle CPU-Leistung selten oder uneinheitlich verwenden, jedoch gelegentlich Spitzenlasten verarbeiten müssen. T2-Instances eignen sich gut für Allzweck-Workloads, z. B. Webserver, Entwicklungsumgebungen und kleine Datenbanken. Wenn Sie Probleme mit einer langsamen Migration beheben und einen T2-Instance-Typ verwenden, überprüfen Sie die Host-Metrik „CPU-Auslastung“. Diese kann Ihnen zeigen, ob Sie die Baseline für diesen Instance-Typ übersteigen.

Die C4-Instance-Klassen wurden für höchste Prozessorleistung für rechenintensive Workloads konzipiert. Sie erreichen eine erheblich höhere Pakete-pro-Sekunde (PPS)-Leistung, weniger Netzwerkverstopfung und eine geringere Netzwerklatenz. AWS DMS kann CPU-intensiv sein, insbesondere bei heterogenen Migrationen und Replikationen wie etwa der Migration von Oracle zu PostgreSQL. C4-Instances können eine gute Wahl für diese Situationen sein.

Die R4-Instance-Klassen sind speicheroptimiert für speicherintensive Workloads. Laufende Migrationen oder Replikationen von Transaktionssystemen mit hohem Durchsatz unter Verwendung von AWS DMS können zeitweise große Mengen an CPU und Speicher verbrauchen. R4-Instances verfügen über mehr Arbeitsspeicher pro vCPU.

AWS DMS-Unterstützung für die Instance-Klassen R5 und C5

Die R5-Instance-Klassen sind speicheroptimierte Instances, die darauf ausgelegt sind, eine schnelle Leistung bei Workloads zu erreichen, die große Datensätze im Arbeitsspeicher verarbeiten. Laufende Migrationen oder Replikationen von Transaktionssystemen mit hohem Durchsatz unter Verwendung von AWS DMS können zeitweise große Mengen an CPU und Speicher verbrauchen. R5-Instances bieten 5 Prozent mehr Speicher pro vCPU als R4 und die größte Größe bietet 768 GiB Arbeitsspeicher. Darüber hinaus bieten R5-Instances eine Verbesserung des Preises pro GiB um 10 Prozent und eine um ~ 20 % höhere CPU-Leistung als R4.

Die C5-Instance-Klassen sind für rechenintensive Workloads optimiert und bieten eine kostengünstige hohe Leistung bei einem niedrigen Preis-Leistungs-Verhältnis. Sie erreichen eine deutlich höhere Netzwerkleistung. Elastic Network Adapter (ENA) bietet C5-Instances mit bis zu 25 Gbit/s Netzwerkbandbreite und bis zu 14 Gbit/s dedizierter Bandbreite für Amazon EBS. AWS DMS kann CPU-intensiv sein, insbesondere bei heterogenen Migrationen und Replikationen wie der Migration von Oracle zu PostgreSQL. C5-Instances können für solche Situationen eine gute Wahl sein.

Speicher

Abhängig von der Instance-Klasse umfasst Ihre Replikations-Instance entweder 50 GB oder 100 GB Datenspeicher. Dieser Speicher wird für Protokolldateien und alle zwischengespeicherten Änderungen verwendet, die während des Ladevorgangs erfasst werden. Wenn Ihr Quellsystem ausgelastet ist oder umfangreiche Transaktionen verarbeitet, müssen Sie möglicherweise Ihren Speicherplatz erweitern. Wenn Sie mehrere Aufgaben auf dem Replikationsserver ausführen, benötigen Sie möglicherweise auch eine Speichererweiterung. Normalerweise reicht die Standardmenge aber aus.

Alle Speicher-Volumes in AWS DMS sind GP2- oder Allzweck-Solid-State-Laufwerke (SSDs). GP2-Volumes verfügen über eine Basisleistung von drei E/A-Vorgängen pro Sekunde (IOPS) und können auf Kreditbasis bis zu 3 000 IOPS gesteigert werden. Als Faustregel gilt: Überprüfen Sie die `ReadIOPS`- und `WriteIOPS`-Metriken für die Replikations-Instance. Stellen Sie sicher, dass die Summe dieser Werte die Basisleistung für dieses Volume nicht überschreitet.

Multi-AZ

Wenn Sie sich für eine Multi-AZ-Instance entscheiden, können Sie damit Ihre Migration vor Speicherausfällen schützen. Die meisten Migrationen sind vorübergehend und nicht für einen langen Zeitraum vorgesehen. Wenn Sie AWS DMS für fortlaufende Replikationszwecke verwenden, kann die Wahl einer Multi-AZ-Instance Ihre Verfügbarkeit verbessern, falls ein Speicherproblem auftreten sollte.

Wenn Sie eine einzelne AZ- oder Multi-AZ-Replikations-Instance während eines FULL LOAD-Vorgangs verwenden und ein Failover oder ein Host-Austausch stattfindet, ist zu erwarten, dass die Volllast-Aufgabe fehlschlägt. Für die verbleibenden Tabellen, die nicht abgeschlossen wurden oder sich in einem Fehlerstatus befinden, können Sie die Aufgabe ab dem Zeitpunkt des Fehlers neu starten.

Paralleles Laden mehrerer Tabellen

Standardmäßig lädt AWS DMS immer acht Tabellen gleichzeitig. Möglicherweise lässt sich eine Leistungsverbesserung erzielen, wenn Sie diese Menge etwas erhöhen, sofern Sie einen sehr großen Replikationsserver verwenden wie einen `dms.c4.xlarge` oder eine größere Instance. Es kann jedoch auch vorkommen, dass die Leistung durch Erhöhen der parallelen Ausführung an einem bestimmten Punkt abfällt. Wenn Ihr Replikationsserver relativ klein ist, z. B. ein `dms.t2.medium`, sollten Sie die Anzahl der parallel geladenen Tabellen reduzieren.

Um diese Zahl in der AWS Management Console zu ändern, öffnen Sie die Konsole, wählen Sie Aufgaben, wählen Sie aus, ob Sie eine Aufgabe erstellen oder ändern möchten, und wählen Sie dann Erweiterte Einstellungen. Ändern Sie unter Tuning Settings (Optimierung der Einstellungen) die Option Maximum number of tables to load in parallel (Maximale Anzahl parallel zu ladender Tabellen).

Um diese Zahl mit der AWS CLI zu ändern, ändern Sie den `MaxFullLoadSubTasks`-Parameter unter `TaskSettings`.

Verwendung paralleler Volllast

Sie können eine parallele Ladung aus Oracle-, Microsoft SQL Server-, MySQL-, Sybase- und IBM-Db2-LUW-Quellen verwenden, die auf Partitionen und Unterpartitionen basieren. Dadurch kann

die Gesamtdauer des Vollladevorgangs verbessert werden. Darüber hinaus können Sie bei der Ausführung einer AWS DMS-Migrationsaufgabe die Migration großer oder partitionierter Tabellen beschleunigen. Teilen Sie dazu die Tabelle in Segmente auf und laden Sie die Segmente parallel in dieselbe Migrationsaufgabe.

Wenn Sie paralleles Laden verwenden möchten, können Sie mit der `parallel-load`-Option eine Tabellenzuweisungsregel des Typs `table-settings` angeben. In der `table-settings`-Regel geben Sie die Auswahlkriterien für die Tabelle bzw. Tabellen an, die Sie parallel laden möchten. Um die Auswahlkriterien anzugeben, legen Sie das Element `type` für `parallel-load` auf eine der folgenden Optionen fest:

- `partitions-auto`
- `subpartitions-auto`
- `partitions-list`
- `ranges`
- `none`

Weitere Informationen zu diesen Einstellungen finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Arbeiten mit Indizes, Auslösern und Einschränkungen bezüglich der referenziellen Integrität

Indizes, Auslöser und Einschränkungen bezüglich der referenziellen Integrität können sich auf Ihre Migrationsleistung auswirken kann dazu führen, dass Ihre Migration fehlschlägt. Wie sich diese auf die Migration auswirken, hängt davon ab, ob Ihre Replikationsaufgabe eine Volllastaufgabe oder eine laufende Replikationsaufgabe (Change Data Capture, CDC) ist.

Für eine Volllastaufgabe empfehlen wir, Primärschlüsselindizes, Sekundärindizes, referentielle Integritätsbeschränkungen und DML-Trigger (Data Manipulation Language) zu löschen. Sie können auch ihre Erstellung bis zum Abschluss der Volllastaufgaben verzögern. Während einer Volllast-Aufgabe sind keine Indizes erforderlich und es entstehen Fixkosten für die Wartung, wenn sie vorhanden sind. Da die Aufgabe zum vollständigen Laden Tabellen gruppenweise lädt, wird gegen Einschränkungen bezüglich der referenziellen Integrität verstoßen. Gleichermaßen kann das Einfügen, Aktualisieren und Löschen von Auslösern Fehler verursachen, wenn z. B. für eine zuvor im Bulk-Verfahren geladene Tabelle das Einfügen einer Zeile ausgelöst wird. Andere Arten von Auslöser beeinträchtigen aufgrund der zusätzlichen Verarbeitung ebenfalls die Leistung.

Sie können Primärschlüssel und sekundäre Indizes vor einer Volllast-Aufgabe erstellen, wenn Ihre Daten-Volumes relativ klein sind und die zusätzliche Migrationszeit nicht von Bedeutung ist.

Einschränkungen bezüglich der referenziellen Integrität und Auslöser sollten immer deaktiviert werden.

Für eine Volllast-plus-CDC-Aufgabe empfehlen wir, dass Sie vor der CDC-Aufgabe sekundäre Indizes hinzufügen. Da AWS DMS logische Replikation verwendet, sollten sekundäre Indizes, die DML-Operationen unterstützen, vorhanden sein, um vollständige Tabellen-Scans zu verhindern. Sie können die Replikationsaufgabe vor der CDC-Phase anhalten, um Indizes und referenzielle Integritätsbeschränkungen zu erstellen, bevor Sie die Aufgabe neu starten.

Sie sollten Auslöser unmittelbar vor dem Cutover aktivieren.

Schalten Sie Backups und Transaktionsprotokollierung aus

Bei der Migration zu einer Amazon-RDS-Datenbank wird empfohlen, Sicherungen und Multi-AZ für das Ziel zu deaktivieren, bis Sie zum Cutover bereit sind. Auch wenn die Migration zu Systemen erfolgt, die keine Amazon-RDS-Systeme sind, sollten Sie bis zum Cutover jegliche Protokollierung auf dem Ziel deaktivieren.

Verwenden mehrerer Aufgaben

Manchmal lässt sich durch die Verwendung mehrerer Aufgaben für eine einzelne Migration die Leistung verbessern. Wenn Sie über Tabellensätze verfügen, die nicht an gängigen Transaktionen teilnehmen, können Sie Ihre Migration möglicherweise in mehrere Aufgaben aufteilen. Die Transaktionskonsistenz innerhalb einer Aufgabe wird gewahrt, daher ist es wichtig, dass Tabellen in separaten Aufgaben nicht an gemeinsamen Transaktionen beteiligt sind. Außerdem lesen alle Aufgaben den Transaktionsstrom unabhängig voneinander. Achten Sie also darauf, die Quelldatenbank nicht zu sehr zu belasten.

Sie können mehrere Aufgaben verwenden, um separate Replikations-Streams zu erstellen. Auf diese Weise können Sie die Lesevorgänge auf der Quelle, die Prozesse auf der Replikations-Instance und die Schreibvorgänge in der Zieldatenbank parallelisieren.

Optimieren der Verarbeitung von Änderungen

Standardmäßig verarbeitet AWS DMS Änderungen in einem transaktionalen Modus, bei dem die Integrität der Transaktionen gewahrt wird. Wenn Sie sich temporäre Ausfälle bei der Transaktionsintegrität leisten können, können Sie stattdessen die Option `batch optimized apply` verwenden. Bei dieser Option werden Transaktionen auf effiziente Weise gruppiert und in Stapeln angewendet, um die Effizienz zu erhöhen. Die Verwendung der Option „Batch optimized Apply“ verstößt fast immer gegen Einschränkungen der referenziellen Integrität. Wir empfehlen daher, diese Einschränkungen während des Migrationsprozesses zu deaktivieren und sie im Rahmen des Übergangsprozesses wieder zu aktivieren.

Verwenden Ihres eigenen Vor-Ort-Nameservers

Normalerweise verwendet eine AWS DMS-Replikations-Instance den Domain Name System (DNS)-Resolver in einer Amazon-EC2-Instance, um Domain-Endpunkte aufzulösen. Sie können jedoch Ihren eigenen On-Premises-Namensserver verwenden, um bestimmte Endpunkte aufzulösen, wenn Sie den Amazon-Route-53-Resolver verwenden. Mit diesem Tool können Sie Abfragen zwischen On-Premises-Systemen und AWS durchführen und eingehende und ausgehende Endpunkte, Weiterleitungsregeln und eine private Verbindung verwenden. Zu den Vorteilen der Verwendung eines On-Premises-Namensservers gehören verbesserte Sicherheit und Benutzerfreundlichkeit hinter einer Firewall.

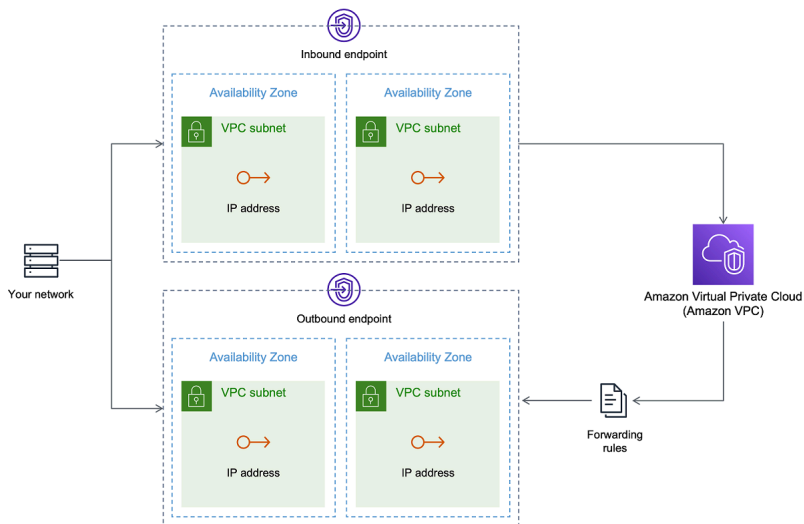
Wenn Sie über eingehende Endpunkte verfügen, können Sie DNS-Abfragen verwenden, die ihren Ursprung On-Premises haben, um von AWS gehostete Domains aufzulösen. Die Endpunkte werden durch Zuweisung der IP-Adresse in jedem Subnetz konfiguriert, für das Sie einen Resolver bereitstellen möchten. Verwenden Sie AWS Direct Connect oder ein Virtual Private Network (VPN), um die Konnektivität zwischen Ihrer On-Premises-DNS-Infrastruktur und AWS herzustellen.

Ausgehende Endpunkte stellen eine Verbindung zu Ihrem On-Premises-Namensserver her. Der Namensserver gewährt nur Zugriff auf IP-Adressen, die in einer Zulassungsliste und in einem ausgehenden Endpunkt enthalten sind. Die IP-Adresse Ihres Nameservers ist die Ziel-IP-Adresse. Wenn Sie eine Sicherheitsgruppe für einen ausgehenden Endpunkt auswählen, wählen Sie dieselbe Sicherheitsgruppe, die von der Replikations-Instance verwendet wird.

Verwenden Sie Weiterleitungsregeln, um Domains auszuwählen, die an den Namensserver weitergeleitet werden sollen. Für einen ausgehenden Endpunkt können mehrere Weiterleitungsregeln vorhanden sein. Der Geltungsbereich der Weiterleitungsregel ist Ihre Virtual Private Cloud (VPC). Mithilfe einer mit einer VPC verknüpften Weiterleitungsregel können Sie einen logisch isolierten Bereich der AWS-Cloud bereitstellen. Von diesem logisch isolierten Bereich aus können Sie AWS-Ressourcen in einem virtuellen Netzwerk starten.

Domains, die in Ihrer On-Premises-DNS-Infrastruktur gehostet werden, können als bedingte Weiterleitungsregeln konfiguriert werden, die ausgehende DNS-Abfragen ermöglichen. Wenn eine Abfrage an eine dieser Domains durchgeführt wird, lösen die Regeln den Versuch aus, DNS-Anforderungen an DNS-Server weiterzuleiten, die mit den Regeln konfiguriert wurden. Auch hier ist eine private Verbindung über AWS Direct Connect oder VPN erforderlich.

Das folgende Diagramm zeigt die Route-53-Resolver-Architektur.



Weitere Informationen zu Route 53 DNS Resolver finden Sie unter [Erste Schritte mit Route 53 Resolver](#) im Entwicklerhandbuch zu Amazon Route 53.

Verwenden von Amazon Route 53 Resolver mit AWS DMS

Sie können einen On-Premises-Namensserver für AWS DMS zur Auflösung von Endpunkten mit [Amazon Route 53 Resolver](#) einrichten.

So erstellen Sie einen On-Premises-Namensserver für AWS DMS basierend auf Route 53

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.
2. Wählen Sie in der Route-53-Konsole die AWS-Region aus, in der Sie Ihren Route 53 Resolver konfigurieren möchten. Der Route 53 Resolver ist spezifisch für eine Region.
3. Wählen Sie die Abfragerichtung – eingehend, ausgehend oder beides.
4. Geben Sie Ihre Konfiguration für eingehende Abfragen an:
 - a. Geben Sie einen Endpunktnamen ein und wählen Sie eine VPC.
 - b. Weisen Sie ein oder mehrere Subnetze innerhalb der VPC zu (wählen Sie beispielsweise zwei für höhere Verfügbarkeit).
 - c. Weisen Sie bestimmte IP-Adressen zu, die als Endpunkte verwendet werden sollen, oder lassen Sie diese durch Route 53 Resolver automatisch zuweisen.
5. Erstellen Sie eine Regel für Ihre lokale Domäne, damit Workloads innerhalb der VPC DNS-Abfragen an Ihre DNS-Infrastruktur weitergeleitet werden können.
6. Geben Sie eine oder mehrere IP-Adressen für Ihre On-Premises-DNS-Server ein.

7. Übermitteln Sie Ihre Regel.

Wenn alles erstellt wurde, ist Ihre VPC mit Ihren eingehenden und ausgehenden Regeln verknüpft und kann mit dem Routing des Datenverkehrs beginnen.

Weitere Informationen zu Route 53 Resolver finden Sie unter [Erste Schritte mit Route 53 Resolver](#) im Entwicklerhandbuch zu Amazon Route 53.

Migrieren von Binary Large Objects (LOBs)

Im Allgemeinen migriert AWS DMS LOB-Daten in zwei Phasen:

1. AWS DMS erstellt eine neue Zeile in der Zieltabelle und füllt die Zeile mit allen Daten mit Ausnahme des zugehörigen LOB-Werts.
2. AWS DMS aktualisiert die Zeile in der Zieltabelle mit den LOB-Daten.

Dieser Migrationsvorgang für LOBs erfordert, dass alle LOB-Spalten in der Zieltabelle während der Migration nullwertfähig sein müssen. Dies gilt auch dann, wenn die LOB-Spalten in der Quelltable nicht nullwertfähig sind. Wenn AWS DMS die Zieltabelle erstellt, werden LOB-Spalten standardmäßig als löscher festgelegt. In manchen Fällen können Sie die Zieltabelle mithilfe eines anderen Mechanismus wie Import oder Export erstellen. Stellen Sie in solchen Fällen sicher, dass die LOB-Spalten Null-Werte zulassen, bevor Sie die Migrationsaufgabe starten.

Für diese Anforderung gibt es eine Ausnahme. Angenommen, Sie führen eine homogene Migration von einer Oracle-Quelle zu einem Oracle-Ziel durch und wählen Limited Lob-Modus. In diesem Fall wird die gesamte Zeile auf einmal gefüllt, einschließlich aller LOB-Werte. In einem solchen Fall kann AWS DMS die LOB-Spalten der Zieltabelle bei Bedarf mit nicht löscheren Einschränkungen erstellen.

Verwenden des begrenzten LOB-Modus

AWS DMS nutzt zwei Methoden, um Leistung und Benutzerfreundlichkeit gegeneinander abzuwägen, wenn Ihre Migration LOB-Werte enthält:

1. Mit Limited LOB-Modus (Begrenzter LOB-Modus) werden alle LOB-Werte bis zu einer vom Benutzer angegebenen Größe (Standard: 32 KB) migriert. LOB-Werte, die größer als das Größenlimit sind, müssen manuell migriert werden. Limited LOB mode (Begrenzter LOB-Modus),

der Standard für alle Migrationsaufgaben, bietet in der Regel die beste Leistung. Sie müssen jedoch sicherstellen, dass die Einstellung des Parameters Max. LOB-Größe korrekt ist. Stellen Sie diesen Parameter sollte auf die maximale LOB-Größe für alle Ihre Tabellen ein.

2. Full LOB mode (Vollständiger LOB-Modus) migriert alle LOB-Daten in Ihren Tabellen unabhängig von der Größe. Full LOB mode (Vollständiger LOB-Modus) bietet die Möglichkeit, alle LOB-Daten in Ihren Tabellen zu verschieben, aber der Prozess kann die Leistung signifikant beeinträchtigen.

Bei einigen Datenbank-Engines, z. B. PostgreSQL, behandelt AWS DMS JSON-Datentypen wie LOBs. Stellen Sie sicher, dass bei Auswahl von Begrenzter LOB-Modus die Option Max. LOB-Größe auf einen Wert festgelegt ist, der nicht dazu führt, dass die JSON-Daten gekürzt werden.

AWS DMS bietet vollen Support für die Verwendung großer Objektdatentypen (BLOBs, CLOBs und NCLOBs). Für die folgenden Quellendpunkte werden LOBs uneingeschränkt unterstützt:

- Oracle
- Microsoft SQL Server
- ODBC

Für die folgenden Zielendpunkte werden LOBs uneingeschränkt unterstützt:

- Oracle
- Microsoft SQL Server

Der folgende Zielendpunkt verwendet eine eingeschränkte LOB-Unterstützung. Sie können keine unbegrenzte LOB-Größe für diesen Zielendpunkt verwenden.

- Amazon Redshift
- Amazon S3

Für Endpunkte mit uneingeschränkter LOB-Unterstützung können Sie auch eine Obergrenze für LOB-Datentypen festlegen.

Verbesserte LOB-Leistung

Bei der Migration von LOB-Daten können Sie die folgenden verschiedenen LOB-Optimierungseinstellungen angeben.

LOB-Einstellungen pro Tabelle

Mithilfe der LOB-Einstellungen pro Tabelle können Sie LOB-Einstellungen auf Aufgabenebene für einige oder alle Tabellen überschreiben. Definieren Sie dazu die `lob-settings` in Ihrer `table-settings`-Regel. Im Folgenden finden Sie eine Beispieltabelle, die einige große LOB-Werte enthält.

```

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

```

Erstellen Sie dann eine Migrationsaufgabe und ändern Sie die LOB-Behandlung für Ihre Tabelle mithilfe der neuen `lob-settings`-Regel. Der `bulk-max-size`-Wert bestimmt die maximale LOB-Größe (KB). Es wird gekürzt, wenn diese größer als die angegebene Größe ist.

```

{
  "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"
}
]
}
```

Selbst wenn diese AWS DMS-Aufgabe mit `FullLobMode : true` erstellt wurde, weisen die LOB-Einstellungen pro Tabelle AWS DMS an, dass die LOB-Daten in dieser bestimmten Tabelle auf 16 000 gekürzt werden. Sie können dies anhand der Aufgabenprotokolle überprüfen.

```
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
```

Inline-LOB-Einstellungen

Wenn Sie eine AWS DMS-Aufgabe erstellen, bestimmt der LOB-Modus, wie LOBs behandelt werden.

Der vollständige LOB-Modus und der eingeschränkte LOB-Modus haben jeweils ihre eigenen Vor- und Nachteile. Der Inline-LOB-Modus kombiniert die Vorteile des vollständigen LOB-Modus und des eingeschränkten LOB-Modus.

Sie können den Inline-LOB-Modus verwenden, wenn Sie sowohl kleine als auch große LOBs replizieren müssen und die meisten LOBs klein sind. Wenn Sie diese Option wählen, überträgt die AWS DMS-Aufgabe bei Volllast die kleinen LOBs inline, was effizienter ist. Die AWS DMS-Aufgabe überträgt die großen LOBs, indem sie eine Suche in der Quelltable durchführt.

Während der Änderungsverarbeitung werden sowohl kleine als auch große LOBs repliziert, indem eine Suche in der Quelltable durchgeführt wird.

Wenn Sie den Inline-LOB-Modus verwenden, überprüft die AWS DMS-Aufgabe alle LOB-Größen, um zu bestimmen, welche LOB-Größen inline übertragen werden sollen. LOBs, die größer als die

angegebene Größe sind, werden im vollständigen LOB-Modus repliziert. Wenn Sie also wissen, dass die meisten LOBs größer als die angegebene Einstellung sind, ist es besser, diese Option nicht zu verwenden. Lassen Sie stattdessen eine unbegrenzte LOB-Größe zu.

Sie konfigurieren diese Option mithilfe eines Attributs in den Aufgabeneinstellungen, `InlineLobMaxSize`, das nur verfügbar ist, wenn `FullLobMode` auf `true` gesetzt ist. Der Standardwert für `InlineLobMaxSize` ist 0 und der Bereich liegt zwischen 1 und 102 400 Kilobyte (100 MB).

Sie könnten beispielsweise die folgenden AWS DMS-Aufgabeneinstellungen verwenden. Hier führt die Einstellung von `InlineLobMaxSize` auf einen Wert von 5 dazu, dass alle LOBs, die kleiner als oder gleich 5 KiB (5 120 Byte) sind, inline übertragen werden.

```
{
  "TargetMetadata": {
    "TargetSchema": "",
    "SupportLobs": true,
    "FullLobMode": true,
    "LobChunkSize": 64,
    "LimitedSizeLobMode": false,
    "LobMaxSize": 32,
    "InlineLobMaxSize": 5,
    "LoadMaxFileSize": 0,
    "ParallelLoadThreads": 0,
    "ParallelLoadBufferSize": 0,
    "BatchApplyEnabled": false,
    "TaskRecoveryTableEnabled": false},
  . . .
}
```

Verbessern der Leistung beim Migrieren umfangreicher Tabellen mit Zeilenfilterung

Wenn Sie die Leistung beim Migrieren großer Tabellen verbessern möchten, können Sie die Migration in mehr als eine Aufgabe unterteilen. Sie können einen Schlüssel oder einen Partitionsschlüssel verwenden, um die Migration mittels Zeilenfilterung in mehrere Aufgaben aufzuteilen. Mit einer Ganzzahl-Primärschlüssel-ID von 1 bis 8.000.000 können Sie mittels Zeilenfilterung acht Aufgaben erstellen, um jeweils eine Million Datensätze zu migrieren.

So wenden Sie die Zeilenfilterung in der Konsole an:

1. Öffnen Sie die AWS Management Console.
2. Wählen Sie Aufgaben und erstellen Sie eine neue Aufgabe.
3. Wählen Sie die Registerkarte Tabellenzuordnungen aus und erweitern Sie Auswahlregeln.
4. Wählen Sie Neue Auswahlregel hinzufügen. Sie können dann einen Spaltenfilter mit der Bedingung „kleiner oder gleich“, „größer oder gleich“, „gleich“ oder einer Bereichsbedingung (zwischen zwei Werten) hinzufügen. Weitere Informationen zur Spaltenfilterung finden Sie unter [Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole](#).

Alternativ können Sie bei einer großen Tabelle, die nach Datum partitioniert ist, Daten basierend auf dem Datum migrieren. Angenommen, Sie haben eine Tabelle nach Monat partitioniert und nur die Daten des aktuellen Monats werden aktualisiert. In diesem Fall können Sie für jede statische Monatspartition eine Volllastaufgabe und für die aktuell aktualisierte Partition eine Volllast-plus-CDC-Aufgabe erstellen.

Wenn Ihre Tabelle über einen einspaltigen Primärschlüssel oder einen eindeutigen Index verfügt, können Sie von Ihrer AWS DMS-Aufgabe die Tabelle segmentieren lassen, indem sie ein paralleles Laden des Typs „Bereiche“ verwendet, um die Daten parallel zu laden. Weitere Informationen finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Fortlaufende Replikation

AWS DMS bietet eine fortlaufende Replikation von Daten, um die Synchronität der Quell- und Zieldatenbank aufrechtzuerhalten. Es wird allerdings nur eine begrenzte Menge an DDL-Anweisungen (Data Definition Language) repliziert. AWS DMS verbreitet keine Elemente wie Indizes, Benutzer, Berechtigungen, gespeicherte Prozeduren und andere Datenbankänderungen, die sich nicht direkt auf die Tabellendaten beziehen.

Wenn Sie die fortlaufende Replikation verwenden möchten, sollten Sie die Option Multi-AZ bei der Erstellung Ihrer Replikations-Instance aktivieren. Die Option Multi-AZ bietet für die Replikations-Instance hohe Verfügbarkeit und Failover-Unterstützung. Diese Option kann sich jedoch auf die Leistung auswirken und die Replikation verlangsamen, während Änderungen auf das Zielsystem angewendet werden.

Bevor Sie Ihre Quell- oder Zieldatenbanken aktualisieren, empfehlen wir, dass Sie alle AWS DMS-Aufgaben beenden, die auf diesen Datenbanken ausgeführt werden. Setzen Sie die Aufgaben fort, nachdem Ihre Upgrades abgeschlossen sind.

Während der fortlaufenden Replikation ist es wichtig, die Netzwerkbandbreite zwischen Ihrem Quelldatenbanksystem und Ihrer AWS DMS-Replikations-Instance zu ermitteln. Stellen Sie sicher, dass das Netzwerk während der laufenden Replikation keine Engpässe verursacht.

Es ist auch wichtig, die Änderungsrate und die Generierung von Archivprotokollen pro Stunde auf Ihrem Quelldatenbanksystem zu ermitteln. Auf diese Weise können Sie sich einen Überblick über den Durchsatz verschaffen, den Sie bei fortlaufender Replikation erzielen können.

Reduzieren der Last Ihrer Quelldatenbank

AWS DMS nutzt einige Ressourcen in Ihrer Quelldatenbank. Während einer Aufgabe zum vollständigen Laden führt AWS DMS eine vollständige Tabellenprüfung der Quelltablelle parallel für jede verarbeitete Tabelle aus. Zudem fragt jede Aufgabe, die Sie als Teil einer Migration erstellen, die Quelldatenbank im Rahmen des CDC-Prozesses nach Änderungen ab. Bei einigen Quelldatenbanken, wie Oracle, müssen Sie möglicherweise die in das Änderungsprotokoll Ihrer Datenbanken geschriebene Datenmenge erhöhen, damit AWS DMS den CDC-Vorgang durchführen kann.

Wenn Sie feststellen, dass Ihre Quelldatenbank überlastet ist, können Sie die Anzahl der Aufgaben und/oder Tabellen pro Aufgabe für die Migration reduzieren. Da die einzelnen Aufgaben unabhängig voneinander Änderungen erhalten, kann die Änderungserfassungslast durch die Konsolidierung der Aufgaben verringert werden.

Reduzierung von Engpässen in Ihrer Zieldatenbank

Versuchen Sie während der Migration, alle Prozesse zu entfernen, die ggf. um die Schreibressourcen in Ihrer Zieldatenbank konkurrieren.

- Schalten Sie unnötige Auslöser aus.
- Schalten Sie sekundäre Indizes beim ersten Laden aus und aktivieren Sie sie später während der fortlaufenden Replikation wieder.
- Bei Amazon-RDS-Datenbanken empfiehlt es sich, Backups und Multi-AZ bis zum Cutover auszuschalten.

- Bei der Migration zu Nicht-RDS-Systemen sollten Sie jegliche Protokollierung auf dem Ziel bis zum Cutover deaktivieren.

Verwendung der Datenvalidierung während der Migration

Um sicherzustellen, dass Ihre Daten korrekt von der Quelle zum Ziel migriert werden, empfehlen wir nachdrücklich, Datenvalidierung zu verwenden. Wenn Sie die Datenvalidierung für eine Aufgabe aktivieren, vergleicht AWS DMS die Quell- und Zieldaten sofort, nachdem eine Tabelle vollständig geladen wurde.

Die Datenvalidierung funktioniert mit den folgenden Datenbanken dort, wo AWS DMS sie als Quell- und Ziel-Endpunkte unterstützt:

- Oracle
- PostgreSQL
- MySQL
- MariaDB
- Microsoft SQL Server
- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- IBM Db2 (LUW)
- Amazon Redshift

Weitere Informationen finden Sie unter [Datenvalidierung mit AWS DMS](#).

Überwachen Ihrer AWS DMS-Aufgaben mithilfe von Metriken

Sie haben mehrere Möglichkeiten, die Metriken für Ihre Aufgaben mithilfe der AWS DMS-Konsole zu überwachen:

Host-Metriken

Host-Metriken finden Sie auf der Registerkarte CloudWatch Metriken für jede bestimmte Replikations-Instance. Hier können Sie überwachen, ob Ihre Replikations-Instance die richtige Größe hat.

Metriken für die Replikationsaufgabe

Metriken für Replikationsaufgaben, einschließlich eingehender und festgeschriebener Änderungen, sowie die Latenz zwischen dem Replikationshost und Quell-/Zielfdatenbanken finden Sie auf der Registerkarte CloudWatch Metriken für jede bestimmte Aufgabe.

Tabellenmetriken

Sie finden einzelne Tabellenmetriken auf der Registerkarte Tabellenstatistiken für jede einzelne Aufgabe. Zu diesen Metriken gehören die folgenden Zahlen:

- Zeilen, die während des Vollladevorgangs geladen wurden.
- Einfüge-, Aktualisierungs- und Löschvorgänge seit dem Start der Aufgabe.
- DDL-Operationen seit dem Start der Aufgabe.

Weitere Informationen zum Überwachen von Metriken finden Sie unter [Überwachen von AWS-DMS-Aufgaben](#).

Ereignisse und Benachrichtigungen

AWS DMS verwendet Amazon SNS zum Bereitstellen von Benachrichtigungen, wenn ein AWS DMS-Ereignis auftritt, z. B. beim Erstellen oder Löschen einer Replikations-Instance. Sie können mit diesen Benachrichtigungen in jeder Form arbeiten, die von Amazon SNS für eine AWS-Region unterstützt wird. Dazu können E-Mail-Nachrichten, Textnachrichten oder Aufrufe an einen HTTP-Endpunkt gehören.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon-SNS-Ereignissen und Benachrichtigungen in AWS Database Migration Service](#).

Verwenden des Aufgabenprotokolls für die Behebung von Migrationsproblemen

In einigen Fällen kann AWS DMS auf Probleme stoßen, bei denen Warnungen oder Fehlermeldungen nur im Aufgabenprotokoll erscheinen. Insbesondere Probleme mit der Kürzung von Daten oder Ablehnung von Zeilen aufgrund von Fremdschlüsselverstößen werden nur in das Aufgabenprotokoll geschrieben. Daher ist es wichtig, das Aufgabenprotokoll bei einer

Datenbankmigration zu überprüfen. Um das Aufgabenprotokoll anzuzeigen, konfigurieren Sie Amazon CloudWatch als Teil der Aufgabenerstellung.

Weitere Informationen finden Sie unter [Überwachen von Replikationsaufgaben mit Amazon CloudWatch](#).

Problembehandlung bei Replikationsaufgaben mit Time Travel

Um AWS DMS-Migrationsprobleme zu beheben, können Sie mit Time Travel arbeiten. Weitere Informationen zu Time Travel finden Sie unter [Time-Travel-Aufgabeneinstellungen](#).

Seien Sie sich der folgenden Überlegungen bewusst, wenn Sie mit Time Travel arbeiten:

- Um zu hohe Belastung für eine DMS-Replikations-Instance zu vermeiden, aktivieren Sie Time Travel nur für Aufgaben, die Debugging benötigen.
- Wenn Sie Time Travel zur Fehlerbehebung bei Replikationsaufgaben verwenden, die möglicherweise mehrere Tage dauern, sollten Sie die Metriken der Replikations-Instance im Hinblick auf den Ressourcenaufwand überwachen. Dieses Konzept eignet sich insbesondere für Fälle, in denen Quelldatenbanken über einen längeren Zeitraum mit hohen Transaktionslasten belastet werden. Weitere Details finden Sie unter [Überwachen von AWS-DMS-Aufgaben](#).
- Ist die Einstellung `EnableRawData` der Time-Travel-Aufgabe auf `true` gesetzt, ist die Auslastung des Aufgabenspeichers während der DMS-Replikation möglicherweise höher als wenn Time Travel nicht aktiviert ist. Wenn Sie Time Travel für längere Zeiträume aktivieren, überwachen Sie Ihre Aufgabe.
- Derzeit können Sie Time Travel nur auf Aufgabenebene aktivieren. Änderungen an allen Tabellen werden in Time-Travel-Protokollen protokolliert. Wenn Sie Probleme mit bestimmten Tabellen in einer Datenbank mit hohem Transaktionsvolumen beheben möchten, erstellen Sie eine separate Aufgabe.

Ändern des Benutzers und des Schemas für ein Oracle-Ziel

Wenn Sie Oracle als Ziel verwenden, migriert AWS DMS die Daten zu dem Schema, das im Besitz des Zielpunkt-Benutzers ist.

Nehmen Sie zum Beispiel an, dass Sie ein Schema mit dem Namen `PERFDATA` zu einem Oracle-Zielpunkt migrieren und der Benutzername des Zielpunktes `MASTER` ist. AWS DMS

stellt eine Verbindung mit dem Oracle-Ziel als MASTER her und füllt das MASTER-Schema mit Datenbankobjekten von PERFDATA.

Um dieses Verhalten zu übergehen, müssen Sie eine Schematransformation bereitstellen. Wenn Sie beispielsweise die PERFDATA-Schemaobjekte zu einem PERFDATA-Schema am Zielpunkt migrieren möchten, können Sie die folgende Transformation verwenden.

```
{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "object-locator": {
    "schema-name": "PERFDATA"
  },
  "rule-target": "schema",
  "rule-action": "rename",
  "value": "PERFDATA"
}
```

Weitere Informationen zu Umwandlungen finden Sie unter [Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON](#).

Ändern von Tabellen- und Index-Tabellenräumen für ein Oracle-Ziel

Wenn Oracle als Ziel verwendet wird, migriert AWS DMS alle Tabellen und Indizes zum Standardtabellenraum auf dem Ziel. Nehmen Sie beispielsweise an, dass Ihre Quelle eine andere Datenbank-Engine als Oracle ist. Alle Zieltabellen und Indizes werden zum selben Standard-Tabellenräumen migriert.

Um dieses Verhalten zu übergehen, müssen Sie entsprechende Transformationen von Tabellenräumen bereitstellen. Nehmen Sie beispielsweise an, dass Sie Tabellen und Indizes zu Tabellen- und Index-Tabellenräumen im Oracle-Ziel migrieren möchten, die nach dem Schema in der Quelle benannt sind. In diesem Fall können Sie Transformationen verwenden, die den folgenden ähnlich sind. Hier hat das Schema an der Quelle den Namen INVENTORY und die entsprechenden Tabellen- und Index-Tabellenräume im Ziel haben die Namen INVENTORYTBL und 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"
}
```

Weitere Informationen zu Umwandlungen finden Sie unter [Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON](#).

Wenn Oracle sowohl Quelle als auch Ziel ist, können Sie vorhandene Tabellen- oder Index-Tabellenraum-Zuweisungen beibehalten, indem Sie das zusätzliche Verbindungsattribut für Oracle Source, `enableHomogenousTablespace=true`, festlegen. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).

Aktualisieren der Engine-Version einer Replikations-Instance

AWS veröffentlicht regelmäßig neue Versionen der AWS DMS-Replikations-Engine-Software mit neuen Funktionen und Leistungsverbesserungen. Jede Version der Replikations-Engine-Software verfügt über eine eigene Versionsnummer. Es ist wichtig, die vorhandene Version Ihrer AWS DMS-Replikations-Instance zu testen, auf der eine Produktionslast ausgeführt wird, bevor Sie Ihre Replikations-Instance auf eine neuere Version aktualisieren. Weitere Informationen zu automatischen Versions-Upgrades finden Sie unter [AWS DMS-Versionshinweise](#).

Grundlegendes zu Ihren Migrationskosten

AWS Database Migration Service unterstützt Sie bei der einfachen und sicheren Migration zu AWS zu niedrigen Kosten. Sie zahlen nur für Ihre Replikations-Instances und zusätzlichen Protokollspeicher. Jede Datenbank-Migrations-Instance beinhaltet ausreichend Speicherplatz für Auslagerungsspeicher, Replikationsprotokolle und Daten-Cache für die meisten Replikationen. Die eingehende Datenübertragung ist kostenlos.

Möglicherweise benötigen Sie beim ersten Laden oder während der Spitzenlastzeit mehr Ressourcen. Mithilfe von CloudWatch-Metriken können Sie die Ressourcenauslastung der Replikations-Instance genau überwachen. Anschließend können Sie die Größe der Replikations-Instance je nach Nutzung hoch- und herunterskalieren.

Weitere Informationen zur Schätzung Ihrer Migrationskosten finden Sie unter:

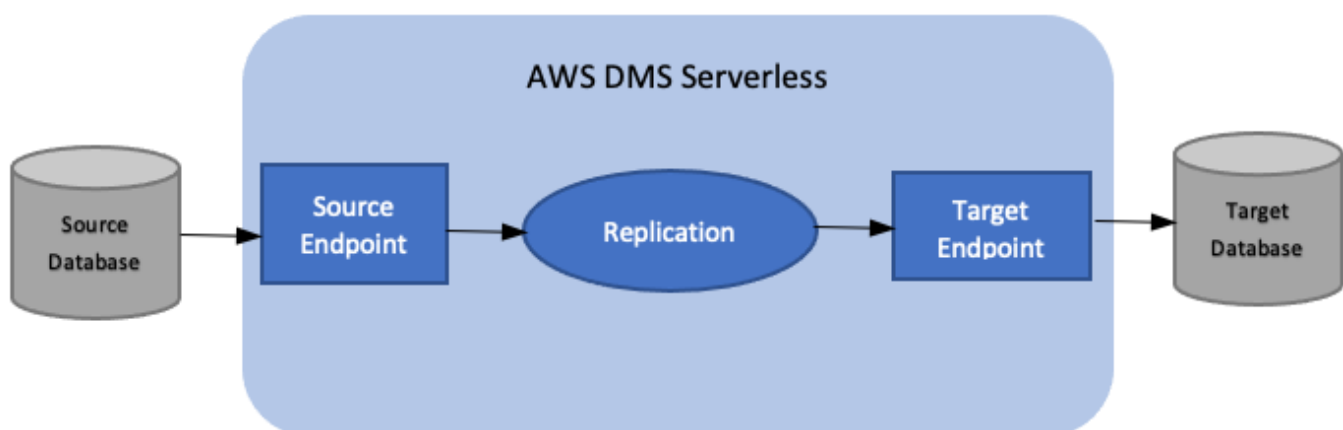
- [AWS Database Migration Service-Preise](#)
- [AWS Pricing Calculator](#)

Arbeiten mit AWS DMS Serverless

AWS DMS Serverless ist eine Funktion, die automatische Bereitstellung, Skalierung, integrierte Hochverfügbarkeit und ein pay-for-use Abrechnungsmodell bietet, um die betriebliche Flexibilität zu erhöhen und Ihre Kosten zu optimieren. Das Serverless-Feature macht Aufgaben zur Verwaltung der Replikation von Instances wie Kapazitätsschätzung, Bereitstellung, Kostenoptimierung und Verwaltung von Versionen und Patches der Replikations-Engines überflüssig.

Bei AWS DMS Serverless erstellen Sie, ähnlich wie bei der aktuellen Funktionalität von AWS DMS (in diesem Dokument als AWS DMS Standard bezeichnet), Quell- und Zielverbindungen mithilfe von Endpunkten. Nachdem Sie Ihre Quell- und Zielendpunkte erstellt haben, erstellen Sie eine Replikationskonfiguration, die Konfigurationseinstellungen für die jeweilige Replikation enthält. Sie können die Replikationen verwalten, indem Sie sie starten, stoppen, ändern oder löschen. Jede Replikation hat Einstellungen, die entsprechend den Anforderungen Ihrer Datenbankmigration konfiguriert werden können. Sie geben diese Einstellungen entweder mithilfe einer JSON-Datei oder mithilfe des AWS DMS Abschnitts von an. AWS Management Console Weitere Informationen zu Replikationseinstellungen finden Sie unter [Arbeiten mit AWS DMS Endpunkten](#). Nach dem Start der Replikation stellt AWS DMS Serverless eine Verbindung zur Quelldatenbank her und erfasst die Datenbankmetadaten, um den Replikations-Workload zu analysieren. Anhand dieser Metadaten wird die erforderliche Kapazität AWS DMS berechnet und bereitgestellt und die Datenreplikation gestartet.

Das folgende Diagramm zeigt den Prozess der AWS DMS serverlosen Replikation.



Note

AWS DMS Serverless verwendet die Standard-Engine-Version. Informationen zur Standard-Engine-Version finden Sie unter [Versionshinweise](#).

In den folgenden Themen finden Sie weitere Informationen zu AWS DMS Serverless.

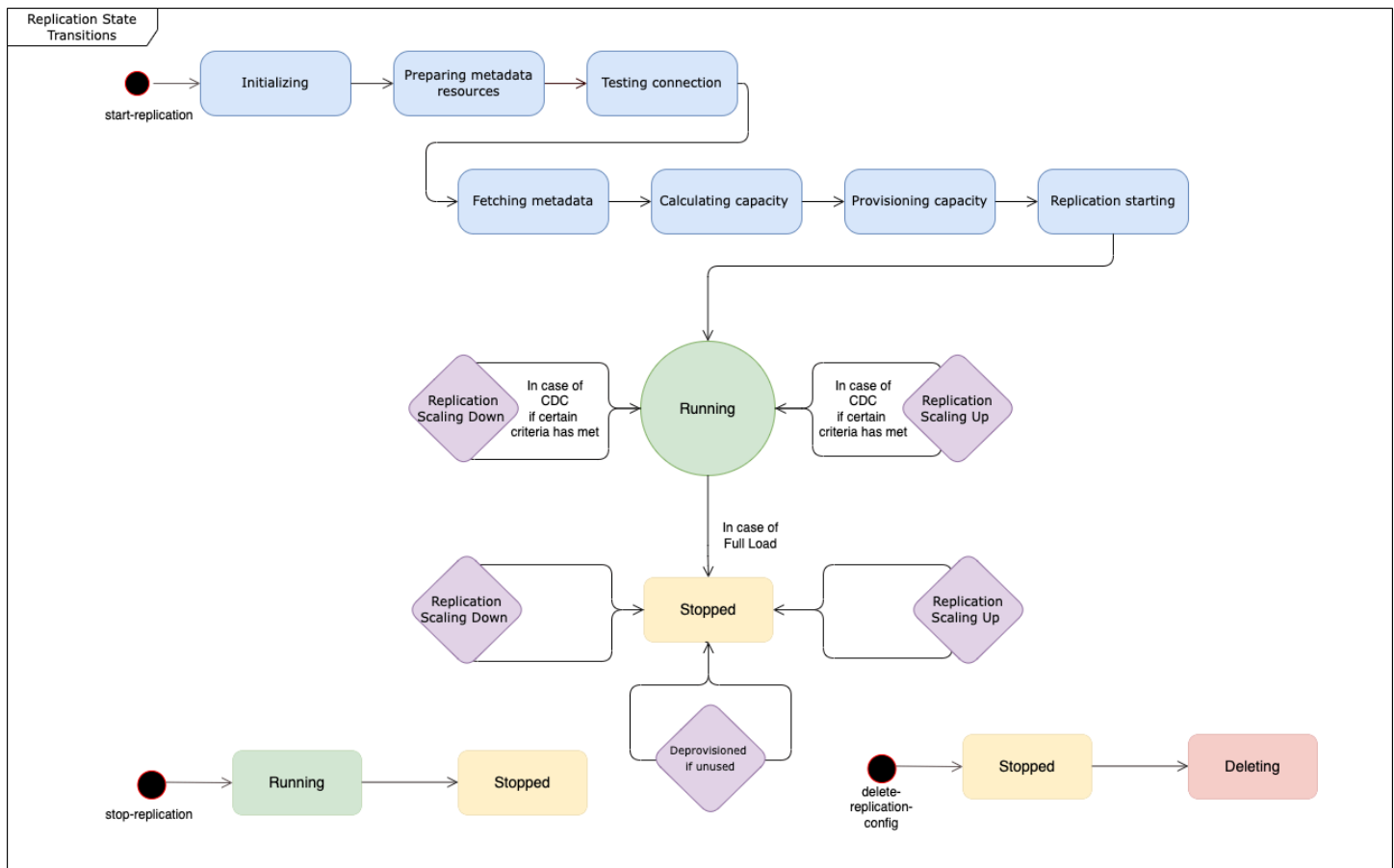
Themen

- [AWS DMS Serverlose Komponenten](#)
- [AWS DMS Einschränkungen ohne Server](#)

AWS DMS Serverlose Komponenten

Um die Ressourcen zu verwalten, die für die Durchführung einer Replikation benötigt werden, verfügt AWS DMS Serverless über detaillierte Statusinformationen, die verschiedene interne Aktionen des Dienstes aufzeigen. Wenn Sie die Replikation starten, berechnet AWS DMS Serverless die Kapazitätslast, stellt die berechnete Kapazität bereit und startet die Datenreplikation gemäß den folgenden Replikationsstatus.

Das folgende Diagramm zeigt die Zustandsübergänge für eine AWS DMS serverlose Replikation.



- Der erste Status nach dem Start der Replikation ist Initialisieren. In diesem Status werden alle erforderlichen Parameter initialisiert.
- Zu den unmittelbar folgenden Status gehören Vorbereiten von Metadatenressourcen, Verbindung testen und Abrufen von Metadaten. In diesen Zuständen stellt AWS DMS Serverless eine Verbindung zu Ihrer Quelldatenbank her, um die Informationen abzurufen, die zur Vorhersage der benötigten Kapazität erforderlich sind.
 - Wenn der Replikationsstatus Verbindung testen lautet, überprüft AWS DMS Serverless, ob die Verbindung zu Ihren Quell- und Zieldatenbanken erfolgreich eingerichtet wurde.
 - Der Replikationsstatus nach Verbindung testen lautet Abrufen von Metadaten. AWS DMS ruft hier die Informationen ab, die zur Berechnung der Kapazität benötigt werden.
 - Sobald die erforderlichen Informationen von AWS DMS abgerufen wurden, lautet der nächste Status „Kapazität berechnen“. Hier berechnet das System die Größe der zugrunde liegenden Ressourcen, die für die Durchführung der Replikation erforderlich sind.
- Der Statusübergang nach Kapazität berechnen ist der zu Kapazität bereitstellen. Während sich die Replikation in diesem Status befindet, initialisiert AWS DMS Serverless die zugrunde liegenden Rechenressourcen.

- Der Replikationsstatus nach erfolgreicher Bereitstellung aller Ressourcen lautet Starten der Replikation. In diesem Zustand beginnt AWS DMS Serverless mit der Replikation von Daten. Die Phasen einer Replikation umfassen Folgendes:
 - **Volllast:** In dieser Phase repliziert DMS den Quelldatenspeicher so, wie er zu Beginn der Replikation war.
 - **CDC (initial):** In dieser Phase repliziert DMS die Änderungen am Quelldatenspeicher, die während der Volllastphase vorgenommen wurden. DMS führt diese Phase nur aus, wenn die Aufgabeneinstellung `StopTaskCachedChangesNotApplied` `false` ist.
 - **CDC (fortlaufend):** Nach der ersten CDC-Phase repliziert DMS Änderungen in der Quelldatenbank, sobald sie auftreten. DMS setzt die Replikation nach der ersten CDC-Phase nur dann fort, wenn die Aufgabeneinstellung `StopTaskCachedChangesApplied` `false` ist.
- Der endgültige Status ist `Running`. Im Status `Running` erfolgt die Replikation der Daten.
- Eine Replizierung, die Sie beenden, geht in den Status `Gestoppt` über. Sie können eine gestoppte Replizierung unter den folgenden Umständen neu starten:
 - Sie können eine Replikation, deren Bereitstellung durch DMS aufgehoben wurde, nicht neu starten.
 - Sie können eine gestoppte reine CDC- oder Volllast- und CDC-Replikation mithilfe der Aktion [StartReplication](#) neu starten. [StartReplication](#) Sie können eine gestoppte Replizierung nicht über die Konsole neu starten.
 - Sie können eine gestoppte Replikation, die PostgreSQL als Engine verwendet, nicht neu starten.

Dieses Thema enthält die folgenden Abschnitte.

- [Unterstützte Engine-Versionen](#)
- [Erstellen einer Serverless-Anwendung](#)
- [AWS DMS Serverlose Replikationen ändern](#)
- [Compute-Konfiguration](#)
- [Grundlegendes zur automatischen Skalierung im serverlosen Modus AWS DMS](#)
- [Überwachung serverloser Replikationen AWS DMS](#)

Für AWS DMS Serverless gibt es im linken Navigationsbereich der AWS DMS Konsole eine neue Option: Serverlose Replikationen. Für Serverless-Replikationen geben Sie Replikationen anstelle von Replikations-Instance-Typen oder Aufgaben an, um eine Replikation zu definieren. Darüber hinaus geben Sie die maximalen und minimalen DMS-Kapazitätseinheiten (DCUs) an, die DMS für

die Replikation bereitstellen soll. Eine DCU besteht aus 2 GB RAM. AWS DMS stellt Ihrem Konto jede DCU in Rechnung, die Ihre Replikation derzeit verwendet. AWS DMS Preisinformationen finden Sie unter Preise für den [AWS Database Migration Service](#).

AWS DMS stellt dann automatisch Replikationsressourcen auf der Grundlage Ihrer Tabellenzuordnungen und der prognostizierten Größe Ihrer Arbeitslast bereit. Diese Kapazitätseinheit ist ein Wert im Bereich der von Ihnen angegebenen Mindest- und Höchstwerte für Kapazitätseinheiten.

Unterstützte Engine-Versionen

Mit AWS DMS Serverless müssen Sie keine Engine-Versionen auswählen und verwalten, da der Service diese Einstellung übernimmt. AWS DMS Serverless unterstützt die folgenden Quellen:

- Microsoft SQL Server
- PostgreSQL-kompatible Datenbanken
- MySQL-kompatible Datenbanken
- MariaDB
- Oracle
- IBM Db2

AWS DMS Serverless unterstützt die folgenden Ziele:

- Microsoft SQL Server
- PostgreSQL
- MySQL-kompatible Datenbanken
- Oracle
- Amazon S3
- Amazon-Redshift
- Amazon-DynamoDB
- Amazon-Kinesis-Data-Streams
- Amazon Managed Streaming für Apache Kafka
- OpenSearch Amazon-Dienst
- Amazon DocumentDB (mit MongoDB-Kompatibilität)
- Amazon Neptune

Als Teil von AWS DMS Serverless haben Sie Zugriff auf Konsolenbefehle, mit denen Sie AWS DMS serverlose Replikationen erstellen, konfigurieren, starten und verwalten können. Um diese Befehle im Bereich Serverless-Replikationen der Konsole auszuführen, müssen Sie einen der folgenden Schritte durchführen:

- Richten Sie eine neue AWS Identity and Access Management (IAM-) Richtlinie und eine IAM-Rolle ein, an die Sie diese Richtlinie anhängen möchten.
- Verwenden Sie eine AWS CloudFormation Vorlage, um den Zugriff bereitzustellen, den Sie benötigen.

AWS DMS Serverless setzt voraus, dass in Ihrem Konto eine Service Linked Role (SLR) vorhanden ist. AWS DMS verwaltet die Erstellung und Verwendung dieser Rolle. Weitere Informationen darüber, wie Sie sicherstellen können, dass Sie über die erforderliche SLR verfügen, finden Sie unter [Serviceverknüpfte Rolle für AWS DMS Serverless](#).

Erstellen einer Serverless-Anwendung

Gehen Sie wie folgt vor, um eine serverlose Replikation zwischen zwei vorhandenen AWS DMS Endpunkten zu erstellen. Informationen zum Erstellen von AWS DMS Endpunkten finden Sie unter [Erstellen der Quell- und Zielendpunkte](#)

Erstellen einer Serverless-Anwendung

1. Melden Sie sich unter <https://console.aws.amazon.com/dms/v2/> bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich Serverless-Replikationen und dann Replikation erstellen aus.
3. Geben Sie auf der Seite Replikation erstellen Ihre Konfiguration für die Serverless-Replikation an:

Option	Aktion
Name	Geben Sie einen Namen zur Identifizierung der Replikation ein, z. B. DMS-replication .
Beschreibender Amazon-Resource Name (ARN) – optional	Lassen Sie das Feld leer.

Option	Aktion
Endpunkt der Quelldatenbank	Wählen Sie vorhandene Endpunkte in Ihrem Konto aus. Beachten Sie, dass AWS DMS Serverless nur eine Teilmenge der Endpunkttypen unterstützt, die der AWS DMS Standard unterstützt.
Endpunkt der Zieldatenbank	Wählen Sie vorhandene Endpunkte in Ihrem Konto aus. Beachten Sie, dass AWS DMS Serverless nur eine Teilmenge der Endpunkttypen unterstützt, die der Standard unterstützt. AWS DMS
Typ der Replikation	<p>Wählen Sie einen Replikationstyp, der Ihren Anforderungen entspricht:</p> <ul style="list-style-type: none"> • Volllast: AWS DMS migriert nur vorhandene Daten. • Vollständige Datenerfassung und Erfassung von Änderungsdaten (CDC): AWS DMS migriert vorhandene Daten und Änderungen, die während der Replikation auftreten. • Change Data Capture (CDC): migriert AWS DMS nur Änderungen, die nach dem Start der Replikation vorgenommen werden.

Legen Sie im Abschnitt **Einstellungen** die Einstellungen fest, die für Ihre Replikation erforderlich sind.

Richten Sie im Abschnitt **Tabellenzuordnungen** die Tabellenzuordnung ein, um Regeln für die Auswahl und Filterung von Daten zu definieren, die Sie replizieren. Bevor Sie Ihre Zuweisung angeben, sollten Sie den Abschnitt in der Dokumentation zur Zuweisung der Datentypen für Ihre Quell- und Zieldatenbank lesen.

Legen Sie im Abschnitt **Compute-Einstellungen** die folgenden Einstellungen fest. Informationen zu den Compute-Config-Einstellungen finden Sie unter [Compute-Konfiguration](#).

Option	Aktion
VPC	Wählen Sie eine vorhandene VPC aus.
Subnetzgruppe	Wählen Sie eine vorhandene Subnetzgruppe aus.
VPC-Sicherheitsgruppe(n)	Wählen Sie Standard, falls noch keine Auswahl getroffen wurde.
AWS KMS-Schlüssel	Lassen Sie (Standard) aws/dmsausgewählt.
Bereitstellung	Lassen Sie dies unverändert.
Availability Zone	Lassen Sie dies unverändert.
Minimale DMS-Kapazitätseinheiten (DCU) – (optional)	Lassen Sie das Feld leer, um den Standardwert von 1 DCU zu verwenden.
Maximale DMS-Kapazitätseinheiten (DCU)	Wählen Sie 16 DCU.

Lassen Sie die Einstellungen unter **Wartung** unverändert.

4. Wählen Sie Replikation erstellen.

AWS DMS erstellt eine serverlose Replikation, um Ihre Migration durchzuführen.

AWS DMS Serverlose Replikationen ändern

Verwenden Sie die Aktion `modify-replication-config`, um Ihre Replikationskonfiguration zu ändern. Sie können nur eine AWS DMS Replikationskonfiguration ändern, die sich im Status `CREATEDSTOPPED`, oder `FAILED` befindet.

Um eine serverlose Replikationskonfiguration zu ändern, verwenden Sie den AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.

2. Wählen Sie im Navigationsbereich die Option Serverless-Replikationen aus.
3. Wählen Sie die Replikation aus, die Sie ändern möchten. In der folgenden Tabelle werden die Änderungen beschrieben, die Sie auf der Grundlage des aktuellen Status der Replikation vornehmen können.

Einstellung	Beschreibung	Zulässige Zustände
Name	Sie können den Namen der Replikation ändern. Geben Sie einen Namen für die Replikation an, der aus 8 bis 16 druckbaren ASCII-Zeichen (ohne /, " und @) besteht. Der Name sollte für Ihr Konto und die gewählte AWS -Region eindeutig sein. Sie können dem Namen einige Details hinzufügen, z. B. die AWS Region und die Aufgabe, die Sie gerade ausführen, zum Beispiel: west2-mysql1mysql-config1 .	ReplicationState ist CREATED, STOPPED oder FAILED.
Endpunkt der Quelldatenbank	Wählen Sie einen neuen vorhandenen Quellendpunkt als Quelle für die Replikation.	ReplicationState ist CREATED oder FAILED, wenn ProvisionState null ist.
Endpunkt der Zieldatenbank	Wählen Sie einen neuen vorhandenen Zielendpunkt als Ziel für die Replikation.	ReplicationState ist CREATED oder FAILED, wenn ProvisionState null ist.
Typ der Replikation	Sie können den Typ einer Serverless-Replikation ändern.	ReplicationState ist CREATED oder FAILED, wenn ProvisionState null ist.

Einstellung	Beschreibung	Zulässige Zustände
Einstellungen für die Replikation	Sie können die Replikationseinstellungen ändern, einschließlich des Vorbereitungsmodus für die Zieltabelle, ob LOB-Spalten in die Replikation einbezogen werden sollen, die maximale LOB-Größe, die Überprüfung und die Protokollierung. Weitere Informationen finden Sie unter Task settings (Aufgabeneinstellungen) .	ReplicationState ist CREATED, STOPPED oder FAILED.
Tabellenzuordnung	Sie können die Tabellenzuordnungen für eine Serverless-Replikation ändern, einschließlich der Auswahlregeln und der Transformationsregeln. Weitere Informationen finden Sie unter Table mapping (Tabellenzuweisung) .	ReplicationState ist CREATED, STOPPED oder FAILED.

Einstellung	Beschreibung	Zulässige Zustände
Compute-Konfiguration	<p>Sie können die Compute-Konfigurationseinstellungen für eine Serverlose-Replikation ändern, einschließlich der Netzwerkeinstellungen, Skalierungseinstellungen und Wartungseinstellungen. Informationen zu den Compute-Config-Einstellungen finden Sie unter Compute-Konfiguration.</p>	<ul style="list-style-type: none"> • Sie können die folgenden Skalierungs-, Wartungs- und Netzwerkeinstellungen ändern, wenn der <code>ReplicationState</code> <code>CREATED</code>, <code>STOPPED</code> oder <code>FAILED</code> ist: <ul style="list-style-type: none"> • <code>MinCapacityUnits</code> • <code>MaxCapacityUnits</code> • <code>MultiAZ</code> • <code>PreferredMaintenanceWindow</code> • <code>VpcSecurityGroupIds</code> • Sie können die folgenden Netzwerk- und Sicherheitseinstellungen ändern, wenn der <code>ReplicationState</code> <code>CREATED</code> oder <code>FAILED</code> bzw. wenn der <code>ProvisionState</code> <code>null</code> ist: <ul style="list-style-type: none"> • <code>AvailabilityZone</code> • <code>DnsNameServers</code> • <code>KmsKeyId</code>

Einstellung	Beschreibung	Zulässige Zustände
		<ul style="list-style-type: none"> • <code>ReplicationSubnetGroup</code>

Compute-Konfiguration

Sie konfigurieren Ihre Replikationsbereitstellung mithilfe des Parameters „Compute-Konfiguration“ oder des Konsolenabschnitts. Die Felder im „Compute-Konfiguration“-Objekt umfassen Folgendes:

Option	Beschreibung
MinCapacityUnits	Dies ist die Mindestanzahl von DMS-Kapazitätseinheiten (DCU), die bereitgestellt AWS DMS werden. Dies ist auch die minimale DCU, auf die Autoscaling herunterskaliert werden könnte.
MaxCapacityUnits	Dies ist die maximale Anzahl von DMS-Kapazitätseinheiten (DCU), die AWS DMS je nach der Kapazität prognose Ihrer Replikation bereitstellen kann. Dies ist auch die maximale DCU, auf die Autoscaling hochskaliert werden könnte.
KmsKeyId	Der Verschlüsselungsschlüssel, der zum Verschlüsseln der Replikationsspeicher- und Verbindungsinformationen verwendet werden soll. Wenn Sie (Standard) <code>aws/dms</code> wählen, AWS DMS verwendet den Standard-KMS-Schlüssel, der Ihrem Konto zugeordnet ist, und. AWS-Region Eine Beschreibung und die Kontonummer werden angezeigt, zusammen mit den Schlüssel-ARN. Weitere Informationen zur Verwendung des Verschlüsselungsschlüssels finden Sie unter Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen . Lassen Sie für dieses Tutorial die Option (Standard) <code>aws/dms</code> ausgewählt.

Option	Beschreibung
ReplicationSubnetGroupIds	Wählen Sie die Replikationssubnetzgruppe in der von Ihnen ausgewählten VPC aus, in der die Replikation erstellt werden soll. Wenn sich Ihre Quelldatenbank in einer VPC befindet, wählen Sie die Subnetzgruppe, die die Quelldatenbank enthält, als Speicherort für Ihre Replikation aus. Weitere Informationen zu Replikationssubnetzgruppen finden Sie unter Erstellen einer Replikationssubnetzgruppe .
VpcSecurityGroupIds	Die Replikations-Instance wird in einer VPC erstellt. Wenn sich Ihre Quelldatenbank in einer VPC befindet, wählen Sie die VPC-Sicherheitsgruppe aus, die Zugriff auf die DB-Instance bietet, in der sich die Datenbank befindet.
PreferredMaintenanceWindow	Dieser Parameter definiert einen wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time). Die Standardinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem Zeitblock von jeweils 8 Stunden ausgewählt wird AWS-Region und an einem zufälligen Wochentag stattfindet.
MultiAZ	Verwenden Sie diesen optionalen Parameter zum Erstellen eines Standby-Replikats Ihrer Replikation in einer anderen Availability Zone, um Failover-Unterstützung zu bieten. Wenn Sie Change Data Capture (CDC) oder die fortlaufende Replikation verwenden möchten, sollten Sie diese Option aktivieren.

Grundlegendes zur automatischen Skalierung im serverlosen Modus AWS DMS

Nachdem Sie eine Replikation bereitgestellt haben und sie sich in diesem RUNNING Status befindet, verwaltet der AWS DMS Service die Kapazität der zugrunde liegenden Ressourcen, um sie an sich ändernde Workloads anzupassen. Dieses Management skaliert die Replikationsressourcen auf der Grundlage der folgenden Replikationseinstellungen:

- `MinCapacityUnits`
- `MaxCapacityUnits`

Replikationen werden nach einem Zeitraum, in dem ein oberer Nutzungsschwellenwert überschritten wird, hochskaliert, und wenn die Kapazitätsauslastung über einen längeren Zeitraum unter einem Mindestschwellenwert für die Kapazitätsnutzung liegt, herunterskaliert.

Optimierung der automatischen Skalierung im serverlosen Modus AWS DMS

Um die Parameter für die automatische Skalierung der Replikation `MaxCapacityUnits` zu optimieren, empfehlen wir, den Maximalwert einzustellen und die Bereitstellung der Ressourcen AWS DMS verwalten zu lassen. Es wird empfohlen, die größte Einstellung für die maximale DCU-Kapazität zu wählen, um den größtmöglichen Nutzen aus Auto Scaling zu ziehen und Spitzen im Transaktionsvolumen auszugleichen. Der Preisrechner zeigt die maximalen monatlichen Kosten an, wenn Ihre Replikation kontinuierlich die maximale DCU verwendet. Die maximale DCU entspricht nicht den tatsächlichen Kosten, da Sie nur für die genutzte Kapazität zahlen.

Wenn Ihre Replikation die Ressourcen nicht mit voller Kapazität nutzt, AWS DMS werden die Ressourcen schrittweise aufgehoben, um Ihnen Kosten zu sparen. Da die Bereitstellung und die Aufhebung der Bereitstellung von Ressourcen jedoch einige Zeit in Anspruch nimmt, empfehlen wir Ihnen, Ihre `MinCapacityUnits`-Einstellung auf einen Wert festzulegen, mit dem alle zu erwartenden plötzlichen Spitzen Ihres Replikations-Workloads bewältigt werden können. Dadurch wird verhindert, dass Ihre Replikation zu wenig bereitgestellt wird, und gleichzeitig AWS DMS werden Ressourcen für die höhere Workload-Ebene bereitgestellt.

Wenn Sie Ihre Replikation mit einer Einstellung für die maximale Kapazität bereitstellen, die für die Datenanforderungen zu niedrig ist, um plötzliche Spitzen im Replikations-Workload bewältigen zu können, wird die `CapacityUtilization`-Metrik möglicherweise konsistent auf ihrem Maximalwert angezeigt. Dies kann dazu führen, dass Ihre Replikation fehlschlägt. Wenn Ihre Replikation aufgrund unzureichend bereitgestellter Ressourcen fehlschlägt, AWS DMS wird ein out-of-memory Ereignis in

Ihren Replikationsprotokollen erzeugt. Wenn der Fehler out-of-memory auf einen plötzlichen Anstieg Ihrer Replikationslast zurückzuführen ist, wird die Replikation automatisch skaliert und neu gestartet.

Überwachung serverloser Replikationen AWS DMS

AWS bietet verschiedene Tools zur Überwachung Ihrer AWS DMS serverlosen Replikationen und zur Reaktion auf potenzielle Vorfälle:

- [AWS DMS Metriken für serverlose Replikation](#)
- [AWS DMS Serverlose Replikationsprotokolle](#)

AWS DMS Metriken für serverlose Replikation

Die Überwachung der serverlosen Replikation umfasst CloudWatch Amazon-Metriken für die folgenden Statistiken. Diese Statistiken sind nach den einzelnen Serverless-Replikationen gruppiert.

Metrik	Einheiten	Beschreibung
CapacityUtilization	Prozent	Prozentsatz des insgesamt von der Serverless-Replikation verwendeten Speichers
CDC IncomingChanges	Prozent	Die Gesamtzahl der Änderungsereignisse bei a point-in-time , die darauf warten, auf das Ziel angewendet zu werden. Beachten Sie, dass dies nicht dasselbe ist wie eine Messung der Transaktionsänderungsrate am Quellendpunkt. Eine große Zahl für diese Metrik weist AWS DMS normalerweise darauf hin, dass die erfassten Änderungen nicht rechtzeitig angewendet werden können, was zu einer hohen Ziellatenz führt.
CDC LatencySource	Sekunden	Die Anzahl der Sekunden zwischen dem zuletzt erfassten Ereignis vom Quellendpunkt und dem aktuellen Systemzeitstempel der AWS DMS -Instance. CDC LatencySource steht für die Latenz zwischen Quelle und Replikationsinstanz. Ein hoher CDC LatencySource bedeutet, dass der Prozess der Erfassung von Änderungen aus der Quelle verzögert wird. Um die Latenz bei einer laufenden Replikation zu

Metrik	Einheiten	Beschreibung
		<p>ermitteln, können Sie sich diese Metrik zusammen mit LatencyTarget CDC ansehen. Wenn sowohl CDC als auch CDC LatencySource hoch LatencyTarget sind, untersuchen Sie zuerst CDC. LatencySource</p> <p>CDC LatencySource kann 0 sein, wenn es keine Replikationsverzögerung zwischen der Quelle und der Replikation gibt. CDC LatencySource kann auch Null werden, wenn die Replikation versucht, das nächste Ereignis im Transaktionslog der Quelle zu lesen, und es im Vergleich zum letzten Lesen aus der Quelle keine neuen Ereignisse gibt. In diesem Fall setzt die Replikation den CDC LatencySource auf 0 zurück.</p>

Metrik	Einheiten	Beschreibung
CDC LatencyTarget	Sekunden	<p>Die Anzahl der Sekunden zwischen dem ersten Ereigniszeitstempel, für den auf dem Ziel ein Commit ausgeführt werden soll, und dem aktuellen Zeitstempel der AWS DMS -Instance. Die Ziellatenz ist der Unterschied zwischen der Serverzeit der Replikations-Instance und der ältesten unbestätigten Ereignis-ID, die an eine Zielkomponente weitergeleitet wurde. Mit anderen Worten, die Ziellatenz ist der Zeitstempelunterschied zwischen der Replikations-Instance und dem ältesten Ereignis, das angewendet, aber vom TRG-Endpunkt nicht bestätigt wurde (99 %). Wenn der CDC hoch LatencyTarget ist, bedeutet dies, dass der Prozess der Anwendung von Änderungsereignissen auf das Ziel verzögert ist. Um die Latenz bei einer laufenden Replikation zu ermitteln, können Sie sich diese Metrik zusammen mit LatencySource CDC ansehen. Wenn CDC hoch, aber CDC LatencySource nicht hoch LatencyTarget ist, untersuchen Sie, ob:</p> <ul style="list-style-type: none"> • Das Ziel enthält keine Primärschlüssel oder Indizes • In der Ziel- oder Replikations-Instance treten Ressourcenengpässe auf • Netzwerkprobleme treten zwischen der Replikation und dem Ziel auf
CDC ThroughputBandwidthTarget	KB pro Sekunde	<p>Für das Ziel übertragene ausgehende Daten in KB pro Sekunde. CDC ThroughputBandwidth zeichnet ausgehende Daten auf, die an Probenahmestellen übertragen werden. Wenn kein Netzwerkverkehr gefunden wird, ist der Wert null. Da CDC keine Langläufer-Transaktionen ausgibt, wird der Netzwerkdatenverkehr u. U. nicht aufgezeichnet.</p>
CDC ThroughputRowsSource	Zeilen pro Sekunde	<p>Eingehende Änderungen von der Quelle in Zeilen pro Sekunde.</p>

Metrik	Einheiten	Beschreibung
CDC ThroughputRowsTarget	Zeilen pro Sekunde	Ausgehende Änderungen für das Ziel in Zeilen pro Sekunde.
FullLoadThroughputBandwidthTarget	KB pro Sekunde	Für das Ziel übertragene ausgehende Daten von Volllast in KB pro Sekunde.
FullLoadThroughputRowsTarget	Zeilen pro Sekunde	Ausgehende Änderungen von einem vollständigen Ladevorgang für das Ziel in Zeilen pro Sekunde

AWS DMS Serverlose Replikationsprotokolle

Sie können Amazon verwenden CloudWatch , um Replikationsinformationen während eines AWS DMS Migrationsprozesses zu protokollieren. Sie aktivieren die Protokollierung während der Auswahl der Replikationseinstellungen.

Serverlose Replikationen laden Statusprotokolle in Ihr CloudWatch Konto hoch, um einen besseren Überblick über den Fortschritt der Replikation zu erhalten und Sie bei der Fehlerbehebung zu unterstützen.

AWS DMS lädt serverlos verknüpfte Protokolle mit dem Präfix in eine spezielle Protokollgruppe hoch. `dms-serverless-replication-<your replication config resource ID>` Innerhalb dieser Protokollgruppe gibt es einen Protokollstream mit dem Namen `dms-serverless-replication-orchestrator-<your replication config resource ID>`. In diesem Protokollstream wird der Replikationsstatus Ihrer Replikation sowie eine zugehörige Meldung mit weiteren Einzelheiten zu der Arbeit, die in dieser Phase ausgeführt wird, gemeldet. Beispiele für Protokolleinträge finden Sie nachfolgend in [Beispiele für Serverless-Replikationsprotokolle](#).

Note

AWS DMS erstellt weder die Protokollgruppe noch den Stream, bis Sie die Replikation ausführen. AWS DMS erstellt die Protokollgruppe oder den Protokollstream nicht, wenn Sie nur die Replikation erstellen.

Gehen Sie wie folgt vor, um Protokolle einer Replikation anzuzeigen, die ausgeführt wurde:

1. Öffnen Sie die AWS DMS Konsole und wählen Sie im Navigationsbereich Serverlose Replikationen aus. Das Dialogfeld Serverless-Replikationen wird angezeigt.
2. Gehen Sie zum Abschnitt Konfiguration und wählen Sie in der Spalte „Allgemein“ die Option Serverless-Protokolle anzeigen aus. Die CloudWatch Protokollgruppe wird geöffnet.
3. Suchen Sie den Abschnitt „Protokolle der Migrationsaufgaben“ und wählen Sie „ CloudWatch Protokolle anzeigen“.

Wenn Ihre Replikation fehlschlägt, AWS DMS wird ein Protokolleintrag mit dem Replikationsstatus und einer Meldung erstellt `failed`, in der der Grund für den Fehler beschrieben wird. Als ersten Schritt zur Behebung einer fehlgeschlagenen Replikation sollten Sie Ihre CloudWatch Protokolle überprüfen.

Note

Wie bei AWS DMS Classic haben Sie die Möglichkeit, eine detailliertere Protokollierung des Fortschritts der Datenmigration selbst zu aktivieren, d. h. der von der zugrunde liegenden Replikationsaufgabe ausgegebenen Protokolle. Sie können diese Protokolle in Ihren Replikationseinstellungen aktivieren, indem Sie `EnableLogging` im `Logging`-Feld auf `true` setzen, wie im folgenden JSON-Beispiel:

```
{
  "Logging": {
    "EnableLogging": true
  }
}
```

Wenn Sie diese Protokolle aktivieren, werden sie erst in der `running`-Phase Ihrer Serverless- Replikation angezeigt. Sie werden in derselben Protokollgruppe wie der vorherige Protokollstream angezeigt, befinden sich jedoch unter dem neuen Protokollstream `dms-serverless-serv-res-id-{unique identifier}`. Im folgenden Abschnitt finden Sie Informationen zur Interpretation von Serverless-Replikationsprotokollen.

Beispiele für Serverless-Replikationsprotokolle

Dieser Abschnitt enthält Beispiele für Protokolleinträge für Serverless-Replikationen.

Beispiel: Start der Replikation

Wenn Sie eine serverlose Replikation ausführen, AWS DMS wird ein Protokolleintrag erstellt, der dem folgenden ähnelt:

```
{'replication_state':'initializing', 'message': 'Initializing the replication workflow.'}
```

Beispiel: Replikationsfehler

Wenn einer der Endpunkte der Replikation nicht richtig konfiguriert ist, wird ein Protokolleintrag AWS DMS erstellt, der dem folgenden ähnelt:

```
{'replication_state':'failed', 'message': 'Test connection failed for endpoint X.', 'failure_message': 'X'}
```

Wenn Sie diese Meldung nach einem Fehler in Ihrem Protokoll sehen, stellen Sie sicher, dass der angegebene Endpunkt fehlerfrei und korrekt konfiguriert ist.

AWS DMS Einschränkungen ohne Server

AWS DMS Serverless hat die folgenden Einschränkungen:

- Sie können nur eine AWS DMS Replikationskonfiguration ändern, die sich im `CREATED` Status `STOPPED`, oder `FAILED` befindet. Einzelheiten dazu, welche Einstellungen Sie unter welchen Bedingungen ändern können, finden Sie unter [AWS DMS Serverlose Replikationen ändern](#).
- Sie können nur eine AWS DMS Replikationskonfiguration löschen `STOPPED`, die sich im `FAILED` Status oder befindet.
- Für eine Replikation steht ein statischer zugewiesener Speicher von 100 GB zur Verfügung. Wenn Ihre Replikation aufgrund von Anforderungen wie etwa lang andauernden Transaktionen oder Caching mehr Speicher beansprucht, empfehlen wir Ihnen, Ihren Workload in separate Serverless-Replikationen zu partitionieren. Sie können Ihren Workload tabellarisch oder nach Anforderung partitionieren, indem Sie beispielsweise die gesamte Replikation, an der LOBs beteiligt sind, in einer separaten Serverless-Replikation zusammenfassen.
- Im Gegensatz zu Replikationsinstanzen haben AWS DMS serverlose Replizierungen keine öffentliche IP-Adresse für Verwaltungsaufgaben. Sie verwalten Serverless-Replikationen über die Konsole.

- Diese Version von AWS DMS Serverless unterstützt nicht alle Quell- und Zielendpunkttypen, AWS DMS die der Standard unterstützt. Eine Liste der unterstützten Engine-Typen finden Sie unter [AWS DMS Serverlose Komponenten](#).
- Serverless-Replikationen müssen mithilfe von VPC-Endpunkten auf Abhängigkeiten zugreifen. Sie müssen VPC-Endpunkte verwenden, um auf die folgenden Endpunkttypen zuzugreifen:
 - Amazon Amazon S3
 - Amazon Kinesis
 - AWS Secrets Manager
 - Amazon-DynamoDB
 - Amazon-Redshift
 - OpenSearch Amazon-Dienst

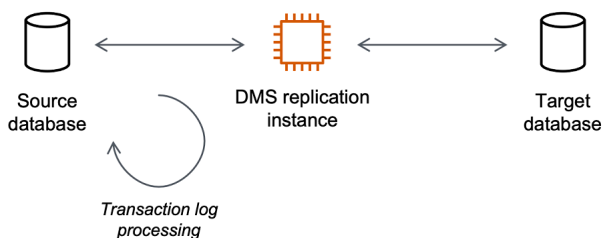
Informationen zum Einrichten von VPC-Endpunkten finden Sie unter [Konfigurieren von VPC-Endpunkten als Quell- und Zielendpunkte in AWS DMS](#).

- AWS DMS Serverless unterstützt keine Ansichten mit Auswahl- und Transformationsregeln.
- AWS DMS Serverless unterstützt die Verwendung von vom AWS Kunden verwalteten Schlüsseln nicht. AWS DMS Serverless unterstützt nur die Verwendung des Standard-DMS-Schlüssels. Weitere Informationen finden Sie unter [Datenschutz in AWS Database Migration Service](#).
- DMS Serverless unterstützt keine SSL-Verbindungen für DB2-Endpunkte.

Mit einer AWS DMS Replikationsinstanz arbeiten

Wenn Sie eine AWS DMS Replikationsinstanz erstellen, AWS DMS erstellt sie auf einer Amazon EC2 EC2-Instance in einer Virtual Private Cloud (VPC), die auf dem Amazon VPC-Service basiert. Sie verwenden diese Replikations-Instance für die Durchführung Ihrer Datenbankmigration. Die Replikations-Instance bietet hohe Verfügbarkeit und Failover-Support über eine Multi-AZ-Bereitstellung bei Auswahl der Option Multi-AZ.

In einer Multi-AZ-Bereitstellung AWS DMS wird automatisch eine synchrone Standby-Replik der Replikationsinstanz in einer anderen Availability Zone bereitgestellt und verwaltet. Die primäre Replikations-Instance wird über die Availability Zones synchron auf das Standby-Replikat repliziert. Dieser Ansatz sorgt für Datenredundanz, vermeidet das Einfrieren von E/A-Vorgängen und minimiert Latenzspitzen.



AWS DMS verwendet eine Replikationsinstanz, um eine Verbindung zu Ihrem Quelldatenspeicher herzustellen, die Quelldaten zu lesen und die Daten für den Zieldatenspeicher zu formatieren. Eine Replikations-Instance lädt zudem die Daten in den Ziel-Datenspeicher. Der Großteil dieser Verarbeitung erfolgt im Arbeitsspeicher. Allerdings werden umfangreiche Transaktionen ggf. auf Festplatte gepuffert. Zwischengespeicherte Transaktionen und Protokolldateien werden ebenfalls auf Festplatte geschrieben.

Sie können eine AWS DMS Replikationsinstanz in den folgenden AWS Regionen erstellen.

Name der Region	Region	Endpoint	Protocol (Protokol I)
USA Ost (Ohio)	us-east-2	dms.us-east-2.amazonaws.com	HTTPS
		dms-fips.us-east-2.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Nord-Virginia)	us-east-1	dms.us-east-1.amazonaws.com	HTTPS
		dms-fips.us-east-1.amazonaws.com	HTTPS
USA West (Nordkalifornien)	us-west-1	dms.us-west-1.amazonaws.com	HTTPS
		dms-fips.us-west-1.amazonaws.com	HTTPS
USA West (Oregon)	us-west-2	dms.us-west-2.amazonaws.com	HTTPS
		dms-fips.us-west-2.amazonaws.com	HTTPS
Afrika (Kapstadt)	af-south-1	dms.af-south-1.amazonaws.com	HTTPS
Asien-Pazifik (Hongkong)	ap-east-1	dms.ap-east-1.amazonaws.com	HTTPS
Asien-Pazifik (Hyderabad)	ap-south-2	dms.ap-south-2.amazonaws.com	HTTPS
Asien-Pazifik (Jakarta)	ap-southeast-3	dms.ap-southeast-3.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokol I)
Asien-Pazifik (Melbourne)	ap-southeast-4	dms.ap-southeast-4.amazonaws.com	HTTPS
Asien-Pazifik (Mumbai)	ap-south-1	dms.ap-south-1.amazonaws.com	HTTPS
Asien-Pazifik (Osaka)	ap-northeast-3	dms.ap-northeast-3.amazonaws.com	HTTPS
Asien-Pazifik (Seoul)	ap-northeast-2	dms.ap-northeast-2.amazonaws.com	HTTPS
Asien-Pazifik (Singapur)	ap-southeast-1	dms.ap-southeast-1.amazonaws.com	HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	dms.ap-southeast-2.amazonaws.com	HTTPS
Asien-Pazifik (Tokio)	ap-northeast-1	dms.ap-northeast-1.amazonaws.com	HTTPS
Kanada (Zentral)	ca-central-1	dms.ca-central-1.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Kanada West (Calgary)	ca-west-1	dms.ca-west-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	dms.eu-central-1.amazonaws.com	HTTPS
Europa (Irland)	eu-west-1	dms.eu-west-1.amazonaws.com	HTTPS
Europa (London)	eu-west-2	dms.eu-west-2.amazonaws.com	HTTPS
Europa (Mailand)	eu-south-1	dms.eu-south-1.amazonaws.com	HTTPS
Europa (Paris)	eu-west-3	dms.eu-west-3.amazonaws.com	HTTPS
Europa (Spanien)	eu-south-2	dms.eu-south-2.amazonaws.com	HTTPS
Europa (Stockholm)	eu-north-1	dms.eu-north-1.amazonaws.com	HTTPS
Europa (Zürich)	eu-central-2	dms.eu-central-2.amazonaws.com	HTTPS
Israel (Tel Aviv)	il-central-1	dms.il-central-1.amazonaws.com	HTTPS

Name der Region	Region	Endpoint	Protocol (Protokoll)
Naher Osten (Bahrain)	me-south-1	dms.me-south-1.amazonaws.com	HTTPS
Naher Osten (VAE)	me-central-1	dms.me-central-1.amazonaws.com	HTTPS
Südamerika (São Paulo)	sa-east-1	dms.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-Ost)	us-gov-east-1	dms.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-West)	us-gov-west-1	dms.us-gov-west-1.amazonaws.com	HTTPS

AWS DMS unterstützt eine spezielle AWS Region namens „Region“ AWS GovCloud (US) , die es US-Regierungsbehörden und Kunden ermöglicht, sensible Workloads in die Cloud zu verlagern. AWS GovCloud (US) geht auf die spezifischen regulatorischen und Compliance-Anforderungen der US-Regierung ein. Weitere Informationen zu AWS GovCloud (US) finden Sie unter [Was ist AWS GovCloud \(USA\)?](#)

In den folgenden Abschnitten erhalten Sie weitere Informationen zu Replikations-Instances.

Themen

- [Auswahl der richtigen AWS DMS-Replikationsinstanz für Ihre Migration](#)
- [Auswahl der besten Größe für eine Replikations-Instance](#)
- [Arbeiten mit Versionen der Replikations-Engine](#)

- [Öffentliche und private Replikations-Instances](#)
- [IP-Adressierung und Netzwerktypen](#)
- [Einrichten eines Netzwerks für eine Replikations-Instance](#)
- [Festlegen eines Verschlüsselungsschlüssels für eine Replikations-Instance](#)
- [Erstellen einer Replikations-Instance](#)
- [Ändern einer Replikations-Instance](#)
- [Neustarten einer Replikations-Instance.](#)
- [Löschen einer Replikations-Instance](#)
- [Arbeiten mit dem AWS DMS -Wartungszeitfenster](#)

Auswahl der richtigen AWS DMS-Replikationsinstanz für Ihre Migration

AWS DMS erstellt die Replikationsinstanz auf einer Amazon EC2 EC2-Instance. AWS DMS unterstützt derzeit die Amazon EC2 EC2-Instance-Klassen T2, T3, C4, C5, C6i, R4, R5 und R6i für Replikationsinstanzen:

- T2-Instances sind Burstable Performance Instances, die eine grundlegende CPU-Leistung mit der Fähigkeit bieten, darüber hinauszugehen. Die Basisleistung und die Steigerbarkeit unterliegen dem CPU-Guthaben. T2-Instances erhalten kontinuierlich CPU-Guthaben mit einer festgelegten Rate, die von der Instance-Größe abhängt. Sie akkumulieren CPU-Guthaben, wenn sie inaktiv sind, und verbrauchen CPU-Guthaben, wenn sie aktiv sind.

T2-Instances sind eine gute Wahl für eine Vielzahl von Allzweck-Workloads. Zu den Beispielen gehören Microservices, interaktive Anwendungen mit geringer Latenz, kleine und mittelgroße Datenbanken, virtuelle Desktops, Entwicklungs-, Aufbau- und Staging-Umgebungen, Code-Repositorys und Produktprototypen.

- T3-Instances sind der Burstable-Allzweck-Instance-Typ der nächsten Generation. Dieser Typ bietet eine grundlegende CPU-Leistung mit der Fähigkeit, die CPU-Nutzung jederzeit so lange wie nötig zu steigern. T3-Instances bieten ein ausgewogenes Verhältnis von Computing-, Speicher- und Netzwerkressourcen und sind ideal für Workloads mit moderater CPU-Nutzung, bei der temporäre Nutzungsspitzen auftreten. T3-Instances akkumulieren CPU-Guthaben, wenn ein Workload unter dem Basisschwellenwert arbeitet. Jedes CPU-Guthaben bietet der T3-Instance die Möglichkeit, bei Bedarf eine Minute lang die Leistung eines vollen CPU-Kerns zu nutzen.

T3-Instances können jederzeit so lange wie möglich im unlimited-Modus erweitert werden.

Weitere Informationen zum unlimited-Modus finden Sie unter [Arbeiten mit dem unbegrenzten Modus für Burstable Performance Instances](#).

- C4-Instances sind für rechenintensive Workloads optimiert und bieten eine kostengünstige hohe Leistung bei günstigem Preis-Leistungs-Verhältnis. Sie erzielen eine deutlich höhere PPS-Leistung (Packet per Second), einen geringeren Netzwerk-Jitter und eine geringere Netzwerklatenz. AWS DMS kann auch CPU-intensiv sein, insbesondere bei heterogenen Migrationen und Replikationen wie der Migration von Oracle zu PostgreSQL. C4-Instances können eine gute Wahl für diese Situationen sein.
- C5-Instances sind der Instance-Typ der nächsten Generation, der für die Ausführung anspruchsvoller rechenintensiver Workloads kostengünstige Höchstleistung zu einem niedrigen Preis pro Rechenleistung bietet. Dazu gehören Workloads wie Hochleistungs-Webserver, High Performance Computing (HPC), Stapelverarbeitung, Ad-Serving, hoch skalierbare Multiplayer-Spiele und Videokodierung. Andere Workloads, für die sich C5-Instances eignen, sind etwa wissenschaftliche Modellierung, verteilte Analysen sowie Inferenz für Machine und Deep Learning. Die C5-Instances sind mit einer Auswahl an Prozessoren von Intel und AMD erhältlich.
- C6i-Instances bieten ein um bis zu 15 % besseres Computing-Preis-Leistungs-Verhältnis als vergleichbare Gen5-Instances für eine Vielzahl von Workloads sowie eine Always-On-Speicherverschlüsselung. C6i-Instances sind ideal geeignet für rechenintensive Workloads, z. B. Stapelverarbeitung, verteilte Analysen, High Performance Computing (HPC), Ad-Serving, hoch skalierbare Multiplayer-Spiele und Videokodierung.
- R4-Instance-Klassen sind speicheroptimiert für speicherintensive Workloads. Laufende Migrationen oder Replikationen von Transaktionssystemen mit hohem Durchsatz unter Verwendung von AWS DMS können zudem große Mengen an CPU und Speicher verbrauchen. R4-Instances bieten mehr Speicher pro vCPU als Instance-Typen früherer Generationen.
- R5-Instances sind die nächste Generation arbeitsspeicheroptimierter Instance-Typen für Amazon EC2. R5-Instances sind gut geeignet für speicherintensive Anwendungen, z. B. Hochleistungsdatenbanken, verteilte Web-Scale-In-Memory-Caches, mittelgroße In-Memory-Datenbanken, Big-Data-Analytik in Echtzeit und andere Unternehmensanwendungen. Laufende Migrationen oder Replikationen von Transaktionssystemen mit hohem Durchsatz können ebenfalls große Mengen an CPU und Arbeitsspeicher beanspruchen. AWS DMS
- R6i-Instances bieten ein um bis zu 15 % besseres Computing-Preis-Leistungs-Verhältnis als vergleichbare Gen5-Instances für eine Vielzahl von Workloads sowie eine Always-On-Speicherverschlüsselung. R6i-Instances sind SAP-zertifiziert und eignen sich ideal für Workloads wie SQL- und NoSQL-Datenbanken, verteilte Web-Scale-In-Memory-Caches wie Memcached und

Redis, In-Memory-Datenbanken wie SAP HANA und Big-Data-Analytik in Echtzeit wie Hadoop- und Spark-Cluster.

Jede Replikations-Instance verfügt über eine spezifische Konfiguration für Arbeitsspeicher und vCPU. Die folgende Tabelle zeigt die Konfiguration für die einzelnen Replikations-Instance-Typen. Informationen zu den Preisen finden Sie auf der Seite [AWS Database Migration Service -Service – Preise](#).

Instance-Typen für allgemeine Replikationszwecke

Typ	vCPU	Arbeitsspeicher (GiB)
dms.t2.micro	1	1
dms.t2.small	1	2
dms.t2.medium	2	4
dms.t2.large	2	8
dms.t3.micro	2	1
dms.t3.small	2	2
dms.t3.medium	2	4
dms.t3.large	2	8

Computing-optimierte Replikations-Instance-Typen

Typ	vCPU	Arbeitsspeicher (GiB)
dms.c4.large	2	3,75
dms.c4.xlarge	4	7,5
dms.c4.2xlarge	8	15
dms.c4.4xlarge	16	30

Typ	vCPU	Arbeitsspeicher (GiB)
dms.c5.large	2	4
dms.c5.xlarge	4	8
dms.c5.2xlarge	8	16
dms.c5.4xlarge	16	32
dms.c5.9xlarge	36	72
dms.c5.12xlarge	48	96
dms.c5.18xlarge	72	144
dms.c5.24xlarge	96	192
dms.c6i.large	2	4
dms.c6i.xlarge	4	8
dms.c6i.2xlarge	8	16
dms.c6i.4xlarge	16	32
dms.c6i.8xlarge	32	64
dms.c6i.12xlarge	48	96
dms.c6i.16xlarge	64	128
dms.c6i.24xlarge	96	192
dms.c6i.32xlarge	128	256

Speicher-optimierte Replikations-Instance-Typen

Typ	vCPU	Arbeitsspeicher (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
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

Typ	vCPU	Arbeitsspeicher (GiB)
dms.r6i.24xlarge	96	768
dms.r6i.32xlarge	128	1024

In den obigen Tabellen sind alle Typen von AWS DMS Replikationsinstanzen aufgeführt, aber die Typen, die in Ihrer Region verfügbar sind, können variieren. Um die in Ihrer Region verfügbaren Typen von Replikations-Instances zu sehen, können Sie den folgenden [AWS CLI](#)-Befehl ausführen:

```
aws dms describe-orderable-replication-instances --region your_region_name
```

Themen

- [Entscheidung, welche Instance-Klasse verwendet werden soll](#)
- [Arbeiten mit dem unbegrenzten Modus für Burstable Performance Instances](#)

Entscheidung, welche Instance-Klasse verwendet werden soll

Um herauszufinden, welche Replikationsinstanzklasse für Sie am besten geeignet ist, schauen wir uns den AWS DMS verwendeten Change Data Capture-Prozess (CDC) an.

Angenommen, Sie führen einen vollständigen Ladevorgang plus CDC-Aufgabe (Massenladevorgang plus laufende Replikation) aus. In diesem Fall hat die Aufgabe ein eigenes SQLite-Repository, um Metadaten und andere Informationen zu speichern. Bevor ein Vollladevorgang AWS DMS gestartet wird, werden die folgenden Schritte ausgeführt:

- AWS DMS beginnt mit der Erfassung von Änderungen für die Tabellen, die migriert werden, aus dem Transaktionslog der Quell-Engine (wir nennen diese Änderungen zwischengespeichert). Nachdem das vollständige Laden abgeschlossen ist, werden diese zwischengespeicherten Änderungen gesammelt und auf das Ziel angewendet. Abhängig von der Menge der zwischengespeicherten Änderungen können diese Änderungen direkt aus dem Arbeitsspeicher angewendet werden, wo sie zuerst bis zu einem festen Schwellenwert gesammelt werden. Alternativ können sie auch von der Festplatte aus angewendet werden, wohin Änderungen geschrieben werden, wenn sie nicht im Arbeitsspeicher gehalten werden können.
- Nachdem die zwischengespeicherten Änderungen übernommen wurden, wird standardmäßig ein transaktionaler Apply-Prozess auf der Zielinstantz AWS DMS gestartet.

AWS DMS verwendet während der Phase der angewendeten zwischengespeicherten Änderungen und der Phase der laufenden Replikationen zwei Stream-Puffer, jeweils einen für eingehende und ausgehende Daten. AWS DMS verwendet außerdem eine wichtige Komponente, den sogenannten Sorter, bei dem es sich um einen weiteren Speicherpuffer handelt. Im Folgenden sind zwei wichtige Anwendungen der Sorter-Komponente beschrieben (es gibt noch andere):

- Er verfolgt alle Transaktionen und stellt sicher, dass er nur relevante Transaktionen in den Ausgangspuffer weiterleitet.
- Es stellt sicher, dass Transaktionen in der gleichen Commit-Reihenfolge wie auf der Quelle weitergeleitet werden.

Wie Sie sehen, haben wir drei wichtige Speicherpuffer in dieser Architektur für CDC in AWS DMS. Wenn einer dieser Puffer unter Speicherdruck steht, kann es bei der Migration zu Leistungsproblemen kommen, die zu Ausfällen führen können.

Wenn Sie hohe Workloads mit einer hohen Anzahl von Transaktionen pro Sekunde (TPS) in diese Architektur einbinden, finden Sie möglicherweise den zusätzlichen Speicher nützlich, der von R5- und R6i-Instances zur Verfügung gestellt wird. Sie können R5- und R6i-Instances verwenden, um eine große Anzahl von Transaktionen im Speicher zu halten und Speicherdruckprobleme bei laufenden Replikationen zu vermeiden.

Arbeiten mit dem unbegrenzten Modus für Burstable Performance Instances

Eine als `unlimited` konfigurierte Burstable Performance Instance, wie etwa eine T3-Instance, kann eine hohe CPU-Auslastung je nach Bedarf in jedem erforderlichen Zeitraum aufrechterhalten. Der stündliche Preis für die Instance kann automatisch alle CPU-Auslastungsspitzen abdecken. Dies geschieht, wenn die durchschnittliche CPU-Nutzung der Instance in einem fortlaufendem 24-Stunden-Zeitraum oder über die Lebensdauer der Instance hinweg (es gilt der jeweils kürzere Zeitraum) bei oder unterhalb der Baseline liegt.

Für die meisten allgemeinen Workloads bieten als `unlimited` konfigurierte Instances ausreichend Leistung ohne zusätzliche Gebühren. Wenn die Instance für einen längeren Zeitraum mit einer höheren CPU-Nutzung ausgeführt wird, kann sie dies zu einer pauschalen Zusatzgebühr pro vCPU-Stunde tun. Informationen zu den Preisen für T3-Instances finden Sie unter „T3-CPU-Credits“ in [AWS Database Migration Service](#).

Weitere Informationen zum `unlimited`-Modus für T3-Instances finden Sie unter [Unbegrenzter Modus für Burstable Performance Instances](#) im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

Important

Wenn Sie eine `dms.t3.micro` Instance im Rahmen des [AWS Free Tier](#)-Angebots verwenden und sie im `unlimited`-Modus verwenden, können Gebühren anfallen. Insbesondere können Gebühren anfallen, wenn Ihre durchschnittliche Auslastung in einem fortlaufendem 24-Stunden-Zeitraum die Basisauslastung der Instance überschreitet. Weitere Informationen finden Sie unter [Baseline-Nutzung](#) im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

T3-Instances werden standardmäßig als `unlimited` gestartet. Wenn die durchschnittliche CPU-Auslastung über einen Zeitraum von 24 Stunden den Basiswert überschreitet, werden ebenso Gebühren für überschüssiges Guthaben anfallen. In einigen Fällen können Sie T3-Spot-Instances als `unlimited` starten und planen, sie sofort und für eine kurze Dauer zu verwenden. Wenn Sie dies ohne Leerlaufzeit für die Anrechnung von CPU-Guthaben tun, entstehen Gebühren für überschüssiges Guthaben. Wir empfehlen Ihnen, Ihre T3-Spot-Instances im Standardmodus zu starten, um höhere Kosten zu vermeiden. Weitere Informationen finden Sie unter [Überschüssige Guthaben können zu Gebühren führen](#), [T3-Spot-Instances](#) und [Standardmodus für Burstable Performance Instances](#) im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

Auswahl der besten Größe für eine Replikations-Instance

Die Wahl der geeigneten Replikations-Instance hängt von mehreren Faktoren ab, die sich auf Ihren Anwendungsfall beziehen. Um zu verstehen, wie die Replikations-Instance-Ressourcen verwendet werden, lesen Sie die folgende Diskussion. Sie deckt das übliche Szenario einer Volllast + CDC-Aufgabe ab.

AWS DMS lädt während einer Aufgabe zum vollständigen Laden die Tabellen einzeln. Standardmäßig werden acht Tabellen gleichzeitig geladen. AWS DMS erfasst laufende Änderungen an der Quelle während einer Aufgabe zum vollständigen Laden, sodass die Änderungen später auf dem Zielendpunkt angewendet werden können. Die Änderungen werden zwischengespeichert. Wenn der verfügbare Arbeitsspeicher erschöpft ist, werden die Änderungen auf der Festplatte zwischengespeichert. Wenn eine Vollladeaufgabe für eine Tabelle abgeschlossen ist, AWS DMS werden die zwischengespeicherten Änderungen sofort auf die Zieltabelle angewendet.

Nachdem alle ausstehenden zwischengespeicherten Änderungen für eine Tabelle übernommen wurden, befindet sich der Ziel-Endpunkt in einem transaktionskonsistenten Status. Zu diesem

Zeitpunkt ist das Ziel in Bezug auf die letzten zwischengespeicherten Änderungen mit dem Quellendpunkt synchron. AWS DMS beginnt dann mit der laufenden Replikation zwischen der Quelle und dem Ziel. Dazu werden die AWS DMS Änderungsvorgänge aus den Quelltransaktionsprotokollen übernommen und auf transaktionskonsistente Weise auf das Ziel angewendet. (Bei diesem Vorgang wird davon ausgegangen, dass die Option „Batch-optimiert anwenden“ nicht ausgewählt ist). AWS DMS streamt laufende Änderungen nach Möglichkeit durch den Speicher der Replikationsinstanz. Andernfalls werden Änderungen auf die Festplatte der Replikationsinstanz AWS DMS geschrieben, bis sie auf das Ziel angewendet werden können.

Sie haben eine gewisse Kontrolle darüber, wie die Replikations-Instance die Änderungsverarbeitung handhabt und wie der Arbeitsspeicher bei diesem Prozess verwendet wird. Weitere Informationen zur Optimierung der Änderungsverarbeitung finden Sie unter [Einstellungen für die Optimierung der Verarbeitung von Änderungen](#).

Zu berücksichtigende Faktoren

Arbeitsspeicher und Festplattenspeicher sind wichtige Faktoren bei der Auswahl einer geeigneten Replikations-Instance für Ihren Anwendungsfall. Im Folgenden finden Sie eine Beschreibung der Anwendungsfallmerkmale, die analysiert werden müssen, um eine Replikations-Instance auszuwählen.

- Datenbank- und Tabellengröße

Das Daten-Volumen hilft bei der Bestimmung der Aufgabenkonfiguration, um die Leistung bei Volllast zu optimieren. Beispielsweise können Sie für zwei 1-TB-Schemata Tabellen in vier Aufgaben mit 500 GB partitionieren und diese parallel ausführen. Die mögliche Parallelität hängt von der CPU-Ressource ab, die in der Replikations-Instance verfügbar ist. Aus diesem Grund ist es eine gute Idee, die Größe Ihrer Datenbank und Ihrer Tabellen zu kennen, um die Leistung bei Volllast zu optimieren. Dies hilft, die Anzahl der Aufgaben zu bestimmen, die Sie möglicherweise haben können.

- Große Objekte

Die Datentypen, die in Ihrem Migrationsbereich vorhanden sind, können sich auf die Leistung auswirken. Insbesondere große Objekte (LOBs) wirken sich auf die Leistung und den Speicherverbrauch aus. AWS DMS führt einen zweistufigen Prozess durch, um einen LOB-Wert zu migrieren. AWS DMS fügt zunächst die Zeile ohne den LOB-Wert in das Ziel ein. Zweitens aktualisiert AWS DMS die Zeile mit dem LOB-Wert. Dies hat Auswirkungen auf den

Speicher. Daher ist es wichtig, die LOB-Spalten an der Quelle zu identifizieren und ihre Größe zu analysieren.

- Ladefrequenz und Transaktionsgröße

Ladefrequenz und Transaktionen pro Sekunde (TPS) beeinflussen die Speichernutzung. Eine hohe Anzahl von TPS- oder DML-Aktivitäten (Data Manipulation Language) führt zu einer hohen Speicherauslastung. Dies geschieht, weil DMS die Änderungen zwischenspeichert, bis sie auf das Ziel angewendet werden. Während des CDC-Vorgangs führt dies zu Swapping (Schreiben auf die physische Festplatte aufgrund eines Speicherüberlaufs), was zu Latenz führt.

- Tabellenschlüssel und referenzielle Integrität

Informationen zu den Schlüsseln der Tabelle bestimmen den CDC-Modus (Batch Apply oder Transactional Apply), den Sie für die Datenmigration verwenden. Im Allgemeinen ist Transactional Apply langsamer als Batch Apply. Bei Transaktionen mit langer Laufzeit müssen möglicherweise viele Änderungen migriert werden. Wenn Sie transactional apply verwenden, ist AWS DMS möglicherweise mehr Speicher zum Speichern der Änderungen erforderlich als beim Batch-Anwenden. Wenn Sie Tabellen ohne Primärschlüssel migrieren, schlägt Batch Apply fehl und die DMS-Aufgabe wechselt in den Transactional-Apply-Modus. Wenn die referenzielle Integrität zwischen Tabellen während CDC aktiviert ist, AWS DMS verwendet standardmäßig transactional apply. Weitere Informationen zu Batch Apply im Vergleich zu Transactional Apply finden Sie unter [Wie kann ich das DMS-Batch-Apply-Feature verwenden, um die CDC-Replikationsleistung zu verbessern?](#).

Ermitteln Sie anhand dieser Metriken, ob die Replikations-Instance datenverarbeitungs- oder speicheroptimiert sein muss.

Häufige Probleme

Möglicherweise stoßen Sie während der Migration auf die folgenden häufigen Probleme, die zu Ressourcenkonflikten auf der Replikations-Instance führen. Weitere Informationen zu Metriken für Replikations-Instances finden Sie unter [Metriken zur Replikations-Instance](#).

- Wenn der Arbeitsspeicher in einer Replikations-Instance nicht mehr ausreicht, führt dies dazu, dass Daten auf die Festplatte geschrieben werden. Das Lesen von der Festplatte kann zu Latenz führen, die Sie vermeiden können, indem Sie die Größe der Replikations-Instance mit ausreichend Arbeitsspeicher ausstatten.

- Die der Replikations-Instance zugewiesene Festplattengröße kann kleiner als erforderlich sein. Die Festplattengröße wird verwendet, wenn Daten im Speicher überlaufen. Sie wird auch zum Speichern der Aufgabenprotokolle verwendet. Der maximale IOPS-Wert hängt ebenfalls davon ab.
- Das Ausführen mehrerer Aufgaben oder von Aufgaben mit hoher Parallelität wirkt sich auf die CPU-Auslastung der Replikations-Instance aus. Dies verlangsamt die Verarbeitung der Aufgaben und führt zu Latenz.

Bewährte Methoden

Beachten Sie bei der Dimensionierung einer Replikations-Instance diese beiden gängigsten bewährten Methoden. Weitere Informationen finden Sie unter [Bewährte Methoden für AWS Database Migration Service](#).

1. Definieren Sie Ihren Workload und finden Sie heraus, ob er computing- oder speicherintensiv ist. Auf dieser Grundlage können Sie die Klasse und Größe der Replikations-Instance bestimmen:
 - AWS DMS verarbeitet LOBs im Speicher. Dieser Vorgang benötigt eine beträchtliche Menge an Arbeitsspeicher.
 - Die Anzahl der Aufgaben und die Anzahl der Threads wirken sich auf die CPU-Auslastung aus. Vermeiden Sie es, während des Volllastvorgangs mehr als acht `MaxFullLoadSubTasks` zu verwenden.
2. Erhöhen Sie den der Replikations-Instance zugewiesenen Festplattenspeicher, wenn Sie bei Volllast einen hohen Workload haben. Auf diese Weise kann die Replikations-Instance den ihr zugewiesenen maximalen IOPS-Wert verwenden.

Die vorangegangenen Richtlinien decken nicht alle möglichen Szenarien ab. Es ist sehr wichtig, die Besonderheiten Ihres speziellen Anwendungsfalls zu berücksichtigen, wenn Sie die Größe Ihrer Replikations-Instance bestimmen.

Die vorherigen Tests haben gezeigt, dass CPU und Arbeitsspeicher je nach Workload variieren. Insbesondere LOBs wirken sich auf den Arbeitsspeicher aus und die Anzahl der Aufgaben oder die Parallelität wirken sich auf die CPU aus. Nachdem die Migration gestartet wurde, überwachen Sie die Werte für CPU, freigebbaren Arbeitsspeicher, freien Speicherplatz und E/A-Leistung pro Sekunde Ihrer Replikations-Instance. Basierend auf den Daten, die Sie erfassen, können Sie Ihre Replikations-Instance nach Bedarf vergrößern oder verkleinern.

Arbeiten mit Versionen der Replikations-Engine

Die Replication Engine ist die AWS DMS Kernsoftware, die auf Ihrer Replikationsinstanz ausgeführt wird und die von Ihnen angegebenen Migrationsaufgaben ausführt. AWS veröffentlicht regelmäßig neue Versionen der AWS DMS Replication Engine-Software mit neuen Funktionen und Leistungsverbesserungen. Jede Version der Replikations-Engine-Software verfügt über eine eigene Versionsnummer, um sie von anderen Versionen zu unterscheiden.

Wenn Sie eine neue Replikationsinstanz starten, wird auf ihr die neueste AWS DMS Engine-Version ausgeführt, sofern Sie nichts anderes angeben. Weitere Informationen finden Sie unter [Mit einer AWS DMS Replikationsinstanz arbeiten](#).

Wenn Sie eine Replikationsinstanz haben, die gerade läuft, können Sie sie auf eine neuere Engine-Version aktualisieren. (unterstützt AWS DMS keine Downgrades der Engine-Version.) Weitere Informationen zu Versionen des Replikationsmoduls finden Sie unter [AWS DMS-Versionshinweise](#).

Aktualisieren der Engine-Version über die Konsole

Sie können eine AWS DMS Replikationsinstanz mit dem AWS Management Console aktualisieren.

So aktualisieren Sie eine Replikations-Instance über die Konsole

1. Öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Replication instances (Replikations-Instances) aus.
3. Wählen Sie Ihre Replikations-Engine aus und klicken Sie dann auf Modify.
4. Wählen Sie im Feld Engine-Version die Versionsnummer aus, die Sie suchen, und klicken Sie anschließend auf Ändern.

Note

Wir empfehlen, dass Sie alle Aufgaben beenden, bevor Sie die Replikations-Instance aktualisieren. Wenn Sie die Aufgabe nicht beenden, AWS DMS wird sie vor dem Upgrade automatisch beendet. Wenn Sie die Aufgabe manuell anhalten, müssen Sie die Aufgabe nach Abschluss des Upgrades manuell starten. Das Aktualisieren der Replikations-Instance dauert einige Minuten. Sobald die Instance bereit ist, ändert sich ihr Status zu available.

Aktualisierung der Engine-Version mit dem AWS CLI

Sie können eine AWS DMS Replikationsinstanz wie folgt mit dem AWS CLI aktualisieren.

Um eine Replikationsinstanz zu aktualisieren, verwenden Sie AWS CLI

1. Bestimmen Sie den Amazon-Ressourcennamen (ARN) Ihrer Replikations-Instanz anhand des folgenden Befehls.

```
aws dms describe-replication-instances \  
--query "ReplicationInstances[*].  
[ReplicationInstanceIdentifier,ReplicationInstanceArn,ReplicationInstanceClass]"
```

Beachten Sie in der Ausgabe den ARN für die Replikations-Instanz, die Sie aktualisieren möchten, beispielsweise: `arn:aws:dms:us-east-1:123456789012:rep:6EFQQ06U6EDPRCPKLNPL2SCEEY`

2. Bestimmen Sie, welche Replikations-Instanz-Versionen verfügbar sind, anhand des folgenden Befehls.

```
aws dms describe-orderable-replication-instances \  
--query "OrderableReplicationInstances[*].[ReplicationInstanceClass,EngineVersion]"
```

Beachten Sie in der Ausgabe die Engine-Versionsnummer(n), die für Ihre Replikations-Instanz-Klasse verfügbar ist/sind. Sie sollten diese Informationen in der Ausgabe von Schritt 1 sehen.

3. Aktualisieren Sie die Replikations-Instanz anhand des folgenden Befehls.

```
aws dms modify-replication-instance \  
--replication-instance-arn arn \  
--engine-version n.n.n
```

Ersetzen Sie *arn* im vorausgehenden Befehl mit dem tatsächlichen Replikations-Instanz-ARN aus dem vorherigen Schritt.

Ersetzen Sie *n.n.n* durch die gewünschte Engine-Versionsnummer, beispielsweise: `3.4.5`

Note

Das Aktualisieren der Replikations-Instance dauert einige Minuten. Sie können den Replikations-Instance-Status mit dem folgenden Befehl anzeigen.

```
aws dms describe-replication-instances \  
--query "ReplicationInstances[*].\  
[ReplicationInstanceIdentifier,ReplicationInstanceStatus]"
```

Sobald die Replikations-Instance bereit ist, ändert sich ihr Status zu available.

Öffentliche und private Replikations-Instances

Sie können festlegen, ob eine Replikations-Instance eine öffentliche oder private IP-Adresse hat, die von der Instance verwendet wird, um eine Verbindung mit der Quell- und Zieldatenbank herzustellen.

Eine private Replikations-Instance verfügt über eine private IP-Adresse, auf die Sie von außerhalb des Replikationsnetzwerks nicht zugreifen können. Sie verwenden eine private Instance, wenn sich sowohl Quell- als auch Zieldatenbanken in demselben Netzwerk befinden, das mit der Virtual Private Cloud (VPC) der Replikations-Instance verbunden ist. Das Netzwerk kann über ein virtuelles privates Netzwerk (VPN) oder VPC-Peering mit der VPC verbunden werden. AWS Direct Connect

Eine VPC-Peering-Verbindung ist eine Netzwerkverbindung zwischen zwei VPCs. Dies ermöglicht das Routing unter Verwendung der privaten IP-Adressen jeder VPC, als ob sie sich im selben Netzwerk befinden würden. Weitere Informationen zu VPC Peering finden Sie unter [VPC Peering](#) im Amazon VPC Benutzerhandbuch.

Eine öffentliche Replikations-Instance kann die VPC-Sicherheitsgruppe der Replikations-Instance und die öffentliche IP-Adresse der Replikations-Instance oder die öffentliche IP-Adresse des NAT-Gateways verwenden. Diese Verbindungen bilden ein Netzwerk, das Sie für die Datenmigration nutzen.

IP-Adressierung und Netzwerktypen

AWS DMS erstellt Ihre Replikationsinstanz immer in einer Amazon Virtual Private Cloud (VPC). Wenn Sie Ihre VPC erstellen, können Sie die zu verwendende IP-Adressierung festlegen: entweder IPv4, IPv6 oder beide. Wenn Sie dann eine Replikations-Instance erstellen oder ändern, können Sie die

Verwendung eines IPv4-Adressprotokolls oder eines IPv6-Adressprotokolls im Dual-Stack-Modus angeben.

IPv4-Adressen

Wenn Sie eine VPC erstellen, müssen Sie für die VPC einen IPv4-Adressbereich in Form eines Classless Inter-Domain Routing (CIDR)-Blocks festlegen, zum Beispiel 10.0.0.0/16. Eine Subnetzgruppe definiert den Bereich der IP-Adressen in diesem CIDR-Block. Diese IP-Adressen können privat oder öffentlich sein.

Eine private IPv4-Adresse ist eine IP-Adresse, die nicht über das Internet erreichbar ist. Sie können private IPv4-Adressen zur Kommunikation zwischen Ihrer Replikations-Instance und anderen Ressourcen, wie Amazon-EC2-Instances, in derselben VPC verwenden. Jede Replikations-Instance hat eine private IP-Adresse für die Kommunikation in der VPC.

Eine öffentliche IP-Adresse ist eine IPv4-Adresse, die über das Internet erreichbar ist. Sie können öffentliche Adressen zur Kommunikation zwischen Ihrer Replikations-Instance und Ressourcen im Internet verwenden. Sie können kontrollieren, ob Ihre Replikations-Instance eine öffentliche IP-Adresse erhält

Dual-Stack-Modus und IPv6-Adressen

Wenn Sie über Ressourcen verfügen, die über IPv6 mit Ihrer Replikations-Instance kommunizieren müssen, verwenden Sie den Dual-Stack-Modus. Stellen Sie zur Verwendung des Dual-Stack-Modus sicher, dass jedem Subnetz in der Subnetzgruppe, das Sie mit der Replikations-Instance verknüpfen, ein IPv6-CIDR-Block zugeordnet ist. Sie können eine neue Replikations-Subnetzgruppe erstellen oder eine vorhandene Replikations-Subnetzgruppe ändern, um diese Anforderung zu erfüllen. Jede IPv6-Adresse ist global eindeutig. Der IPv6-CIDR-Block für Ihre VPC wird automatisch aus dem Amazon-Pool von IPv6-Adressen zugewiesen. Sie können den Bereich nicht selbst auswählen.

DMS deaktiviert den Internet-Gateway-Zugriff für IPv6-Endpunkte privater Replikations-Instances im Dual-Stack-Modus. DMS tut dies, um sicherzustellen, dass Ihre IPv6-Endpunkte privat und nur von Ihrer VPC aus zugänglich sind.

Sie können die AWS DMS Konsole verwenden, um eine Replikationsinstanz zu erstellen oder zu ändern, und im Abschnitt Netzwerktyp den Dual-Stack-Modus angeben. Die folgende Abbildung zeigt den Abschnitt Network type (Netzwerktyp) in der Konsole.

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.

Referenzen

- Weitere Informationen zu IPv4- und IPv6-Adressen finden Sie unter [IP-Adresszuweisung](#) im Amazon-VPC-Benutzerhandbuch.
- Weitere Hinweise zum Erstellen einer Replikations-Instance im Dual-Stack-Modus finden Sie unter [Erstellen einer Replikations-Instance](#).
- Weitere Informationen zum Ändern einer Replikations-Instance finden Sie unter [Ändern einer Replikations-Instance](#).

Einrichten eines Netzwerks für eine Replikations-Instance

AWS DMS erstellt die Replikationsinstanz immer in einer VPC, die auf Amazon VPC basiert. Geben Sie die VPC an, in der sich Ihre Replikations-Instance befindet. Sie können Ihre Standard-VPC für Ihr Konto und Ihre AWS Region verwenden oder eine neue VPC erstellen.

Stellen Sie sicher, dass das für die VPC Ihrer Replikations-Instance zugewiesene Elastic Network Interface einer Sicherheitsgruppe zugeordnet ist. Stellen Sie außerdem sicher, dass die Regeln dieser Sicherheitsgruppe zulassen, dass der gesamte Datenverkehr auf allen Ports die VPC verlässt (ausgehender Datenverkehr). Dieser Ansatz ermöglicht die Kommunikation von der Replikations-Instance zu Ihren Quell- und Zieldatenbankendpunkten, sofern auf den Endpunkten die korrekten Eingangsregeln aktiviert sind. Wir empfehlen, die Standardeinstellungen für die Endpunkte zu verwenden. Diese erlauben ausgehenden Zugriff auf allen Ports für alle Adressen.

Die Quell- und Zielendpunkte greifen auf die Replikations-Instance zu, die sich in der VPC befindet, indem sie sich entweder mit der VPC verbinden oder sich selbst innerhalb der VPC befinden. Die Datenbankendpunkte müssen Netzwerk-Zugriffskontrolllisten (ACLs) und Sicherheitsgruppenregeln

(falls zutreffend) enthalten, die eingehenden Zugriff von der Replikations-Instance erlauben. Wie Sie dies einrichten, hängt von der verwendeten Netzwerkkonfiguration ab. Sie können die VPC-Sicherheitsgruppe der Replikations-Instance, die private oder öffentliche IP-Adresse der Replikations-Instance oder die öffentliche IP-Adresse des NAT-Gateways verwenden. Diese Verbindungen bilden ein Netzwerk, das Sie für die Datenmigration nutzen.

Note

Da sich eine IP-Adresse aufgrund von Änderungen an der zugrunde liegenden Infrastruktur ändern kann, empfehlen wir Ihnen, einen VPC-CIDR-Bereich zu verwenden oder den ausgehenden Datenverkehr Ihrer Replikations-Instance über eine mit NAT GW verknüpfte Elastic IP weiterzuleiten. Weitere Informationen zum Erstellen einer VPC, einschließlich eines CIDR-Blocks, finden Sie unter [Arbeiten mit VPCs und Subnetzen](#) im Amazon-Virtual-Private-Cloud-Benutzerhandbuch. Weitere Informationen zu Elastic IP-Adressen finden Sie im Amazon-Elastic-Compute-Cloud-Benutzerhandbuch unter [Elastic IP-Adressen](#).

Netzwerkkonfigurationen für die Datenbankmigration

Sie können mehrere verschiedene Netzwerkkonfigurationen mit dem AWS Database Migration Service verwenden. Im Folgenden werden einige häufige Konfigurationen für ein Netzwerk erörtert, die für die Datenbankmigration verwendet werden.

Themen

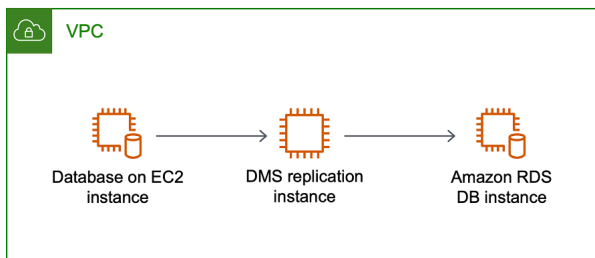
- [Konfiguration mit allen Datenbankmigrationskomponenten in einer VPC](#)
- [Konfiguration mit mehreren VPCs](#)
- [Konfiguration mit gemeinsam genutzten VPCs](#)
- [Konfiguration für ein Netzwerk zu einer VPC unter Verwendung AWS Direct Connect eines VPN](#)
- [Konfiguration für ein Netzwerk zu einer VPC über das Internet](#)
- [Konfiguration mit einer RDS-DB-Instance, die sich nicht in einer VPC befindet, zu einer DB-Instance in einer VPC mit ClassicLink](#)

Wenn möglich, empfehlen wir, dass Sie eine DMS-Replikations-Instance in derselben Region wie Ihr Zielpunkt und in derselben VPC oder demselben Subnetz wie Ihr Zielpunkt erstellen.

Konfiguration mit allen Datenbankmigrationskomponenten in einer VPC

Die einfachste Netzwerkkonfiguration für die Datenbankmigration besteht darin, dass sich Quellendpunkt, Replikations-Instance und Zielendpunkt alle in derselben VPC befinden. Diese Konfiguration eignet sich, wenn sich Ihre Quell- und Zielendpunkte auf einer Amazon-RDS-DB-Instance oder einer Amazon-EC2-Instance befinden.

Die folgende Abbildung zeigt eine Konfiguration, bei der sich eine Datenbank auf einer Amazon EC2-Instance mit der Replikations-Instance verbindet und die Daten zu einer Amazon RDS-DB-Instance migriert werden.



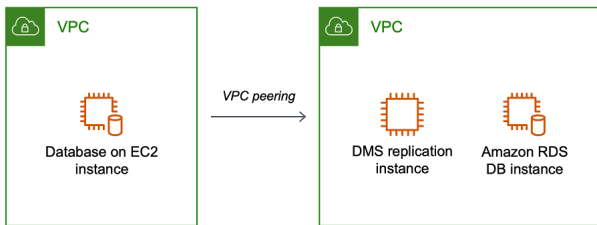
Die VPC-Sicherheitsgruppe, die in dieser Konfiguration verwendet wird, muss eingehenden Zugriff am Datenbank-Port von der Replikations-Instance erlauben. Dafür stehen Ihnen mehrere Optionen zur Verfügung. Stellen Sie sicher, dass die von der Replikations-Instance verwendete Sicherheitsgruppe Zugang zu den Endpunkten hat. Oder Sie können den VPC-CIDR-Bereich, NAT GW Elastic IP oder die private IP-Adresse der Replikations-Instance zulassen, falls Sie eine verwenden. Wir empfehlen jedoch nicht, die private IP-Adresse der Replikations-Instance zu verwenden, da dies Ihre Replikation unterbrechen kann, wenn sich die Replikations-IP-Adresse ändert.

Konfiguration mit mehreren VPCs

Wenn sich Ihre Quell- und Zielendpunkte in verschiedenen VPCs befinden, können Sie Ihre Replikations-Instance in einer der VPCs erstellen. Anschließend können Sie die beiden VPCs mithilfe von VPC-Peering verknüpfen.

Eine VPC Peering-Verbindung ist eine Netzwerkverbindung zwischen zwei VPCs. Sie ermöglicht die Weiterleitung über die privaten IP-Adressen jeder VPC, als befänden sie sich im selben Netzwerk. Sie können eine VPC-Peering-Verbindung zwischen Ihren eigenen VPCs, mit einer VPC in einem anderen AWS Konto oder mit einer VPC in einer anderen Region herstellen. AWS Weitere Informationen zu VPC Peering finden Sie unter [VPC Peering](#) im Amazon VPC Benutzerhandbuch.

Die folgende Abbildung zeigt eine Beispielkonfiguration unter Verwendung von VPC-Peering. Hier wird die Quell-Datenbank auf einer Amazon-EC2-Instance in einer VPC über VPC-Peering mit einer VPC verbunden. Diese VPC enthält die Replikations-Instance und die Zieldatenbank auf einer Amazon-RDS-DB-Instance.



Um VPC-Peering zu implementieren, folgen Sie den Anweisungen unter [Arbeiten mit VPC-Peering-Verbindungen](#) in der Dokumentation Amazon Virtual Private Cloud, VPC Peering. Stellen Sie sicher, dass die Routing-Tabelle einer VPC den CIDR-Block der anderen enthält. Wenn VPC A beispielsweise das Ziel 10.0.0.0/16 und VPC B das Ziel 172.31.0.0 verwendet, sollte die Routing-Tabelle von VPC A 172.31.0.0 enthalten und die Routing-Tabelle von VPC B muss 10.0.0.0/16 enthalten. Ausführlichere Informationen finden Sie unter [Aktualisieren Ihrer Routing-Tabellen für VPC-Peering-Verbindungen](#) in der Dokumentation Amazon Virtual Private Cloud, VPC-Peering.

Die VPC-Sicherheitsgruppen, die in dieser Konfiguration verwendet werden, müssen eingehenden Zugriff am Datenbank-Port von der Replikations-Instance oder den Eingang auf dem CIDR-Block für die VPC für das Peering zulassen.

Konfiguration mit gemeinsam genutzten VPCs

AWS DMS behandelt Subnetze, die für ein teilnehmendes Kundenkonto in einer Organisation gemeinsam genutzt werden, genauso wie normale Subnetze desselben Kontos. Im Folgenden wird beschrieben, wie AWS DMS mit VPCs und Subnetzen umgegangen wird und wie Sie gemeinsam genutzte VPCs verwenden können.

Sie können Ihre Netzwerkkonfiguration so konfigurieren, dass sie in benutzerdefinierten Subnetzen oder VPCs funktioniert, indem Sie `ReplicationSubnetGroup`-Objekte erstellen. Wenn Sie eine `ReplicationSubnetGroup` erstellen, können Sie wählen, ob Sie Subnetze von einer bestimmten VPC in Ihrem Konto angeben möchten. Die Liste der von Ihnen angegebenen Subnetze muss mindestens zwei Subnetze enthalten, die sich in separaten Availability Zones befinden, und alle Subnetze müssen sich in derselben VPC befinden. Beim Erstellen einer geben Kunden nur `ReplicationSubnetGroup` Subnetze an. AWS DMS bestimmt die VPC in Ihrem Namen, da jedes Subnetz mit genau einer VPC verknüpft ist.

Wenn Sie eine AWS DMS `ReplicationInstance` oder eine erstellen, können Sie wählen AWS DMS `ReplicationConfig`, ob Sie eine `ReplicationSubnetGroup` und/oder eine VPC-Sicherheitsgruppe angeben möchten, in der die `ReplicationInstance` oder die serverlose Replikation ausgeführt wird. Falls nicht angegeben, AWS DMS wählt der Kundenstandard `ReplicationSubnetGroup` (der in Ihrem Namen AWS DMS erstellt wird, falls nicht für alle Subnetze in der Standard-VPC angegeben) und die Standard-VPC-Sicherheitsgruppe.

Sie können wählen, ob Sie Ihre Migrationen in einer von Ihnen angegebenen Availability Zone oder in einer der Availability Zones in Ihrer `ReplicationSubnetGroup` ausführen möchten. Wenn AWS DMS versucht wird, eine Replikationsinstanz zu erstellen oder eine serverlose Replikation zu starten, werden die Verfügbarkeitszonen Ihrer Subnetze in Verfügbarkeitszonen im Hauptdienstkonto übersetzt, um sicherzustellen, dass wir Instances in der richtigen Availability Zone starten, auch wenn die Availability Zone-Zuordnungen zwischen den beiden Konten nicht identisch sind.

Wenn Sie eine gemeinsam genutzte VPC verwenden, müssen Sie sicherstellen, dass Sie `ReplicationSubnetGroup`-Objekte erstellen, die den Subnetzen zugeordnet sind, die Sie von einer gemeinsam genutzten VPC aus verwenden möchten. Wenn Sie eine `ReplicationInstance` oder eine `ReplicationConfig` erstellen, müssen Sie eine `ReplicationSubnetGroup` für die gemeinsam genutzte VPC und eine VPC-Sicherheitsgruppe angeben, die Sie für Ihre gemeinsam genutzte VPC mit Ihrer Create-Anfrage erstellt haben.

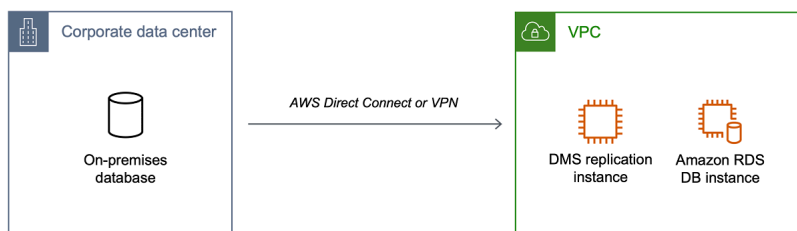
Beachten Sie Folgendes im Zusammenhang mit der Verwendung einer gemeinsam genutzten VPC:

- Der VPC-Eigentümer kann keine Ressource für einen Teilnehmer freigeben, aber der Teilnehmer kann eine `ServiceResource` im Subnetz des Eigentümers erstellen.
- Der VPC-Eigentümer kann nicht auf eine Ressource (z. B. eine Replikations-Instance) zugreifen, die der Teilnehmer erstellt, da alle Ressourcen kontospezifisch sind. Solange Sie die Replikations-Instance jedoch in der gemeinsam genutzten VPC erstellen, kann sie unabhängig vom Eigentümerkonto auf die Ressourcen in der VPC zugreifen, sofern der Replikationsendpunkt oder die Aufgabe über die korrekten Berechtigungen verfügt.
- Da Ressourcen kontospezifisch sind, können andere Teilnehmer nicht auf Ressourcen zugreifen, die anderen Konten gehören. Es gibt keine Berechtigungen, die Sie anderen Konten erteilen können, damit sie auf Ressourcen zugreifen können, die in der gemeinsam genutzten VPC mit Ihrem Konto erstellt wurden.

Konfiguration für ein Netzwerk zu einer VPC unter Verwendung AWS Direct Connect eines VPN

Remote-Netzwerke können über verschiedene Optionen wie AWS Direct Connect oder eine Software- oder Hardware-VPN-Verbindung eine Verbindung zu einer VPC herstellen. Diese Optionen werden oft verwendet, um lokale Services zu integrieren, wie beispielsweise Überwachung, Authentifizierung, Sicherheit, Daten oder andere Systeme, indem ein internes Netzwerk in die AWS Cloud erweitert wird. Mit dieser Art von Netzwerkerweiterung können Sie problemlos eine Verbindung mit von AWS gehosteten Ressourcen wie einer VPC herstellen.

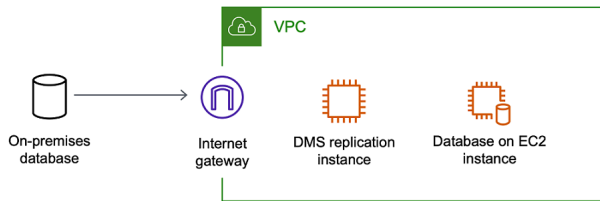
Die folgende Abbildung zeigt eine Konfiguration, bei der der Quellendpunkt eine lokale Datenbank in einem firmeneigenen Rechenzentrum ist. Die Verbindung erfolgt über AWS Direct Connect oder ein VPN mit einer VPC, die die Replikations-Instance und eine Zieldatenbank auf einer Amazon-RDS-DB-Instance enthält.



Bei dieser Konfiguration muss die VPC-Sicherheitsgruppe eine Routing-Regel enthalten, die den für einen VPC-CIDR-Bereich oder eine spezifische IP-Adresse bestimmten Datenverkehr an einen Host sendet. Dieser Host muss dazu in der Lage sein, den von der VPC kommenden Datenverkehr in ein lokales VPN weiterzuleiten. In diesem Fall enthält der NAT-Host seine eigenen Sicherheitsgruppeneinstellungen. Diese Einstellungen müssen Datenverkehr aus dem VPC-CIDR-Bereich, der privaten IP-Adresse oder der Sicherheitsgruppe der Replikations-Instance in die NAT-Instance zulassen. Wir empfehlen jedoch nicht, die private IP-Adresse der Replikations-Instance zu verwenden, da dies Ihre Replikation unterbrechen kann, wenn sich die Replikations-IP-Adresse ändert.

Konfiguration für ein Netzwerk zu einer VPC über das Internet

Wenn Sie kein VPN verwenden oder keine Verbindung AWS Direct Connect zu AWS Ressourcen herstellen, können Sie Ihre Datenbank über das Internet migrieren. In diesem Fall können Sie entweder zu einer AmazonEC2-Instance oder einer Amazon-RDS-DB-Instance migrieren. Diese Konfiguration umfasst eine öffentliche Replikations-Instance in einer VPC mit einem Internet-Gateway, das den Zielendpunkt und die Replikations-Instance enthält.



Informationen zum Anfügen eines Internet-Gateways zu Ihrer VPC finden Sie unter [Anfügen eines Internet-Gateways](#) im Amazon VPC Benutzerhandbuch.

Die VPC-Routing-Tabelle muss Routing-Regeln enthalten, die den nicht für die VPC bestimmten Datenverkehr standardmäßig an das Internet-Gateway senden. Bei dieser Konfiguration stammt die Verbindung mit dem Endpunkt scheinbar von der öffentlichen IP-Adresse der Replikations-Instance und nicht von der privaten IP-Adresse. Weitere Informationen finden Sie unter [VPC-Routing-Tabellen](#) im Amazon-VPC-Benutzerhandbuch.

Konfiguration mit einer RDS-DB-Instance, die sich nicht in einer VPC befindet, zu einer DB-Instance in einer VPC mit ClassicLink

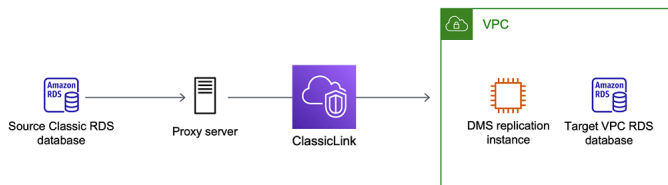
EC2-Classic wird am 15. August 2022 eingestellt. Wir empfehlen Ihnen die Migration von EC2-Classic zu einer VPC. Weitere Informationen finden Sie unter [Migration von EC2-Classic zu einer VPC](#) im Benutzerhandbuch für Amazon EC2 und im Blog [EC2-Classic Networking is Retiring – Here's How to Prepare](#) (EC2-Classic Networking geht in den Ruhestand – So bereiten Sie sich vor).

Um eine Amazon RDS-DB-Instance, die sich nicht in einer VPC befindet, mit einem DMS-Replikationsserver und einer DB-Instance in einer VPC zu verbinden, können Sie sie ClassicLink mit einem Proxy-Server verwenden.

ClassicLink ermöglicht es Ihnen, eine EC2-Classic-DB-Instance mit einer VPC in Ihrem Konto innerhalb derselben Region zu verknüpfen. AWS Nachdem Sie die Verknüpfung erstellt haben, kann die DB-Quell-Instance mit der Replikations-Instance in der VPC über ihre privaten IP-Adressen kommunizieren.

Da die Replikationsinstanz in der VPC nicht direkt auf die Quell-DB-Instance auf der EC2-Classic-Plattform zugreifen kann ClassicLink, verwenden Sie einen Proxyserver. Der Proxy-Server verbindet die Quell-DB-Instance mit der VPC, die die Replikations-Instance und Ziel-DB-Instance enthält. Der Proxyserver verwendet ClassicLink , um eine Verbindung zur VPC herzustellen. Die Port-

Weiterleitung auf dem Proxy-Server ermöglicht die Kommunikation zwischen der Quell-DB-Instance und der Ziel-DB-Instance in der VPC.



Verwendung ClassicLink mit AWS Database Migration Service

Sie können eine Amazon RDS-DB-Instance, die sich nicht in einer VPC befindet, mit einem AWS DMS-Replikationsserver und einer DB-Instance verbinden, die sich in einer VPC befinden. Dazu können Sie Amazon EC2 ClassicLink mit einem Proxy-Server verwenden.

Das folgende Verfahren zeigt, wie Sie es ClassicLink für diesen Zweck verwenden. Dieses Verfahren verbindet eine Amazon RDS-Quell-DB-Instance, die sich nicht in einer VPC befindet, mit einer VPC, die eine AWS DMS-Replikations-Instance und eine Ziel-DB-Instance enthält.

- Erstellen Sie eine AWS DMS-Replikationsinstanz in einer VPC. (Alle Replikations-Instances werden in einer VPC erstellt.)
- Verknüpfen Sie eine VPC-Sicherheitsgruppe mit der Replikations-Instance und der DB-Ziel-Instance. Wenn zwei Instances gemeinsam eine VPC-Sicherheitsgruppe verwenden, können sie standardmäßig miteinander kommunizieren.
- Richten Sie einen Proxy-Server auf einer EC2-Classic-Instance ein.
- Stellen Sie eine Verbindung ClassicLink zwischen dem Proxyserver und der VPC her.
- Erstellen Sie AWS DMS-Endpunkte für die Quell- und Zieldatenbanken.
- Erstellen Sie eine AWS DMS-Aufgabe.

Wird verwendet, ClassicLink um eine Datenbank auf einer DB-Instance, die sich nicht in einer VPC befindet, zu einer Datenbank auf einer DB-Instance in einer VPC zu migrieren

1. Erstellen Sie eine AWS DMS-Replikationsinstanz und weisen Sie eine VPC-Sicherheitsgruppe zu:
 - a. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter https://console.aws.amazon.com/dms/v2/.](https://console.aws.amazon.com/dms/v2/)

Wenn Sie als AWS Identity and Access Management (IAM-) Benutzer angemeldet sind, stellen Sie sicher, dass Sie über die entsprechenden Zugriffsberechtigungen

- verfügen. AWS DMS Weitere Informationen zu den erforderlichen Berechtigungen für die Datenbankmigration finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).
- b. Wählen Sie auf der Seite Dashboard Replication Instance (Replikations-Instance). Befolgen Sie die Anweisungen unter [Schritt 1: Erstellen einer Replikations-Instance über die AWS DMS-Konsole](#) zum Erstellen einer Replikations-Instance.
 - c. Nachdem Sie die AWS DMS-Replikationsinstanz erstellt haben, öffnen Sie die EC2-Servicekonsole. Wählen Sie im Navigationsbereich Netzwerk-Interfaces aus.
 - d. Wählen Sie das DMS und NetworkInterface dann im Menü Aktionen die Option Sicherheitsgruppen ändern aus.
 - e. Wählen Sie die Sicherheitsgruppe aus, die Sie für die Replikations-Instance und die DB-Ziel-Instance verwenden möchten.
2. Verknüpfen Sie die Sicherheitsgruppe aus dem letzten Schritt mit der DB-Ziel-Instance:
- a. Öffnen Sie die Amazon RDS-Konsole. Wählen Sie im Navigationsbereich Instances aus.
 - b. Wählen Sie die DB-Ziel-Instance aus. Wählen Sie für Instance-Aktionen die Option Ändern.
 - c. Wählen Sie für den Parameter Sicherheitsgruppe die Sicherheitsgruppe aus, die Sie im vorherigen Schritt verwendet haben.
 - d. Wählen Sie Weiter und dann DB-Instance ändern.
3. Schritt 3: Einrichten eines Proxy-Servers auf einer EC2-Classic-Instance mit NGINX. Verwenden Sie ein AMI Ihrer Wahl, um eine EC2-Classic-Instance zu starten. Das folgende Beispiel basiert auf AMI Ubuntu Server 14.04 LTS (HVM).

So richten Sie einen Proxy-Server auf einer EC2-Classic-Instance ein

- a. Stellen Sie eine Verbindung mit der EC2-Classic-Instance her und installieren Sie NGINX mithilfe der folgenden Befehle:

```
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. Bearbeiten Sie die NGINX-Daemon-Datei, `/etc/init/nginx.conf`, mit folgendem Code:

```
# /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. Erstellen Sie eine NGINX-Konfigurationsdatei unter `/usr/local/nginx/conf/nginx.conf`. Fügen Sie in der Konfigurationsdatei Folgendes hinzu:

```
# /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. Starten Sie von der Befehlszeile NGINX mithilfe der folgenden Befehle:

```
Prompt> sudo initctl reload-configuration
Prompt> sudo initctl list | grep nginx
Prompt> sudo initctl start nginx
```

4. Stellen Sie eine ClassicLink Verbindung zwischen dem Proxyserver und der Ziel-VPC her, die die Ziel-DB-Instance und die Replikationsinstanz enthält:
 - a. Öffnen Sie die EC2-Konsole und wählen Sie die EC2-Classic-Instance aus, auf der der Proxy-Server ausgeführt wird.
 - b. Wählen Sie für Aktionen die Option ClassicLink und anschließend Link to VPC aus.
 - c. Wählen Sie die Sicherheitsgruppe aus, die Sie zuvor in diesem Verfahren verwendet haben.
 - d. Wählen Sie Mit VPC verknüpfen.
5. Schritt 5: Erstellen Sie AWS DMS-Endpunkte mithilfe des Verfahrens unter. [Schritt 2: Angeben von Quell- und Zielendpunkten](#) Sie müssen den internen EC2-DNS-Host-Namen des Proxy-Servers als Servernamen verwenden, wenn Sie den Quellendpunkt angeben.
6. Erstellen Sie eine AWS DMS-Aufgabe mit dem Verfahren unter. [Schritt 3: Erstellen einer Aufgabe und Migrieren der Daten](#)

Erstellen einer Replikationssubnetzgruppe

Als Teil des für die Datenbankmigration zu verwendenden Netzwerks müssen Sie angeben, welche Subnetze in Ihrer Virtual Private Cloud (VPC) verwendet werden sollen. Diese VPC muss auf dem Amazon-VPC-Service basieren. Ein Subnetz ist ein IP-Adressbereich in Ihrer VPC in einer bestimmten Availability Zone. Diese Subnetze können auf die Availability Zones der AWS Region verteilt werden, in der sich Ihre VPC befindet.

Wenn Sie eine Replikationsinstanz oder ein Instanzprofil in der AWS DMS-Konsole erstellen, können Sie das von Ihnen gewählte Subnetz verwenden.

Sie erstellen eine Replikationssubnetzgruppe, um zu definieren, welche Subnetze verwendet werden. Sie müssen Subnetze in mindestens zwei Availability Zones angeben.

So erstellen Sie eine Replikationssubnetzgruppe

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter https://console.aws.amazon.com/dms/v2/.](https://console.aws.amazon.com/dms/v2/)

Wenn Sie als IAM-Benutzer angemeldet sind, müssen Sie über die entsprechenden Berechtigungen für den Zugriff auf AWS DMS verfügen. Weitere Informationen zu den erforderlichen Berechtigungen für die Datenbankmigration finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
3. Wählen Sie Create subnet group (Subnetz-Grupp erstellen) aus.
4. Geben Sie auf der Seite Replikations-Subnetzgruppe bearbeiten die Informationen zu Ihrer Replikationssubnetzgruppe an. In der folgenden Tabelle sind die Einstellungen beschrieben.

Option	Aktion
Name	Geben Sie einen Namen für die Replikationssubnetzgruppe an, der aus 8 bis 16 druckbaren ASCII-Zeichen (ohne /, " und @) besteht. Der Name sollte für Ihr Konto in der von Ihnen ausgewählten AWS Region eindeutig sein. Sie können dem Namen Informationen hinzufügen, z. B. die AWS Region und die Aufgabe, die Sie gerade ausführen, angeben DMS-default-VPC .
Beschreibung	Geben Sie eine kurze Beschreibung für die Replikationssubnetzgruppe ein.
VPC	Wählen Sie die gewünschte VPC für die Datenbankmigration aus. Beachten Sie, dass die VPC mindestens

Option	Aktion
	s über ein Subnetz in mindestens zwei Availability Zones verfügen muss.
Subnetze hinzufügen	Wählen Sie die Subnetze aus, die in der Replikationsnetzgruppe enthalten sein sollen. Sie müssen Subnetze in mindestens zwei Availability Zones auswählen.

5. Wählen Sie Create subnet group (Subnetz-Gruppe erstellen) aus.

Auflösen von Domain-Endpunkten mithilfe von DNS

Normalerweise verwendet eine AWS DMS Replikationsinstanz den Domain Name System (DNS) -Resolver in einer Amazon EC2 Instance, um Domain-Endpunkte aufzulösen. Wenn Sie eine DNS-Auflösung benötigen, können Sie den Amazon Route 53 Resolver verwenden. Weitere Informationen zur Verwendung von Route 53 DNS Resolver finden Sie unter [Erste Schritte mit Route 53 Resolver](#).

Für Informationen dazu, wie Sie Ihren eigenen On-Premises-Namensserver verwenden, um bestimmte Endpunkte aufzulösen, wenn Sie den Amazon Route 53 Resolver verwenden, siehe [Verwenden Ihres eigenen Vor-Ort-Nameservers](#).

Festlegen eines Verschlüsselungsschlüssels für eine Replikations-Instance

AWS DMS verschlüsselt den von einer Replikationsinstanz verwendeten Speicher und die Verbindungsinformationen des Endpunkts. Um den von einer Replikationsinstanz verwendeten Speicher zu verschlüsseln, verwendet AWS DMS eine, AWS KMS key die für Ihr Konto eindeutig ist. Sie können diesen KMS-Schlüssel mit AWS Key Management Service (AWS KMS) anzeigen und verwalten. Sie können den Standard-KMS-Schlüssel in Ihrem Konto (aws/dms) verwenden oder Sie können einen eigenen KMS-Schlüssel erstellen. Wenn Sie bereits über einen AWS KMS Verschlüsselungsschlüssel verfügen, können Sie diesen Schlüssel auch für die Verschlüsselung verwenden.

Sie können Ihren eigenen Verschlüsselungsschlüssel angeben, indem Sie eine KMS-Schlüssel-ID zur Verschlüsselung Ihrer AWS DMS-Ressourcen angeben. Wenn Sie Ihren

eigenen Verschlüsselungsschlüssel angeben, muss das Benutzerkonto, mit dem die Datenbankmigration durchgeführt wird, Zugriff auf diesen Schlüssel haben. Weitere Informationen zum Erstellen eigener Verschlüsselungsschlüssel und zum Erteilen von Zugriffsrechten auf einen Verschlüsselungsschlüssel für Benutzer finden Sie im [Entwicklerhandbuch für AWS KMS](#).

Wenn Sie keine KMS-Schlüssel-ID angeben, verwendet AWS DMS Ihren Standard-Verschlüsselungsschlüssel. KMS erstellt den Standard-Verschlüsselungsschlüssel für AWS DMS für Ihr AWS Konto. Ihr AWS Konto hat für jede AWS Region einen anderen Standardverschlüsselungsschlüssel.

Um die Schlüssel zu verwalten, die für die Verschlüsselung Ihrer AWS DMS-Ressourcen verwendet werden, verwenden Sie AWS KMS. Sie finden es in der AWS Management Console, indem Sie im Navigationsbereich nach KMS suchen.

AWS KMS kombiniert sichere, hochverfügbare Hardware und Software zu einem für die Cloud skalierten Schlüsselverwaltungssystem. Mithilfe AWS KMS können Sie Verschlüsselungsschlüssel erstellen und die Richtlinien definieren, die steuern, wie diese Schlüssel verwendet werden können. AWS KMS unterstützt AWS CloudTrail, sodass Sie die Verwendung von Schlüsseln überprüfen können, um sicherzustellen, dass die Schlüssel ordnungsgemäß verwendet werden. Ihre AWS KMS Schlüssel können in Kombination mit AWS DMS und anderen unterstützten AWS Diensten verwendet werden. Zu den unterstützten AWS -Services gehören Amazon RDS, Amazon S3, Amazon Elastic Block Store (Amazon EBS) und Amazon Redshift.

Wenn Sie Ihre AWS DMS-Ressourcen mit einem bestimmten Verschlüsselungsschlüssel erstellt haben, können Sie den Verschlüsselungsschlüssel für diese Ressourcen nicht ändern. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel ermitteln, bevor Sie Ihre AWS DMS-Ressourcen erstellen.

Erstellen einer Replikations-Instance

Ihre erste Aufgabe bei der Migration einer Datenbank besteht darin, eine Replikations-Instance zu erstellen. Die Replikations-Instance benötigt genügend Speicherplatz und Rechenleistung für die Durchführung der zugewiesenen Aufgaben und die Migration der Daten von der Quelldatenbank zur Zieldatenbank. Die erforderliche Größe dieser Instance hängt von der Menge der Daten ab, die Sie migrieren müssen, sowie von den Aufgaben, die die Instance ausführen muss. Weitere Informationen zu Replikations-Instances finden Sie unter [Mit einer AWS DMS Replikationsinstanz arbeiten](#).

So erstellen Sie eine Replikationsinstanz mithilfe der AWS Konsole

1. Wählen Sie im Navigationsbereich der AWS DMS Konsole Replikationsinstanzen und dann Replikationsinstanz erstellen aus.
2. Legen Sie auf der Seite Create replication instance die Angaben zur Replikations-Instance fest. In der folgenden Tabelle sind die möglichen Einstellungen beschrieben.

Option	Aktion
Name	Geben Sie einen Namen für die Replikations-Instanz an, der aus 8 bis 16 druckbaren ASCII-Zeichen (ohne /, " und @) besteht. Der Name sollte für Ihr Konto und die gewählte AWS -Region eindeutig sein. Sie können dem Namen Informationen hinzufügen, z. B. die AWS Region und die Aufgabe, die Sie gerade ausführen, angeben west2-mysql1mysql-instance1 .
Beschreibender Amazon-Ressourcenname (ARN) – optional	Ein benutzerfreundlicher Name, um den Standard-DMS-ARN zu überschreiben. Sie können diesen nach der Erstellung nicht mehr ändern.
Beschreibung	Geben Sie eine kurze Beschreibung für die Replikations-Instance ein.
Instance-Klasse	Wählen Sie eine Instance-Klasse mit der benötigten Konfiguration für die Migration aus. Beachten Sie, dass die Instance über ausreichend Speicherplatz, Netzwerkkapazität und Rechenleistung verfügen muss, um Ihre Migration erfolgreich abzuschließen. Weitere Informationen dazu, wie Sie herausfinden, welche Instance-Klasse sich am besten für Ihre Migration eignet, finden Sie unter Mit einer AWS DMS Replikationsinstanz arbeiten .
Engine-Version	In der AWS DMS Konsole können Sie eine beliebige unterstützte Engine-Version auswählen. Von der AWS CLI aus führt die Replikationsinstanz die neueste

Option	Aktion
	Nicht-Beta-Version der AWS DMS Replikations-Engine aus, sofern Sie in der AWS CLI keine andere Engine-Version angeben.
Hochverfügbarkeit	Verwenden Sie diesen optionalen Parameter zum Erstellen eines Standby-Replikats Ihrer Replikations-Instance in einer anderen Availability Zone, um Failover-Unterstützung zu bieten. Wenn Sie Change Data Capture (CDC) oder die fortlaufende Replikation verwenden möchten, sollten Sie diese Option aktivieren.

Option	Aktion
Allocated storage (GiB) (Zugewiesener Speicher)	<p>Speicherplatz wird in erster Linie von Protokolldateien und zwischengespeicherten Transaktionen belegt. Für die zwischengespeicherten Transaktionen wird Speicherplatz nur verwendet, wenn die zwischengespeicherten Transaktionen auf Festplatte geschrieben werden müssen. Daher verwendet AWS DMS keine nennenswerte Menge an Speicherplatz. Es gibt jedoch folgende Ausnahmen:</p> <ul style="list-style-type: none">• Sehr große Tabellen, bei der viele Transaktionen geladen werden. Das Laden einer großen Tabelle kann einige Zeit in Anspruch nehmen, sodass zwischengespeicherte Transaktionen eher während des Ladens einer großen Tabelle auf Festplatte geschrieben werden.• Aufgaben, die so konfiguriert sind, dass sie vor dem Laden zwischengespeicherter Transaktionen angehalten werden. In diesem Fall werden alle Transaktionen zwischengespeichert, bis der vollständige Ladevorgang für alle Tabellen abgeschlossen ist. Bei dieser Konfiguration kann durch die zwischengespeicherten Transaktionen recht viel Speicherplatz beansprucht werden.• Aufgaben, die mit Tabellen konfiguriert sind, die in Amazon Redshift geladen werden. Diese Konfiguration ist jedoch kein Problem, wenn Amazon Aurora das Ziel ist. <p>In den meisten Fällen reicht die standardmäßig zugewiesene Speichermenge aus. Es ist jedoch immer eine gute Idee, auf speicherbezogene Metriken zu achten. Achten Sie darauf, Ihren Speicherplatz hochzuskalieren, wenn Sie feststellen, dass Sie mehr als die Standardzuweisung verbrauchen.</p>

Option	Aktion
Netzwerktyp	<p>DMS unterstützt den Netzwerktyp IPv4-Adressierungsprotokoll und unterstützt sowohl IPv4- als auch IPv6-Adressierungsprotokoll-Netzwerktypen im Dual-Stack-Modus. Wenn Sie über Ressourcen verfügen, die über ein Netzwerk mit IPv6-Adressierung mit Ihrer Replikations-Instance kommunizieren müssen, verwenden Sie den Dual-Stack-Modus. Informationen zu Einschränkungen im Dual-Stack-Modus finden Sie unter Einschränkungen für Dual-Stack-Netzwerk-DB-Instances im Benutzerhandbuch zum Amazon Relational Database Service.</p>
VPC	<p>Wählen Sie die VPC aus, die Sie verwenden möchten. Wenn sich die Quell- oder Zieldatenbank innerhalb einer VPC befindet, wählen Sie diese VPC aus. Wenn sich Ihre Quell- und Zieldatenbank in verschiedenen VPCs befinden, stellen Sie sicher, dass sie sich in öffentlichen Subnetzen befinden und öffentlich zugänglich sind. Wählen Sie dann die VPC aus, in der sich die Replikations-Instance befinden soll. Die Replikations-Instance muss auf die Daten in der Quell-VPC zugreifen können. Wenn sich weder die Quell- noch die Zieldatenbank in einer VPC befinden, wählen Sie eine VPC aus, in der die Replikations-Instance abgelegt werden soll.</p>
Replication Subnet Group	<p>Wählen Sie die Replikationssubnetzgruppe in der von Ihnen ausgewählten VPC aus, in der die Replikations-Instance erstellt werden soll. Wenn sich Ihre Quelldatenbank in einer VPC befindet, wählen Sie die Subnetzgruppe, die die Quelldatenbank enthält, als Speicherort für Ihre Replikations-Instance aus. Weitere Informationen zu Replikationssubnetzgruppen finden Sie unter Erstellen einer Replikationssubnetzgruppe.</p>

Option	Aktion
Öffentlich zugänglich	Wählen Sie diese Option, wenn Sie möchten, dass die Replikations-Instance über das Internet zugänglich ist. Die Standardeinstellung ist „öffentlich zugänglich“. Sobald die Option ausgewählt wurde, können Sie sie nicht mehr ändern, nachdem Sie die Replikations-Instance erstellt haben.

3. Wählen Sie die Registerkarte Advanced (Erweitert) aus, um ggf. Werte für Netzwerk- und Verschlüsselungseinstellungen festzulegen. In der folgenden Tabelle sind die Einstellungen beschrieben.

Option	Aktion
Availability Zone	Wählen Sie die Availability Zone aus, in der sich Ihre Quelldatenbank befindet.
VPC Security group(s) (VPC-Sicherheitsgruppe(n))	Die Replikations-Instance wird in einer VPC erstellt. Wenn sich Ihre Quelldatenbank in einer VPC befindet, wählen Sie die VPC-Sicherheitsgruppe aus, die Zugriff auf die DB-Instance bietet, in der sich die Datenbank befindet.
KMS-Schlüssel	Wählen Sie den Verschlüsselungsschlüssel aus, der zum Verschlüsseln der Replikationsspeicher- und Verbindungsinformationen verwendet werden soll. Wenn Sie (Standard) aws/dms wählen, wird der mit Ihrem Konto und AWS Ihrer Region verknüpfte Standardschlüssel AWS Key Management Service (AWS KMS) verwendet. Eine Beschreibung und die Kontonummer werden angezeigt, zusammen mit den Schlüssel-ARN. Weitere Informationen zur Verwendung des Verschlüsselungsschlüssels finden Sie unter Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen .

4. Geben Sie die Maintenance-Einstellungen an. In der folgenden Tabelle sind die Einstellungen beschrieben. Weitere Informationen zu den Wartungseinstellungen finden Sie unter [Arbeiten mit dem AWS DMS -Wartungszeitfenster](#).

Option	Aktion
Automatisches Versions-Upgrade	<p>AWS DMS unterscheidet nicht zwischen Haupt- und Nebenversionen. Ein Upgrade von Version 3.4.x zu 3.5.x wird beispielsweise nicht als Hauptversions-Upgrade betrachtet, daher sollten alle Änderungen abwärtskompatibel sein.</p> <p>Wenn Automatisches Versions-Upgrade aktiviert ist, aktualisiert DMS die Version der Replikations-Instance während des Wartungsfensters automatisch, wenn sie veraltet ist.</p> <p>Wenn AutoMinorVersionUpgradeaktiviert, verwendet DMS die aktuelle Standard-Engine-Version, wenn Sie eine Replikationsinstanz erstellen. Wenn Sie beispielsweise die Engine-Version auf eine niedrigere Versionsnummer als die aktuelle Standardversion festlegen, verwendet DMS die Standardversion.</p> <p>Wenn diese Option beim Erstellen einer Replikationsinstanz AutoMinorVersionUpgradenicht aktiviert ist, verwendet DMS die im Parameter Engine-Version angegebene Engine-Version.</p>
Wartungsfenster	<p>Wählen Sie einen wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time) aus.</p> <p>Standard: Ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitblock pro AWS Region ausgewählt wird und an einem zufälligen Wochentag stattfindet.</p>

5. Wählen Sie `Create replication instance` (Replikations-Instance erstellen) aus.

Ändern einer Replikations-Instance

Sie können die Einstellungen für eine Replikations-Instance ändern, um z. B. die Instance-Klasse zu ändern oder den Speicher zu erhöhen.

Wenn Sie eine Replikations-Instance ändern, können Sie die Änderungen sofort anwenden. Damit Änderungen sofort übernommen werden, wählen Sie die Option `Sofort anwenden` in der AWS Management Console aus. Oder verwenden Sie den `--apply-immediately` Parameter beim Aufrufen von AWS CLI, oder setzen Sie den `ApplyImmediately` Parameter auf, `true` wenn Sie die DMS-API verwenden.

Wenn Sie sich entscheiden, die Änderungen nicht sofort zu übernehmen, werden die Änderungen in die Warteschlange für ausstehende Änderungen aufgenommen. Während des nächsten Wartungsfensters, werden alle ausstehenden Änderungen in der Warteschlange angewandt.

Note

Wenn Sie sich entscheiden die Änderungen sofort zu übernehmen, werden alle Änderungen in der auch alle ausstehenden Änderungen in der Warteschlange übernommen. Wenn eine der ausstehenden Änderungen eine Ausfallzeit erfordert, kann die Auswahl von `Apply changes immediately` (Änderungen sofort übernehmen) einen unerwarteten Ausfall verursachen.

Um eine Replikationsinstanz mithilfe der AWS Konsole zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich `Replication instances` (Replikations-Instances) aus.
3. Wählen Sie die Replikations-Instance aus, die Sie ändern möchten. Die folgende Tabelle beschreibt die Änderungen, die Sie vornehmen können.

Option	Aktion
Name	<p>Sie können den Namen der Replikations-Instance ändern. Geben Sie einen Namen für die Replikations-Instance an, der aus 8 bis 16 druckbaren ASCII-Zeichen (ohne /, " und @) besteht. Der Name sollte für Ihr Konto und die gewählte AWS -Region eindeutig sein. Sie können dem Namen Informationen hinzufügen, z. B. die AWS Region und die Aufgabe, die Sie gerade ausführen, mit einbeziehen west2-mysql2mysql-instance1 .</p>
Beschreibung	<p>Überarbeiten oder geben Sie eine kurze Beschreibung für die Replikations-Instance ein.</p>
Instance-Klasse	<p>Sie können die Instance-Klasse ändern. Wählen Sie eine Instance-Klasse mit der benötigten Konfiguration für die Migration aus. Das Ändern der Instance-Klasse bewirkt, dass die Replikations-Instance neu gestartet wird. Dieser Neustart erfolgt während des nächsten Wartungszeitfensters oder sofort, wenn Sie die Option Änderungen sofort anwenden auswählen.</p> <p>Weitere Informationen dazu, wie Sie herausfinden, welche Instance-Klasse sich am besten für Ihre Migration eignet, finden Sie unter Mit einer AWS DMS Replikationsinstanz arbeiten.</p>
Engine-Version	<p>Sie können ein Upgrade für die Engine-Version durchführen, die von der Replikations-Instance verwendet wird. Das Upgraden der Replikations-Engine-Version führt zum Herunterfahren der Replikations-Instance, während sie auf eine neue Version aktualisiert wird.</p>

Option	Aktion
Multi-AZ	Sie können diese Option ändern, um ein Standby-Replikat Ihrer Replikations-Instance in einer anderen Availability Zone für eine Failover-Unterstützung zu erstellen, oder diese Option entfernen. Wenn Sie Change Data Capture (CDC), fortlaufende Replikation, verwenden möchten, sollten Sie diese Option aktivieren.

Option	Aktion
Allocated storage (GiB) (Zugewiesener Speicher)	<p>Speicherplatz wird in erster Linie von Protokolldateien und zwischengespeicherten Transaktionen belegt. Für die zwischengespeicherten Transaktionen wird Speicherplatz nur verwendet, wenn die zwischengespeicherten Transaktionen auf Festplatte geschrieben werden müssen. Daher verwendet AWS DMS keine nennenswerte Menge an Speicherplatz. Es gibt jedoch folgende Ausnahmen:</p> <ul style="list-style-type: none">• Sehr große Tabellen, bei der viele Transaktionen geladen werden. Das Laden einer großen Tabelle kann einige Zeit in Anspruch nehmen, sodass zwischengespeicherte Transaktionen eher während des Ladens einer großen Tabelle auf Festplatte geschrieben werden.• Aufgaben, die so konfiguriert sind, dass sie vor dem Laden zwischengespeicherter Transaktionen angehalten werden. In diesem Fall werden alle Transaktionen zwischengespeichert, bis der vollständige Ladevorgang für alle Tabellen abgeschlossen ist. Bei dieser Konfiguration kann durch die zwischengespeicherten Transaktionen recht viel Speicherplatz beansprucht werden.• Aufgaben, die mit Tabellen konfiguriert sind, die in Amazon Redshift geladen werden. Diese Konfiguration ist jedoch kein Problem, wenn Amazon Aurora das Ziel ist. <p>In den meisten Fällen reicht die standardmäßig zugewiesene Speichermenge aus. Es ist allerdings immer sinnvoll, auf speicherbezogene Metriken zu achten und den Speicherplatz zu erhöhen, wenn Sie feststellen, dass mehr Speicherplatz verbraucht wird, als standardmäßig zugewiesen ist.</p>

Option	Aktion
Netzwerktyp	DMS unterstützt den Netzwerktyp IPv4-Adressierungsprotokoll und unterstützt sowohl IPv4- als auch IPv6-Adressierungsprotokoll-Netzwerktypen im Dual-Stack-Modus. Wenn Sie über Ressourcen verfügen, die über ein Netzwerk mit IPv6-Adressierung mit Ihrer Replikations-Instance kommunizieren müssen, wählen Sie den Dual-Stack-Modus. Informationen zu Einschränkungen im Dual-Stack-Modus finden Sie unter Einschränkungen für Dual-Stack-Netzwerk-DB-Instances im Benutzerhandbuch zum Amazon Relational Database Service .
VPC Security group(s) (VPC-Sicherheitsgruppe(n))	Die Replikations-Instance wird in einer VPC erstellt. Wenn sich Ihre Quelldatenbank in einer VPC befindet, wählen Sie die VPC-Sicherheitsgruppe aus, die Zugriff auf die DB-Instance bietet, in der sich die Datenbank befindet.

Option	Aktion
Automatisches Versions-Upgrade	<p>AWS DMS unterscheidet nicht zwischen Haupt- und Nebenversionen. Ein Upgrade von Version 3.4.x zu 3.5.x wird beispielsweise nicht als Hauptversions-Upgrade betrachtet, daher sollten alle Änderungen abwärtskompatibel sein. Wenn Automatisches Versions-Upgrade aktiviert ist, aktualisiert DMS die Version der Replikations-Instance während des Wartungsfensters automatisch, wenn sie veraltet ist.</p> <p>Wenn Automatisches Versions-Upgrade aktiviert ist, verwendet DMS die aktuelle Standard-Engine-Version, wenn Sie eine Replikations-Instance erstellen. Wenn Sie beispielsweise die Engine-Version auf eine niedrigere Versionsnummer als die aktuelle Standardversion festlegen, verwendet DMS die Standardversion.</p> <p>Wenn Automatische Versions-Upgrade nicht aktiviert ist, wenn Sie eine Replikations-Instance erstellen, verwendet DMS die Engine-Version, die im Parameter Engine-Version angegeben ist.</p>
Wartungsfenster	<p>Wählen Sie einen wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time) aus.</p> <p>Standard: Ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitblock pro AWS Region ausgewählt wird und an einem zufälligen Wochentag stattfindet.</p>

Option	Aktion
Apply changes immediately (Änderungen sofort anwenden)	<p>Wählen Sie diese Option, um alle durchgeführten Änderungen sofort anzuwenden. Abhängig von den gewählten Einstellungen könnte die Auswahl dieser Option zu einem sofortigen Neustart der Replikations-Instance führen.</p> <p>Wenn Sie Verbindung testen wählen, während AWS DMS die Änderungen anwendet, wird eine Fehlermeldung angezeigt. Nachdem die AWS DMS Änderungen an Ihrer Replikationsinstanz vorgenommen wurden, wählen Sie erneut Verbindung testen aus.</p>
Änderungen im nächsten geplanten Wartungszeitfenster anwenden	Wählen Sie diese Option, wenn DMS bis zum nächsten geplanten Wartungszeitfenster warten soll, um Ihre Änderungen zu übernehmen.

Neustarten einer Replikations-Instance.

Sie können eine AWS DMS Replikationsinstanz neu starten, um die Replikations-Engine neu zu starten. Ein Neustart führt zu einem kurzzeitigen Ausfall der Replikations-Instance, bei dem der Instance-Status auf Neustarten gesetzt wird. Wenn die AWS DMS Instanz für Multi-AZ konfiguriert ist, kann der Neustart mit einem Failover durchgeführt werden. Ein AWS DMS Ereignis wird erzeugt, wenn der Neustart abgeschlossen ist.

Wenn es sich bei Ihrer AWS DMS Instance um eine Multi-AZ-Bereitstellung handelt, können Sie beim Neustart einen geplanten Failover von einer AWS Availability Zone in eine andere erzwingen. Wenn Sie einen geplanten Failover Ihrer AWS DMS Instance erzwingen, werden aktive Verbindungen auf der aktuellen Instance AWS DMS geschlossen, bevor automatisch zu einer Standby-Instance in einer anderen Availability Zone gewechselt wird. Ein Neustart mit einem geplanten Failover hilft Ihnen dabei, ein geplantes Failover-Ereignis einer AWS DMS Instanz zu simulieren, z. B. bei der Skalierung der Replikationsinstanzklasse.

Note

Nachdem ein Neustart ein Failover von einer Availability Zone zu einer anderen erzwungen hat, wird die Änderung der Availability Zone möglicherweise für mehrere Minuten nicht widergespiegelt. Diese Verzögerung tritt in der und in Aufrufen der AWS Management Console AND-API auf. AWS CLI AWS DMS

Wenn bei einem Neustart Migrationsaufgaben auf der Replikations-Instance ausgeführt werden, tritt kein Datenverlust auf, aber die Aufgabe wird beendet und der Aufgabenstatus ändert sich zu einem Fehlerstatus.

Wenn sich die Tabellen in der Migrationsaufgabe mitten in einem Massenladevorgang befinden (Volllastphase) und noch nicht gestartet wurden, gehen sie zu einem Fehlerstatus über. Tabellen, die zu diesem Zeitpunkt vollständig sind, bleiben jedoch in einem vollständigen Zustand. Wenn während der Volllastphase ein Neustart stattfindet, empfehlen wir, einen der folgenden Schritte auszuführen.

- Entfernen Sie die Tabellen, die sich in einem vollständigen Zustand befinden, aus der Aufgabe und starten Sie die Aufgabe mit den verbleibenden Tabellen neu.
- Erstellen Sie eine neue Aufgabe mit Tabellen, die sich in einem Fehlerstatus befinden, sowie mit Tabellen, die noch ausstehen.

Wenn sich Tabellen in der Migrationsaufgabe in der laufenden Replikationsphase befinden, wird die Aufgabe nach Abschluss des Neustarts wieder aufgenommen.

Sie können Ihre AWS DMS Replikationsinstanz nicht neu starten, wenn ihr Status nicht im Status Verfügbar ist. Ihre AWS DMS Instance kann aus verschiedenen Gründen nicht verfügbar sein, z. B. aufgrund einer zuvor angeforderten Änderung oder einer Aktion im Wartungsfenster. Die Zeit, die für den Neustart einer AWS DMS Replikationsinstanz benötigt wird, ist in der Regel gering (unter 5 Minuten).

Neustarten einer Replikationsinstanz mithilfe der Konsole AWS

Verwenden Sie die Konsole, um eine Replikationsinstanz neu zu starten. AWS

Um eine Replikationsinstanz mithilfe der AWS Konsole neu zu starten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.

2. Wählen Sie im Navigationsbereich Replication instances (Replikations-Instances) aus.
3. Wählen Sie die Replikations-Instance aus, die Sie neustarten möchten.
4. Wählen Sie Reboot. Das Dialogfeld Replikations-Instance neu starten wird geöffnet.
5. Wählen Sie die Option Neustart mit Failover?, wenn Sie Ihre Replikations-Instance für die Multi-AZ-Bereitstellung konfiguriert haben und möchten, dass ein Failover zu einer anderen AWS - Availability Zone durchgeführt wird.
6. Wählen Sie Reboot.

Neustarten einer Replikations-Instance mithilfe der CLI

Verwenden Sie den AWS CLI [reboot-replication-instance](#) Befehl mit dem folgenden Parameter, um eine Replikationsinstanz neu zu starten:

- `--replication-instance-arn`

Example Beispiel für einen einfachen Neustart

Im folgenden AWS CLI Beispiel wird eine Replikationsinstanz neu gestartet.

```
aws dms reboot-replication-instance \  
--replication-instance-arn arn of my rep instance
```

Example Beispiel für einen einfachen Neustart mit Failover

Im folgenden AWS CLI Beispiel wird eine Replikationsinstanz mit Failover neu gestartet.

```
aws dms reboot-replication-instance \  
--replication-instance-arn arn of my rep instance \  
--force-planned-failover
```

Neustarten einer Replikations-Instance mithilfe der API

Verwenden Sie die AWS DMS [RebootReplicationInstance](#) API-Aktion mit den folgenden Parametern, um eine Replikationsinstanz neu zu starten:

- `ReplicationInstanceArn = arn of my rep instance`

Example Beispiel für einen einfachen Neustart

Im folgenden Code-Beispiel wird eine Replikations-Instance neu gestartet.

```
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 Beispiel für einen einfachen Neustart mit Failover

Das folgende Codebeispiel startet eine Replikationsinstanz neu und führt ein Failover zu einer anderen AWS Availability Zone durch.

```
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
```

Löschen einer Replikations-Instance

Sie können eine AWS DMS Replikationsinstanz löschen, wenn Sie sie nicht mehr verwenden. Wenn Sie Migrationaufgaben haben, die die Replikations-Instance verwenden, müssen Sie die Aufgaben stoppen und löschen, bevor Sie die Replikations-Instance löschen.

Wenn Sie Ihr AWS Konto schließen, werden alle mit Ihrem Konto verknüpften AWS DMS Ressourcen und Konfigurationen nach zwei Tagen gelöscht. Zu diesen Ressourcen gehören alle Replikations-

Instances, Quell- und Ziel-Endpunktkonfiguration, Replikationsaufgaben und SSL-Zertifikate. Wenn Sie sich nach zwei Tagen für eine AWS DMS erneute Nutzung entscheiden, erstellen Sie die benötigten Ressourcen neu.

Wenn Ihre Replikations-Instance alle Kriterien für das Löschen erfüllt und sie über einen längeren Zeitraum im DELETING-Status bleibt, wenden Sie sich an den Support, um das Problem zu beheben.

Löschen einer Replikationsinstanz mithilfe der Konsole AWS

Verwenden Sie die AWS Konsole, um eine Replikationsinstanz zu löschen.

Um eine Replikationsinstanz mithilfe der AWS Konsole zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Replication instances (Replikations-Instances) aus.
3. Wählen Sie die Replikations-Instance aus, die Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Wählen Sie im Dialogfeld (Bestätigung) Delete (Löschen) aus.

Löschen einer Replikations-Instance über die CLI

Verwenden Sie den AWS CLI [delete-replication-instance](#) Befehl mit dem folgenden Parameter, um eine Replikationsinstanz zu löschen:

- `--replication-instance-arn`

Example Beispiel für das Löschen

Im folgenden AWS CLI Beispiel wird eine Replikationsinstanz gelöscht.

```
aws dms delete-replication-instance \  
--replication-instance-arn arn of my rep instance
```

Löschen einer Replikations-Instance über die API

Verwenden Sie die AWS DMS [DeleteReplicationInstance](#) API-Aktion mit den folgenden Parametern, um eine Replikationsinstanz zu löschen:

- `ReplicationInstanceArn` = *arn of my rep instance*

Example Beispiel für das Löschen

Im folgenden Codebeispiel wird eine Replikations-Instance gelöscht.

```
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
```

Arbeiten mit dem AWS DMS -Wartungszeitfenster

Jede AWS DMS Replikationsinstanz hat ein wöchentliches Wartungsfenster, in dem alle verfügbaren Systemänderungen angewendet werden. Sie können sich das Wartungsfenster als Möglichkeit vorstellen, zu kontrollieren, wann Änderungen und Software-Patches auftreten.

Wenn AWS DMS festgestellt wird, dass in einer bestimmten Woche eine Wartung erforderlich ist, erfolgt die Wartung während des 30-minütigen Wartungsfensters, das Sie bei der Erstellung der Replikationsinstanz ausgewählt haben. AWS DMS schließt die meisten Wartungsarbeiten während des 30-minütigen Wartungsfensters ab. Allerdings können größere Änderungen mehr Zeit in Anspruch nehmen.

Auswirkungen der Wartung auf vorhandene Migrationsaufgaben

Wenn eine AWS DMS Migrationsaufgabe auf einer Instanz ausgeführt wird, treten bei der Installation eines Patches die folgenden Ereignisse auf:

- Wenn die Tabellen in der Migrationsaufgabe sich in der laufenden Änderungsphase der Replikation (CDC) befinden, pausiert AWS DMS die Aufgabe für einen Moment, während der Patch angewendet wird. Die Migration wird dann an der Stelle fortgesetzt, an der sie unterbrochen wurde, als der Patch angewendet wurde.

- Wenn AWS DMS eine Tabelle im Rahmen einer Aufgabe zum Migrieren vorhandener Daten oder zum Migrieren vorhandener Daten und Replizieren laufender Änderungen migriert wird, stoppt DMS die Migration für alle Tabellen, die sich in der Volllastphase befinden, während der Patch angewendet wird, und startet sie dann neu. DMS stoppt außerdem alle Tabellen, die sich in der CDC-Phase befinden, und setzt sie fort, während der Patch angewendet wird.

Ändern der Einstellung des Wartungsfensters

Sie können den Zeitrahmen für das Wartungsfenster mithilfe der AWS Management Console, oder der AWS CLI API ändern. AWS DMS

Ändern der Einstellung des Wartungszeitfensters mit der Konsole

Sie können den Zeitraum des Wartungsfensters über die AWS Management Console ändern.

So ändern Sie das bevorzugte Wartungsfenster mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Replication instances (Replikations-Instances) aus.
3. Wählen Sie die Replikations-Instance aus, die Sie ändern möchten, und wählen Sie Modify aus.
4. Erweitern Sie den Abschnitt Wartung und wählen Sie Datum und Uhrzeit Ihres Wartungsfensters aus.
5. Wählen Sie Apply changes immediately aus.
6. Wählen Sie Ändern aus.

Ändern der Einstellung des Wartungsfensters mithilfe der CLI

Verwenden Sie den AWS CLI [modify-replication-instance](#) Befehl mit den folgenden Parametern, um das bevorzugte Wartungsfenster anzupassen.

- `--replication-instance-identifier`
- `--preferred-maintenance-window`

Example

Im folgenden AWS CLI Beispiel wird das Wartungsfenster auf dienstags von 4:00 bis 4:30 Uhr festgelegt. festgelegt.

```
aws dms modify-replication-instance \  
--replication-instance-identifier myrepliance \  
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Ändern der Einstellung des Wartungsfensters mithilfe der API

Verwenden Sie die AWS DMS [ModifyReplicationInstance](#) API-Aktion mit den folgenden Parametern, um das bevorzugte Wartungsfenster anzupassen.

- `ReplicationInstanceIdentifier` = *myrepliance*
- `PreferredMaintenanceWindow` = *Tue:04:00-Tue:04:30*

Example

Im folgenden Codebeispiel wird das Wartungszeitfenster auf dienstags von 04:00 Uhr bis 04:30 Uhr gesetzt. festgelegt.

```
https://dms.us-west-2.amazonaws.com/  
?Action=ModifyReplicationInstance  
&DBInstanceIdentifier=myrepliance  
&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
```

Arbeiten mit AWS-DMS-Endpunkten

Ein Endpunkt stellt eine Verbindung, einen Datenspeichertyp sowie Standortinformationen zu Ihrem Datenspeicher bereit. AWS Database Migration Service verwendet diese Informationen, um eine Verbindung mit einem Datenspeicher herzustellen und Daten von einem Quellendpunkt zu einem Zielendpunkt zu migrieren. Mithilfe der Endpunkteinstellungen können Sie zusätzliche Verbindungsattribute für einen Endpunkt angeben. Diese Attribute können die Protokollierung, die Dateigröße und andere Parameter steuern. Weitere Informationen zu Endpunkteinstellungen finden Sie im Dokumentationsabschnitt zu Ihrem Datenspeicher.

In den folgenden Abschnitten erhalten Sie weitere Informationen zu Endpunkten.

Themen

- [Erstellen der Quell- und Zielendpunkte](#)
- [Quellen für die Datenmigration](#)
- [Ziele für die Datenmigration](#)
- [Konfigurieren von VPC-Endpunkten als Quell- und Zielendpunkte in AWS DMS](#)
- [Von AWS DMS unterstützte DDL-Anweisungen](#)

Erstellen der Quell- und Zielendpunkte

Sie können Quell- und Zielendpunkte beim Anlegen Ihrer Replikations-Instance oder nach dem Anlegen Ihrer Replikations-Instance erstellen. Die Quell- und Zieldatenspeicher können sich auf einer Amazon EC2-Instance (Amazon Elastic Compute Cloud), einer Amazon RDS-DB-Instance (Amazon Relational Database Service) oder einer lokalen Datenbank befinden. (Beachten Sie, dass sich einer Ihrer Endpunkte in einem AWS-Service befinden muss. Sie können AWS DMS nicht verwenden, um von einer On-Premises-Datenbank zu einer anderen On-Premises-Datenbank zu migrieren.)

Im folgenden Verfahren wird davon ausgegangen, dass Sie den Assistenten der AWS-DMS-Konsole gewählt haben. Sie können diesen Schritt auch ausführen, indem Sie Endpunkte im Navigationsbereich der AWS-Konsole und anschließend Endpunkt erstellen auswählen. Wenn Sie den Assistenten der Konsole verwenden, werden die Quell- und Zielendpunkte auf der gleichen Seite erstellt. Wenn Sie den Assistenten nicht verwenden, erstellen Sie die einzelnen Endpunkte separat voneinander.

So geben Sie die Endpunkte der Quell- oder Zieldatenbank bei Verwendung der AWS-Konsole an

1. Geben Sie auf der Seite Connect source and target database endpoints die Verbindungsinformationen für die Quell- oder Zieldatenbank an. In der folgenden Tabelle sind die Einstellungen beschrieben.

Option	Vorgehensweise
Endpunkttyp	Wählen Sie aus, ob dieser Endpunkt der Quell- oder Zielendpunkt ist.
Auswahl der RDS DB-Instance	Wählen Sie diese Option, wenn der Endpunkt eine Amazon RDS-DB-Instance ist.
Endpoint identifizier	Geben Sie den Namen des Endpunkts an. Sie können zum Namen den Endpunkttyp hinzufügen, z. B. oracle-source oder PostgreSQL-target . Der Name muss für alle Replikations-Instances eindeutig sein.
Source engine und Target engine	Wählen Sie den Typ der Datenbank-Engine aus, die den Endpunkt darstellt.

Option	Vorgehensweise
Zugriff auf die Endpunktdatenbank	<p>Wählen Sie die Option aus, die Sie verwenden möchten, um die Anmeldeinformationen für die Endpunktdatenbank anzugeben:</p> <ul style="list-style-type: none"> • Auswahl von AWS Secrets Manager – Verwenden Sie die in AWS Secrets Manager definierten Secrets, um Ihre Anmeldeinformationen verborgen bereitzustellen, wie im Folgenden gezeigt. Weitere Informationen zum Erstellen dieser Secrets und zu den geheimen Zugriffsrollen, die AWS DMS den Zugriff auf sie ermöglichen, finden Sie unter Verwenden von Secrets für den Zugriff auf AWS Database Migration Service -Endpunkte. • Manuelles Bereitstellen von Zugangsinformationen – Verwenden Sie Klartext-Anmeldeinformationen, die Sie direkt eingeben, wie im Folgenden gezeigt.
Auswahl von AWS Secrets Manager	Stellen Sie die folgenden geheimen Anmeldeinformationen ein.
Secret-ID	Geben Sie den vollständigen Amazon-Ressourcennamen (ARN), einen Teil des ARN oder den Anzeigennamen eines Secrets ein, das Sie in AWS Secrets Manager für den Endpunktdatenbank-Zugriff erstellt haben.
IAM-Rolle	Geben Sie den ARN einer geheimen Zugriffsrolle ein, die Sie in IAM erstellt haben, um in Ihrem Namen AWS DMS Zugriff auf das durch die Secret-ID identifizierte Secret zu gewähren. Weitere Informationen zum Erstellen einer geheimen Zugriffsrolle finden Sie unter Verwenden von Secrets für den Zugriff auf AWS Database Migration Service -Endpunkte .

Option	Vorgehensweise
Secret-ID für Oracle Automatic Storage Management (ASM)	(Nur für Oracle-Quellendpunkte, die Oracle ASM verwenden) Geben Sie den vollständigen Amazon-Ressourcennamen (ARN), einen Teil des ARN oder den Anzeigenamen eines Secrets ein, das Sie in AWS Secrets Manager für den Zugriff auf Oracle ASM erstellt haben. Dieses Secret wird in der Regel für den Zugriff auf Oracle ASM auf demselben Server erstellt wie der geheime Schlüssel, der durch die Secret-ID identifiziert wird.
IAM-Rolle für Oracle ASM	(Nur für Oracle-Quellendpunkte, die Oracle ASM verwenden) Geben Sie den ARN einer geheimen Zugriffsrolle ein, die Sie in IAM erstellt haben, um in Ihrem Namen AWS DMS Zugriff auf die Secret-ID für Oracle Automatic Storage Management (ASM) zu gewähren.
Manuelles Bereitstellen von Zugriffsinformationen	Stellen Sie die folgenden Klartext-Anmeldeinformationen ein.
Server name	Geben Sie den Servernamen ein. Für eine lokale Datenbank kann dies die IP-Adresse oder der öffentliche Hostname sein. Für eine Amazon RDS-DB-Instance kann dies der Endpunkt (auch als DNS-Name bezeichnet) für die DB-Instance sein, z. B. mysqlsrvinst.abcd12345678.us-west-2.rds.amazonaws.com .
Port	Geben Sie den Port an, der von der Datenbank verwendet wird.
Secure Socket Layer (SSL)-Modus	Wählen Sie einen SSL-Modus aus, wenn Sie die Verbindungsverschlüsselung für diesen Endpunkt aktivieren möchten. Je nach ausgewähltem Modus müssen Sie möglicherweise Zertifikat- und Serverzertifikatinformationen angeben.

Option	Vorgehensweise
Benutzername	Geben Sie den Benutzernamen mit den für die Datenmigration erforderlichen Berechtigungen ein. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie im Abschnitt zur Sicherheit für die Quell- oder Zieldatenbank-Engine in diesem Benutzerhandbuch.
Passwort	Geben Sie das Passwort für das Konto mit den erforderlichen Berechtigungen ein. Für Passwörter für AWS DMS-Quell- und Zielendpunkte gelten je nach Datenbank-Engine Zeichenbeschränkungen. Weitere Informationen können Sie der folgenden Tabelle entnehmen.
Datenbankname	Bei bestimmten Datenbank-Engines der Name der Datenbank, die Sie als Endpunktdatenbank verwenden möchten.

In der folgenden Tabelle sind die nicht unterstützten Zeichen in Endpunktpasswörtern und Secret-Manager-Secrets für die aufgelisteten Datenbank-Engines aufgeführt. Wenn Sie Kommata (,) in Ihren Endpunktkennwörtern verwenden möchten, verwenden Sie die Secrets-Manager-Unterstützung unter AWS DMS, um den Zugriff auf Ihre AWS DMS-Instances zu authentifizieren. Weitere Informationen finden Sie unter [Verwenden von Secrets für den Zugriff auf AWS Database Migration Service -Endpunkte](#).

Für diese Datenbank-Engine	Die folgenden Zeichen werden in einem Endpunktpasswort und Secret Manager-Secrets nicht unterstützt
Alle	{ }
Microsoft Azure, nur als Quelle	;
Microsoft SQL Server	, ;

Für diese Datenbank-Engine	Die folgenden Zeichen werden in einem Endpunktasswort und Secret Manager-Secrets nicht unterstützt
MySQL-kompatibel, einschließlich MySQL, MariaDB und Amazon Aurora MySQL	;
Oracle	,
PostgreSQL, Amazon Aurora PostgreSQL-kompatible Edition und Amazon Aurora Serverless als Ziel nur für Aurora PostgreSQL-kompatible Edition	; + %
Amazon Redshift, nur als Ziel	, ;

- Wählen Sie Endpunkteinstellungen und AWS KMS key, falls benötigt. Sie können die Endpunktverbindung testen, indem Sie Run test wählen. In der folgenden Tabelle sind die Einstellungen beschrieben.

Option	Vorgehensweise
Endpunkteinstellungen	<p>Wählen Sie hier eventuelle zusätzliche Verbindungsparameter aus. Weitere Informationen zu Endpunkteinstellungen finden Sie im Dokumentationsabschnitt für Ihre Quell-Engine oder Ziel-Engine (angegeben in Schritt 1).</p> <p>Wenn Sie bei einem Oracle-Quellendpunkt, der Oracle ASM verwendet, in Schritt 1 die Option Zugriffsinformationen manuell bereitstellen wählen, müssen Sie möglicherweise auch die Endpunkteinstellung eingeben, um die Oracle-ASM-Benutzeranmeldedaten anzugeben. Weitere Informationen zu diesen Oracle-ASM-Endpunkteinstellungen finden Sie unter Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC.</p>

Option	Vorgehensweise
AWS KMS key	Wählen Sie den Verschlüsselungsschlüssel aus, der zum Verschlüsseln der Replikationsspeicher- und Verbindungsinformationen verwendet werden soll. Wenn Sie (Standard) aws/dms auswählen, wird der Standard-AWS-Key-Management-Service (AWS KMS)-Schlüssel, der mit Ihrem Konto und Ihrer AWS-Region verbunden ist, verwendet. Weitere Informationen zur Verwendung des Verschlüsselungsschlüssels finden Sie unter Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen .
Testen der Endpunktverbindung (optional)	Hinzufügen der VPC und des Replikations-Instanznamens. Wählen Sie zum Testen der Verbindung Run test (Test ausführen) aus.

Quellen für die Datenmigration

AWS Database Migration Service (AWS DMS) kann viele der gängigen Datenbank-Engines als Quelle für die Datenreplikation verwenden. Bei der Datenbankquelle kann es sich um eine selbstverwaltete Engine, die auf einer Amazon-EC2-Instance ausgeführt wird, oder um eine On-Premises-Datenbank handeln. Alternativ kann sich die Datenquelle auch in einem AWS-Service, wie z. B. Amazon RDS oder Amazon S3, befinden.

Eine umfassende Liste gültiger Quellen finden Sie unter [Quellen für AWS DMS](#).

Themen

- [Verwenden einer Oracle-Datenbank als Quelle für AWS DMS](#)
- [Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS](#)
- [Verwenden einer Microsoft Azure SQL-Datenbank als Quelle für AWS DMS](#)
- [Verwenden einer Microsoft-Azure-SQL-Managed-Instance als Quelle für AWS DMS](#)
- [Verwendung des flexiblen Microsoft-Azure-Database-für-PostgreSQL-Servers als Quelle für AWS DMS](#)

- [Verwenden eines flexiblen Microsoft-Azure-Datenbank-für-MySQL-Servers als Quelle für AWS DMS](#)
- [Verwendung von OCI MySQL Heatwave als Quelle für AWS DMS](#)
- [Verwenden von Google Cloud for MySQL als Quelle für AWS DMS](#)
- [Verwenden von Google Cloud für PostgreSQL als Quelle für AWS DMS](#)
- [Verwenden einer PostgreSQL-Datenbank als AWS DMS-Quelle](#)
- [Verwenden einer MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#)
- [Verwenden einer SAP ASE-Datenbank als Quelle für AWS DMS](#)
- [Verwenden von MongoDB als Quelle für AWS DMS](#)
- [Verwenden von Amazon DocumentDB \(mit MongoDB-Kompatibilität\) als Quelle für AWS DMS](#)
- [Verwenden von Amazon S3 als Quelle für AWS DMS](#)
- [Verwendung der IBM Db2 für Linux-, Unix-, Windows- und Amazon RDS-Datenbank \(Db2 LUW\) als Quelle für AWS DMS](#)
- [Verwendung von Datenbanken in IBM Db2 für z/OS als Quelle für AWS DMS](#)

Verwenden einer Oracle-Datenbank als Quelle für AWS DMS

Sie können Daten aus einer oder mehreren Oracle-Datenbanken migrieren mit AWS DMS. Mit einer Oracle-Datenbank als Quelle können Sie Daten zu allen von AWS DMS unterstützten Zielen migrieren.

AWS DMS unterstützt die folgenden Oracle-Datenbankeditionen:

- Oracle Enterprise Edition
- Oracle Standard Edition
- Oracle Express Edition
- Oracle Personal Edition

Hinweise zu Versionen von Oracle-Datenbanken, die als Quelle AWS DMS unterstützt werden, finden Sie unter [Quellen für AWS DMS](#).

Sie können Secure Sockets Layer (SSL) verwenden, um Verbindungen zwischen Ihrem Oracle-Endpunkt und der Replikations-Instance zu verschlüsseln. Weitere Informationen zur Verwendung von SSL mit einem Oracle-Endpunkt finden Sie unter [SSL-Unterstützung für einen Oracle-Endpunkt](#).

AWS DMS unterstützt die Verwendung von Oracle Transparent Data Encryption (TDE) zur Verschlüsselung ruhender Daten in der Quelldatenbank. Weitere Informationen zur Verwendung von Oracle TDE mit einem Oracle-Quellendpunkt finden Sie unter [Unterstützte Verschlüsselungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS](#).

AWS unterstützt die Verwendung von TLS Version 1.2 und höher mit Oracle-Endpunkten (und allen anderen Endpunkttypen) und empfiehlt die Verwendung von TLS Version 1.3 oder höher.

Gehen Sie wie folgt vor, um eine Oracle-Datenbank als AWS DMS Quellendpunkt zu konfigurieren:

1. Erstellen Sie einen Oracle-Benutzer mit den entsprechenden Berechtigungen für den AWS DMS Zugriff auf Ihre Oracle-Quelldatenbank.
2. Erstellen Sie einen Oracle-Quellendpunkt, der der von Ihnen ausgewählten Oracle-Datenbankkonfiguration entspricht. Um eine full-load-only Aufgabe zu erstellen, ist keine weitere Konfiguration erforderlich.
3. Um eine Aufgabe zu erstellen, die die Erfassung von Änderungsdaten übernimmt (eine reine CDC-Aufgabe oder eine Volllast- und CDC-Aufgabe), wählen Sie Oracle LogMiner oder AWS DMS Binary Reader, um Datenänderungen zu erfassen. Die Auswahl von LogMiner oder Binary Reader bestimmt einige der späteren Berechtigungen und Konfigurationsoptionen. Einen Vergleich zwischen Binary Reader LogMiner und Binary Reader finden Sie im folgenden Abschnitt.

Note

Weitere Informationen zu Volllastaufgaben, reinen CDC-Aufgaben sowie Volllast-und-CDC-Aufgaben finden Sie unter [Erstellen einer Aufgabe](#).

Weitere Informationen zur Arbeit mit Oracle-Quelldatenbanken und AWS DMS finden Sie in den folgenden Abschnitten.

Themen

- [Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC](#)
- [Workflows für die Konfiguration einer selbstverwalteten oder AWS-verwalteten Oracle-Quelldatenbank für AWS DMS Konfigurieren einer Oracle-Quelldatenbank](#)
- [Arbeiten mit einer selbstverwalteten Oracle-Datenbank als Quelle für AWS DMS](#)
- [Arbeiten mit einer AWS-verwalteten Oracle-Datenbank als Quelle für AWS DMS](#)


- [Einschränkungen bei der Verwendung von Oracle als Quelle für AWS DMS](#)
- [SSL-Unterstützung für einen Oracle-Endpunkt](#)
- [Unterstützte Verschlüsselungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS](#)
- [Unterstützte Komprimierungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS](#)
- [Replizieren verschachtelter Tabellen unter Verwendung von Oracle als Quelle für AWS DMS](#)
- [REDO wird auf Oracle ASM gespeichert, wenn Oracle als Quelle für verwendet wird AWS DMS](#)
- [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#)
- [Quelldatentypen für Oracle](#)

Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC

In gibt es zwei Methoden zum Lesen der Redo-Logs AWS DMS, wenn Change Data Capture (CDC) für Oracle als Quelle ausgeführt wird: Oracle LogMiner und AWS DMS Binary Reader. LogMiner ist eine Oracle-API zum Lesen der Online-Redo-Logs und archivierten Redo-Log-Dateien. Binary Reader ist eine AWS DMS Methode, die die rohen Redo-Log-Dateien direkt liest und analysiert. Diese Methoden weisen die folgenden Merkmale auf.

Funktion	LogMiner	Binary Reader
Einfache Konfiguration	Ja	Nein
Geringere Auswirkungen auf E/A und CPU des Quellsystems	Nein	Ja
Bessere CDC-Leistung	Nein	Ja
Unterstützung für Oracle-Tabellen-Cluster	Ja	Nein
Unterstützung für alle Typen von Oracle Hybrid Columnar Compression (HCC)	Ja	Teilweise Binary Reader unterstützt QUERY LOW nicht für Aufgaben mit CDC. Alle anderen HCC-

Funktion	LogMiner	Binary Reader
		Typen werden vollständig unterstützt.
Unterstützung für LOB-Spalten nur in Oracle 12c	Nein (LOB-Support ist LogMiner in Oracle 12c nicht verfügbar.)	Ja
Unterstützung für UPDATE-Anweisungen, die sich nur auf LOB-Spalten auswirken	Nein	Ja
Unterstützung für Oracle Transparent Data Encryption (TDE)	Teilweise Bei Verwendung von Oracle LogMiner wird die TDE-Verschlüsselung auf Spaltenebene für Amazon RDS for Oracle AWS DMS nicht unterstützt.	Teilweise Binary Reader unterstützt TDE nur für selbstverwaltete Oracle-Datenbanken.
Unterstützung für alle Oracle-Komprimierungsmethoden	Ja	Nein
Unterstützung für XA-Transaktionen	Nein	Ja
RAC	Ja	Ja
	Nicht empfohlen	Dringend empfohlen

 Note

AWS DMS Verwendet standardmäßig Oracle LogMiner for (CDC).

AWS DMS unterstützt Methoden zur transparenten Datenverschlüsselung (TDE) bei der Arbeit mit einer Oracle-Quelldatenbank. Wenn die von Ihnen angegebenen TDE-Anmeldeinformationen falsch sind, schlägt die AWS DMS Migrationsaufgabe nicht fehl, was sich auf die laufende Replikation verschlüsselter Tabellen auswirken kann. Weitere Informationen zum Angeben der TDE-Anmeldeinformationen finden Sie unter [Unterstützte Verschlüsselungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS](#).

Zu den Hauptvorteilen der Verwendung LogMiner von AWS DMS gehören die folgenden:

- LogMiner unterstützt die meisten Oracle-Optionen, wie Verschlüsselungs- und Komprimierungsoptionen. Binary Reader unterstützt nicht alle Oracle-Optionen, insbesondere die Komprimierung und die meisten Optionen für die Verschlüsselung.
- LogMiner bietet eine einfachere Konfiguration, insbesondere im Vergleich zur Einrichtung mit direktem Zugriff auf Binary Reader oder wenn die Redo-Logs mit Oracle Automatic Storage Management (ASM) verwaltet werden.
- LogMiner unterstützt Tabellencluster zur Verwendung durch. AWS DMS Bei Binary Reader ist das nicht der Fall.

Zu den Hauptvorteilen der Verwendung von Binary Reader mit AWS DMS gehören:

- Bei Migrationen mit einer großen Anzahl von Änderungen LogMiner kann dies einige I/O- oder CPU-Auswirkungen auf den Computer haben, der die Oracle-Quelldatenbank hostet. Bei Binary Reader ist die Wahrscheinlichkeit geringer, dass E/A- oder CPU-Auswirkungen auftreten, da die Protokolle direkt durchsucht werden, anstatt mehrere Datenbankabfragen durchzuführen.
- Bei Migrationen mit einem hohen Änderungsvolumen ist die CDC-Leistung bei Verwendung von Binary Reader in der Regel viel besser als bei Oracle. LogMiner
- Binary Reader unterstützt CDC für LOBs in Oracle Version 12c. LogMiner tut es nicht.

Verwenden Sie im Allgemeinen Oracle LogMiner für die Migration Ihrer Oracle-Datenbank, es sei denn, Sie haben eine der folgenden Situationen:

- Sie müssen mehrere Migrationsaufgaben auf der Oracle-Quelldatenbank ausführen.
- Der Umfang der Änderungen oder des Redo-Protokolls in der Oracle-Quelldatenbank ist hoch oder Sie haben Änderungen und verwenden auch Oracle ASM.

Note

Wenn Sie zwischen der Verwendung von Oracle LogMiner und AWS DMS Binary Reader wechseln, stellen Sie sicher, dass Sie die CDC-Aufgabe neu starten.

Konfiguration für CDC für eine Oracle-Quelldatenbank

Damit ein Oracle-Quellendpunkt für eine Aufgabe zur Erfassung von Datenänderungen (Change Data Capture, CDC) eine Verbindung zur Datenbank herstellen kann, müssen Sie möglicherweise zusätzliche Verbindungsattribute angeben. Dies kann entweder für eine Volllast-und-CDC- oder für eine reine CDC-Aufgabe gelten. Die zusätzlichen Verbindungsattribute, die Sie angeben, hängen von der Methode ab, die Sie für den Zugriff auf die Redo-Logs verwenden: Oracle LogMiner oder AWS DMS Binary Reader.

Sie geben die zusätzlichen Verbindungsattribute an, wenn Sie einen Quellendpunkt erstellen. Wenn Sie mehrere Verbindungsattributeinstellungen verwenden, trennen Sie diese durch Semikola und ohne zusätzliche Leerräume voneinander (z. B. `oneSetting;thenAnother`).

AWS DMS verwendet LogMiner standardmäßig. Sie müssen keine zusätzlichen Verbindungsattribute angeben, um dieses Hilfsprogramm zu verwenden.

Um Binary Reader für den Zugriff auf die Redo-Protokolle verwenden zu können, müssen Sie die folgenden zusätzlichen Verbindungsattribute hinzufügen.

```
useLogMinerReader=N;useBfile=Y;
```

Verwenden Sie das folgende Format für die zusätzlichen Verbindungsattribute, um auf einen Server zuzugreifen, der ASm mit Binary Reader zuzugreifen.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_number+ASM;
```

Setzen Sie den Password-Anforderungsparameter des Quellendpunkts sowohl auf das Oracle-Benutzerpassword als auch das ASM-Passwort (durch ein Komma getrennt) wie nachfolgend dargestellt:

```
oracle_user_password,asm_user_password
```

Wenn die Oracle-Quelle ASM verwendet, können Sie mit Hochleistungsoptionen in Binary Reader für die Transaktionsverarbeitung in großem Umfang arbeiten. Zu diesen Optionen gehören zusätzliche Verbindungsattribute, um die Anzahl der parallelen Threads (`parallelASMReadThreads`) und die Anzahl der Read-Ahead-Puffer (`readAheadBlocks`) anzugeben. Wenn diese Attribute gemeinsam festgelegt werden, kann dies die Leistung der CDC-Aufgabe erheblich verbessern. Die folgenden Einstellungen liefern gute Ergebnisse für die meisten ASM-Konfigurationen.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_number;
+ASM;
parallelASMReadThreads=6;readAheadBlocks=150000;
```

Weitere Hinweise zu Werten, die zusätzliche Verbindungsattribute unterstützen, finden Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).

Die Leistung einer CDC-Aufgabe mit einer Oracle-Quelle, die ASM verwendet, hängt zudem von anderen Einstellungen ab, die von Ihnen festgelegt werden. Zu diesen Einstellungen gehören Ihre zusätzlichen AWS DMS -Verbindungsattribute sowie die SQL-Einstellungen für die Konfiguration der Oracle-Quelle konfiguriert wurde. Weitere Informationen zu zusätzlichen Verbindungsattributen für eine Oracle-Quelle, die ASM verwendet, finden Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).

Sie müssen außerdem einen geeigneten CDC-Startpunkt wählen. In der Regel möchten Sie dabei den Punkt der Transaktionsverarbeitung identifizieren, an dem die früheste offene Transaktion erfasst wird, von der aus CDC beginnen kann. Andernfalls kann es passieren, dass bei der CDC-Aufgabe frühere offene Transaktionen übersehen werden. Für eine Oracle-Quelldatenbank können Sie auf der Grundlage der Oracle-Systemänderungsnummer (System Change Number, SCN) einen nativen CDC-Startpunkt wählen, um diese früheste offene Transaktion zu identifizieren. Weitere Informationen finden Sie unter [Durchführen der Replikation von einem CDC-Startpunkt aus](#).

Weitere Informationen zum Konfigurieren von CDC für eine selbstverwaltete Oracle-Datenbank als Quelle finden Sie unter [Kontoberechtigungen sind erforderlich, wenn Sie Oracle für den Zugriff LogMiner auf die Redo-Logs verwenden](#), [Kontoberechtigungen sind erforderlich, wenn AWS DMS Binary Reader für den Zugriff auf die Redo-Logs verwendet wird](#) und [Erforderliche zusätzliche Kontoberechtigungen bei Verwendung von Binary Reader mit Oracle ASM](#).

Weitere Hinweise zur Konfiguration von CDC für eine AWS-verwaltete Oracle-Datenbank als Quelle finden Sie unter [Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit](#)

[einer RDS for Oracle-Quelle für AWS DMS](#) und [Verwenden von Amazon RDS Oracle Standby \(Lesereplikat\) als Quelle mit Binary Reader für CDC in AWS DMS](#)

Workflows für die Konfiguration einer selbstverwalteten oder AWS-verwalteten Oracle-Quelldatenbank für AWS DMS

Workflows für die Konfiguration einer selbstverwalteten oder AWS-verwalteten Oracle-Quelldatenbank für AWS DMS

Wenden Sie zum Konfigurieren einer selbstverwalteten Quelldatenbank-Instance die folgenden Workflow-Schritte abhängig davon an, wie Sie CDC ausführen.

Für diesen Workflow-Schritt	Gehen Sie wie folgt vor, wenn Sie CDC verwenden LogMiner	Vorgehensweise bei Ausführung von CDC mit Binary Reader
Gewähren Sie Oracle-Kontoberechtigungen.	Siehe Erforderliche Benutzerkontoberechtigungen für eine selbstverwaltete Oracle-Quelle für AWS DMS .	Siehe Erforderliche Benutzerkontoberechtigungen für eine selbstverwaltete Oracle-Quelle für AWS DMS .
Bereiten Sie die Quelldatenbank für die Replikation mit CDC vor.	Siehe Vorbereiten einer selbstverwalteten Oracle-Quelldatenbank für CDC mit AWS DMS .	Siehe Vorbereiten einer selbstverwalteten Oracle-Quelldatenbank für CDC mit AWS DMS .
Gewähren Sie zusätzliche Oracle-Benutzerberechtigungen, die für CDC erforderlich sind.	Siehe Kontoberechtigungen sind erforderlich, wenn Sie Oracle für den Zugriff LogMiner auf die Redo-Logs verwenden .	Siehe Kontoberechtigungen sind erforderlich, wenn AWS DMS Binary Reader für den Zugriff auf die Redo-Logs verwendet wird .
Gewähren Sie einer Oracle-Instance mit ASM zusätzliche Benutzerkontoberechtigungen, die für den Zugriff auf ASM für CDC erforderlich sind.	Keine zusätzliche Aktion. AWS DMS unterstützt Oracle ASM ohne zusätzliche Kontoberechtigungen.	Siehe Erforderliche zusätzliche Kontoberechtigungen bei Verwendung von Binary Reader mit Oracle ASM .

Für diesen Workflow-Schritt	Gehen Sie wie folgt vor, wenn Sie CDC verwenden LogMiner	Vorgehensweise bei Ausführung von CDC mit Binary Reader
Falls Sie dies noch nicht getan haben, konfigurieren Sie die Aufgabe so, dass sie LogMiner oder Binary Reader for CDC verwendet.	Siehe Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC .	Siehe Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC .
Konfigurieren Sie Oracle Standby als Quelle für CDC.	AWS DMS unterstützt Oracle Standby nicht als Quelle.	Siehe Verwenden von selbstverwaltetem Oracle Standby als Quelle mit Binary Reader für CDC in AWS DMS .

Verwenden Sie die folgenden Workflow-Schritte, um eine von AWS-verwaltete Oracle-Quelldatenbank-Instance zu konfigurieren.

Für diesen Workflow-Schritt	Wenn Sie CDC verwenden LogMiner, gehen Sie wie folgt vor	Vorgehensweise bei Ausführung von CDC mit Binary Reader
Gewähren Sie Oracle-Kontoberechtigungen.	Weitere Informationen finden Sie unter Erforderliche Benutzerkontenberechtigungen für eine AWS-verwaltete Oracle-Quelle für AWS DMS .	Weitere Informationen finden Sie unter Erforderliche Benutzerkontenberechtigungen für eine AWS-verwaltete Oracle-Quelle für AWS DMS .
Bereiten Sie die Quelldatenbank für die Replikation mit CDC vor.	Weitere Informationen finden Sie unter Konfiguration einer AWS-verwalteten Oracle-Quelle für AWS DMS .	Weitere Informationen finden Sie unter Konfiguration einer AWS-verwalteten Oracle-Quelle für AWS DMS .
Gewähren Sie zusätzliche Oracle-Benutzerberechtigungen, die für CDC erforderlich sind.	Es sind keine zusätzlichen Kontoberechtigungen erforderlich.	Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary

Für diesen Workflow-Schritt	Wenn Sie CDC verwenden LogMiner, gehen Sie wie folgt vor	Vorgehensweise bei Ausführung von CDC mit Binary Reader
		Reader mit einer RDS for Oracle-Quelle für AWS DMS.
Falls Sie dies noch nicht getan haben, konfigurieren Sie die Aufgabe so, dass sie LogMiner oder Binary Reader for CDC verwendet.	Weitere Informationen finden Sie unter Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC.	Weitere Informationen finden Sie unter Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC.
Konfigurieren Sie Oracle Standby als Quelle für CDC.	AWS DMS unterstützt Oracle Standby nicht als Quelle.	Weitere Informationen finden Sie unter Verwenden von Amazon RDS Oracle Standby (Lesereplikat) als Quelle mit Binary Reader für CDC in AWS DMS.

Arbeiten mit einer selbstverwalteten Oracle-Datenbank als Quelle für AWS DMS

Eine selbstverwaltete Datenbank ist eine Datenbank, die von Ihnen konfiguriert und kontrolliert wird. Dabei kann es sich um eine On-Premises-Datenbank-Instance oder um eine Datenbank in Amazon EC2 handeln. Im Folgenden erfahren Sie mehr über die Rechte und Konfigurationen, die Sie benötigen, wenn Sie eine selbstverwaltete Oracle-Datenbank mit verwenden. AWS DMS

Erforderliche Benutzerkontenberechtigungen für eine selbstverwaltete Oracle-Quelle für AWS DMS

Um eine Oracle-Datenbank als Quelle in zu verwenden AWS DMS, gewähren Sie dem Oracle-Benutzer, der in den Verbindungseinstellungen für den Oracle-Endpunkt angegeben ist, die folgenden Rechte.

Note

Wenn Sie Berechtigungen erteilen, verwenden Sie den tatsächlichen Namen der Objekte und nicht die Synonyme dafür. Verwenden Sie beispielsweise `V_$$OBJECT` einschließlich des Unterstrichs, nicht `V$OBJECT` ohne Unterstrich.

```
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 Unterstützte
  Verschlüsselungsmethoden für die Verwendung von Oracle als Quelle für 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.
```

Erteilen Sie für jede replizierte Tabelle die folgende zusätzliche Berechtigung, wenn Sie eine spezifische Tabellenliste verwenden.

```
GRANT SELECT on any-replicated-table to db_user;
```

Erteilen Sie die folgende zusätzliche Berechtigung, um LOB-Spalten mit dem Validierungs-Feature zu validieren.

```
GRANT EXECUTE ON SYS.DBMS_CRYPT0 TO db_user;
```

Erteilen Sie die folgenden zusätzlichen Rechte, wenn Sie stattdessen Binary Reader verwenden LogMiner.

```
GRANT SELECT ON SYS.DBA_DIRECTORIES TO db_user;
```

Erteilen Sie die folgende zusätzliche Berechtigung, um Ansichten verfügbar zu machen.

```
GRANT SELECT on ALL_VIEWS to dms_user;
```

Um Ansichten verfügbar zu machen, müssen Sie Ihrem Quellendpunkt außerdem das zusätzliche Verbindungsattribut `exposeViews=true` hinzufügen.

Gewähren Sie die folgende zusätzliche Berechtigung, wenn Sie Serverless-Replikationen verwenden.

```
GRANT SELECT on dba_segments to db_user;
```

Weitere Informationen zu Serverless-Replikationen finden Sie unter [Arbeiten mit AWS DMS Serverless](#).

Erteilen Sie die folgenden zusätzlichen Berechtigungen, wenn Sie Oracle-spezifische Bewertungen vor der Migration verwenden.

```
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;
```

Informationen zu Oracle-spezifischen Bewertungen vor der Migration finden Sie unter [Bewertungen von Oracle](#).

Voraussetzungen für die Verarbeitung offener Transaktionen für Oracle Standby

Wenn Sie AWS DMS Versionen 3.4.6 und höher verwenden, führen Sie die folgenden Schritte aus, um offene Transaktionen für Oracle Standby zu verarbeiten.

1. Erstellen Sie einen Datenbank-Link mit dem Namen AWSDMS_DBLINK in der Primärdatenbank. *DMS_USER* verwendet den Datenbank-Link, um eine Verbindung mit der Primärdatenbank herzustellen. Beachten Sie, dass der Datenbank-Link von der Standby-Instance aus ausgeführt wird, um die offenen Transaktionen abzufragen, die in der Primärdatenbank ausgeführt werden. Sehen Sie sich das folgende -Beispiel an.

```
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. Stellen Sie sicher, dass die Verbindung mit dem Datenbank-Link über *DMS_USER* hergestellt ist, wie im folgenden Beispiel dargestellt.

```
select 1 from dual@AWSDMS_DBLINK
```

Vorbereiten einer selbstverwalteten Oracle-Quelldatenbank für CDC mit AWS DMS

Führen Sie die folgenden Schritte aus, um Ihre selbstverwaltete Oracle-Datenbank als Quelle zum Ausführen einer CDC-Aufgabe vorzubereiten:

- [Es wird überprüft, ob die Version der AWS DMS Quelldatenbank unterstützt wird.](#)

- [Sicherstellen, dass der Modus ARCHIVELOG aktiviert ist.](#)
- [Einrichten der zusätzlichen Protokollierung.](#)

Es wird überprüft, ob die Version der AWS DMS Quelldatenbank unterstützt wird

Führen Sie eine Abfrage wie die folgende aus, um zu überprüfen, ob die aktuelle Version der Oracle-Quelldatenbank von AWS DMS unterstützt wird.

```
SELECT name, value, description FROM v$parameter WHERE name = 'compatible';
```

Hier sind `name`, `value` und `description` Spalten irgendwo in der Datenbank, die auf der Grundlage des Wertes von `name` abgefragt werden. Wenn diese Abfrage ohne Fehler ausgeführt wird, wird die aktuelle Version der Datenbank AWS DMS unterstützt und Sie können mit der Migration fortfahren. Wenn die Abfrage einen Fehler AWS DMS auslöst, wird die aktuelle Version der Datenbank nicht unterstützt. Um mit der Migration fortzufahren, konvertieren Sie zunächst die Oracle-Datenbank in eine Version, die von unterstützt wird AWS DMS.

Sicherstellen, dass der Modus ARCHIVELOG aktiviert ist

Sie können Oracle in zwei verschiedenen Modi ausführen: dem ARCHIVELOG-Modus und dem NOARCHIVELOG-Modus. Führen Sie die Datenbank im Modus ARCHIVELOG aus, um eine CDC-Aufgabe auszuführen. Führen Sie die folgende Abfrage aus, um zu ermitteln, ob sich die Datenbank im Modus ARCHIVELOG befindet.

```
SQL> SELECT log_mode FROM v$database;
```

Wenn der Modus NOARCHIVELOG zurückgegeben wird, setzen Sie die Datenbank entsprechend den Oracle-Anweisungen auf ARCHIVELOG.

Einrichten der zusätzlichen Protokollierung

Um laufende Änderungen zu erfassen, AWS DMS müssen Sie eine minimale zusätzliche Protokollierung in Ihrer Oracle-Quelldatenbank aktivieren. Darüber hinaus müssen Sie die zusätzliche Protokollierung für jede replizierte Tabelle in der Datenbank aktivieren.

AWS DMS Fügt standardmäßig PRIMARY KEY zusätzliche Protokollierung für alle replizierten Tabellen hinzu. Um PRIMARY KEY zusätzliche Protokollierung hinzufügen AWS DMS zu können, gewähren Sie für jede replizierte Tabelle die folgende Berechtigung.

```
ALTER on any-replicated-table;
```

Sie können die hinzugefügte standardmäßige PRIMARY KEY zusätzliche Protokollierung deaktivieren, indem Sie das zusätzliche AWS DMS Verbindungsattribut verwenden.

`addSupplementalLogging` Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).

Stellen Sie sicher, dass Sie die zusätzliche Protokollierung aktivieren, wenn Ihre Replikationsaufgabe eine Tabelle mithilfe einer WHERE-Klausel aktualisiert, die nicht auf eine Primärschlüsselspalte verweist.

So richten Sie die zusätzliche Protokollierung manuell ein

1. Führen Sie die folgende Abfrage aus, um zu überprüfen, ob bereits die zusätzliche Protokollierung für die Datenbank aktiviert ist.

```
SELECT supplemental_log_data_min FROM v$database;
```

Ist das zurückgegebene Ergebnis YES oder IMPLICIT, ist die zusätzliche Protokollierung für die Datenbank aktiviert.

Wenn dies nicht der Fall ist, aktivieren Sie die zusätzliche Protokollierung für die Datenbank, indem Sie den folgenden Befehl ausführen.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

2. Stellen Sie sicher, dass die erforderliche zusätzliche Protokollierung für jede replizierte Tabelle hinzugefügt wird.

Berücksichtigen Sie dabei Folgendes:

- Wenn der Tabelle die zusätzliche Protokollierung ALL COLUMNS hinzugefügt wird, müssen Sie keine weitere Protokollierung hinzufügen.
- Wenn ein Primärschlüssel vorhanden ist, fügen Sie die zusätzliche Protokollierung für den Primärschlüssel hinzu. Dazu können Sie entweder das Format zum Hinzufügen der zusätzlichen Protokollierung zum Primärschlüssel selbst verwenden oder die zusätzliche Protokollierung auf den Primärschlüsselspalten der Datenbank hinzufügen.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
```

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
```

- Wenn kein Primärschlüssel vorhanden ist und die Tabelle über einen einzigen eindeutigen Index verfügt, müssen alle Spalten des eindeutigen Index dem zusätzlichen Protokoll hinzugefügt werden.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG GROUP LogGroupName  
(UniqueIndexColumn1 [, UniqueIndexColumn2] ...) ALWAYS;
```

Die Verwendung von SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS fügt die Spalten des eindeutigen Index dem Protokoll nicht hinzu.

- Wenn kein Primärschlüssel vorhanden ist und die Tabelle mehrere eindeutige Indizes hat, AWS DMS wird der erste eindeutige Index in einer alphabetisch aufsteigenden Liste ausgewählt. Sie müssen die zusätzliche Protokollierung für die Spalten des ausgewählten Index wie im vorherigen Punkt hinzufügen.

Die Verwendung von SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS fügt die Spalten des eindeutigen Index dem Protokoll nicht hinzu.

- Wenn kein Primärschlüssel vorhanden ist und kein eindeutiger Index vorhanden ist, fügen Sie die zusätzliche Protokollierung für alle Spalten hinzu.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

In einigen Fällen können der Primärschlüssel oder der eindeutige Index der Zieltabelle und der Primärschlüssel oder der eindeutige Index der Quelltable unterschiedlich sein. Fügen Sie in diesen Fällen die zusätzliche Protokollierung manuell für die Spalten der Quelltable hinzu, die den Primärschlüssel oder den eindeutigen Index der Zieltabelle darstellen.

Wenn Sie den Primärschlüssel der Zieltabelle ändern, sollten Sie die zusätzliche Protokollierung für die Spalten des eindeutigen Zielindex hinzufügen, anstatt für die Spalten des Primärschlüssels oder des eindeutigen Index der Quelle.

Wenn für eine Tabelle ein Filter oder eine Transformation definiert ist, müssen Sie möglicherweise die zusätzliche Protokollierung aktivieren.

Berücksichtigen Sie dabei Folgendes:

- Wenn der Tabelle die zusätzliche Protokollierung `ALL COLUMNS` hinzugefügt wird, müssen Sie keine weitere Protokollierung hinzufügen.
- Wenn die Tabelle über einen eindeutigen Index oder einen Primärschlüssel verfügt, fügen Sie die zusätzliche Protokollierung für jede Spalte hinzu, die von einem Filter oder einer Transformation betroffen ist. Tun Sie dies jedoch nur, wenn sich diese Spalten von den Spalten unterscheiden, die den Primärschlüssel oder den eindeutigen Index darstellen.
- Wenn eine Transformation nur eine Spalte umfasst, fügen Sie diese Spalte nicht einer zusätzlichen Protokollierungsgruppe hinzu. Fügen Sie für eine Transformation `A+B` beispielsweise die zusätzliche Protokollierung für beide Spalten, `A` und `B`, hinzu. Fügen Sie jedoch für eine Transformation `substring(A, 10)` jedoch keine zusätzliche Protokollierung für Spalte `A` hinzu.
- Sie können die zusätzliche Protokollierung `USER_LOG_GROUP` einrichten, um die zusätzliche Protokollierung für Spalten, die den Primärschlüssel oder den eindeutigen Index darstellen, und andere Spalten zu aktivieren, die gefiltert oder transformiert werden. Fügen Sie diese Protokollierung sowohl für die Spalten, die den Primärschlüssel oder den eindeutigen Index darstellen, als auch für die anderen spezifischen Spalten hinzu, die gefiltert oder transformiert werden.

Wenn Sie beispielsweise eine Tabelle mit dem Namen `TEST.LOGGING` mit dem Primärschlüssel `ID` und einem Filter nach der Spalte `NAME` replizieren möchten, können Sie einen Befehl wie den folgenden ausführen, um die zusätzliche Protokollierung der Protokollgruppe zu erstellen.

```
ALTER TABLE TEST.LOGGING ADD SUPPLEMENTAL LOG GROUP TEST_LOG_GROUP (ID, NAME) ALWAYS;
```

Kontoberechtigungen sind erforderlich, wenn Sie Oracle für den Zugriff LogMiner auf die Redo-Logs verwenden

Um mit Oracle auf die Redo-Logs zuzugreifen LogMiner, gewähren Sie dem Oracle-Benutzer die folgenden Rechte, die in den Verbindungseinstellungen für den Oracle-Endpunkt angegeben sind.

```
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.
```

Kontoberechtigungen sind erforderlich, wenn AWS DMS Binary Reader für den Zugriff auf die Redo-Logs verwendet wird

Um mit dem AWS DMS Binary Reader auf die Redo-Logs zuzugreifen, gewähren Sie dem Oracle-Benutzer, der in den Verbindungseinstellungen für den Oracle-Endpunkt angegeben ist, die folgenden Rechte.

```
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 arbeitet mit Oracle-Dateimerkmalen, zu denen Oracle-Verzeichnisse gehören. Jedes Oracle-Verzeichnisobjekt enthält den Namen des Ordners, der die zu verarbeitenden Wiederholungsprotokolldateien enthält. Diese Oracle-Verzeichnisse sind nicht auf Dateisystemebene repräsentiert. Stattdessen handelt es sich um logische Verzeichnisse, die auf Oracle-Datenbankebene erstellt werden. Sie können sie in der Oracle ALL_DIRECTORIES-Ansicht anzeigen.

Wenn Sie diese Oracle-Verzeichnisse erstellen möchten AWS DMS , gewähren Sie die oben angegebene CREATE ANY DIRECTORY Berechtigung. AWS DMS erstellt die Verzeichnisnamen mit dem DMS_ Präfix. Wenn Sie die CREATE ANY DIRECTORY-Berechtigung nicht erteilen, erstellen Sie die entsprechenden Verzeichnisse manuell. In einigen Fällen ist bei der manuellen Erstellung der Oracle-Verzeichnisse der im Oracle-Quellendpunkt angegebene Oracle-Benutzer nicht der Benutzer, der diese Verzeichnisse erstellt hat. In diesen Fällen erteilen Sie auch die READ on DIRECTORY-Berechtigung.

Wenn sich der Oracle-Quellendpunkt im Active Dataguard Standby (ADG) befindet, finden Sie weitere Informationen im Beitrag [How to use Binary Reader with ADG](#) im Database Blog. AWS

Note

AWS DMS CDC unterstützt Active Dataguard Standby nicht, das nicht für die Verwendung des automatischen Redo-Transport-Service konfiguriert ist.

In einigen Fällen können Sie möglicherweise Oracle Managed Files (OMF) zum Speichern der Protokolle verwenden. Oder Ihr Quellendpunkt befindet sich in ADG, sodass Sie die Berechtigung `CREATE ANY DIRECTORY` nicht erteilen können. Erstellen Sie in diesen Fällen die Verzeichnisse mit allen möglichen Protokollspeicherorten manuell, bevor Sie die Replizierungsaufgabe starten. AWS DMS Wenn AWS DMS nicht das erwartete vorangestellte Verzeichnis findet, wird die Aufgabe beendet. Weiterhin löscht AWS DMS nicht die Einträge, die es in der `ALL_DIRECTORIES`-Ansicht erstellt hat, löschen Sie diese daher manuell.

Erforderliche zusätzliche Kontoberechtigungen bei Verwendung von Binary Reader mit Oracle ASM

Wenn Sie mit Binary Reader auf die Redo-Protokolle in Automatic Storage Management (ASM) zugreifen möchten, erteilen Sie dem Oracle-Benutzer, der in den Einstellungen zur Oracle-Endpunktverbindung angegeben ist, die folgenden Berechtigungen:

```
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.
```

Sie können den ASM-Kontozugriff validieren, indem Sie eine Eingabeaufforderung öffnen und eine der folgenden Anweisungen aufrufen, je nach Ihrer zuvor angegebenen Oracle-Version.

Wenn Sie die `SYSDBA`-Berechtigung benötigen, verwenden Sie Folgendes.

```
sqlplus asmuser/asmpassword@asmserver as sysdba
```

Wenn Sie die `SYSASM`-Berechtigung benötigen, verwenden Sie Folgendes.

```
sqlplus asmuser/asmpassword@asmserver as sysasm
```

Verwenden von selbstverwaltetem Oracle Standby als Quelle mit Binary Reader für CDC in AWS DMS

Wenn Sie eine Oracle-Standby-Instance bei Verwendung von Binary Reader für CDC als Quelle konfigurieren möchten, müssen die folgenden Voraussetzungen erfüllt sein:

- AWS DMS unterstützt derzeit nur Oracle Active Data Guard Standby.
- Stellen Sie sicher, dass die Konfiguration von Oracle Data Guard Folgendes verwendet:
 - Redo-Transport-Services für automatisierte Übertragungen von Redo-Daten

- Anwendungsservices, um Redo automatisch auf die Standby-Datenbank anzuwenden

Führen Sie die folgende Abfrage aus, um zu überprüfen, ob diese Anforderungen erfüllt sind.

```
SQL> select open_mode, database_role from v$database;
```

Vergewissern Sie sich anhand der Ausgabe dieser Abfrage, dass die Standby-Datenbank im Modus READ ONLY geöffnet ist und Redo automatisch angewendet wird. Beispielsweise:

```
OPEN_MODE          DATABASE_ROLE
-----          -
READ ONLY WITH APPLY  PHYSICAL STANDBY
```

So konfigurieren Sie eine Oracle-Standby-Instance bei Verwendung von Binary Reader für CDC als Quelle

1. Erteilen Sie zusätzliche Berechtigungen, die für den Zugriff auf Standby-Protokolldateien erforderlich sind.

```
GRANT SELECT ON v_$standby_log TO db_user;
```

2. Erstellen Sie mithilfe der AWS Management Console oder der AWS CLI einen Quellendpunkt für Oracle Standby. Geben Sie die folgenden zusätzlichen Verbindungsattribute an, wenn Sie den Endpunkt erstellen.

```
useLogminerReader=N;useBfile=Y;
```

Note

In können Sie zusätzliche Verbindungsattribute verwenden AWS DMS, um anzugeben, ob Sie aus den Archivprotokollen statt aus den Redo-Logs migrieren möchten. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).

3. Konfigurieren Sie das Ziel für archivierte Protokolle.

Der DMS-Binary-Reader für Oracle-Quellen ohne ASM verwendet Oracle-Verzeichnisse, um auf archivierte Redo-Protokolle zuzugreifen. Wenn Ihre Datenbank so konfiguriert ist, dass

Fast Recovery Area (FRA) als Ziel für Archivprotokolle verwendet wird, ist der Speicherort der Archiv-Redo-Dateien nicht konstant. An jedem Tag, an dem archivierte Redo-Protokolle generiert werden, wird in der FRA ein neues Verzeichnis mit dem Verzeichnisnamenformat JJJJ_MM_TT erstellt. Beispielsweise:

```
DB_RECOVERY_FILE_DEST/SID/archivelog/YYYY_MM_DD
```

Wenn DMS Zugriff auf archivierte Redo-Dateien im neu erstellten FRA-Verzeichnis benötigt und die Primärdatenbank mit Lese-/Schreibzugriff als Quelle verwendet wird, erstellt DMS wie folgt ein neues Oracle-Verzeichnis oder ersetzt ein vorhandenes Oracle-Verzeichnis.

```
CREATE OR REPLACE DIRECTORY dmsrep_taskid AS 'DB_RECOVERY_FILE_DEST/SID/archivelog/YYYY_MM_DD' ;
```

Wenn die Standby-Datenbank als Quelle verwendet wird, kann DMS das Oracle-Verzeichnis nicht erstellen oder ersetzen, da sich die Datenbank im schreibgeschützten Modus befindet. Sie können jedoch einen dieser zusätzlichen Schritte ausführen:

- a. Ändern Sie `log_archive_dest_id_1` so, dass ein tatsächlicher Pfad anstelle von FRA in einer solchen Konfiguration verwendet wird, damit Oracle keine täglichen Unterverzeichnisse erstellt:

```
ALTER SYSTEM SET log_archive_dest_1='LOCATION=full directory path'
```

Erstellen Sie dann ein Oracle-Verzeichnisobjekt, das von DMS verwendet werden soll:

```
CREATE OR REPLACE DIRECTORY dms_archived_logs AS 'full directory path' ;
```

- b. Erstellen Sie ein zusätzliches Archivprotokollziel und ein Oracle-Verzeichnisobjekt, das auf dieses Ziel verweist. Beispielsweise:

```
ALTER SYSTEM SET log_archive_dest_3='LOCATION=full directory path' ;  
CREATE DIRECTORY dms_archived_log AS 'full directory path' ;
```

Fügen Sie dann dem Quellendpunkt der Aufgabe ein zusätzliches Verbindungsattribut hinzu:

```
archivedLogDestId=3
```

- c. Erstellen Sie vorab manuell Oracle-Verzeichnisobjekte, die von DMS verwendet werden sollen.

```
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. Erstellen Sie einen Oracle-Scheduler-Auftrag, der täglich ausgeführt wird und das erforderliche Verzeichnis erstellt.

Verwenden einer benutzerverwalteten Datenbank in Oracle Cloud Infrastructure (OCI) als Quelle für CDC in AWS DMS

Eine benutzerverwaltete Datenbank ist eine Datenbank, die Sie konfigurieren und steuern, z. B. eine Oracle-Datenbank, die in einer virtuellen Maschine (VM), einer Bare-Metal-Instance oder einem Exadata-Server erstellt wurde. Es kann sich auch um Datenbanken handeln, die Sie konfigurieren und steuern und die auf einer dedizierten Infrastruktur wie Oracle Cloud Infrastructure (OCI) ausgeführt werden. Die folgenden Informationen beschreiben die Berechtigungen und Konfigurationen, die Sie benötigen, wenn Sie eine benutzerverwaltete Oracle-Datenbank in OCI als Quelle für die Erfassung von Datenänderungen (CDC) in AWS DMS verwenden.

So konfigurieren Sie eine von OCI gehostete benutzerverwaltete Oracle-Datenbank als Quelle für die Erfassung von Datenänderungen

1. Erteilen Sie erforderliche Benutzerkontoberechtigungen für eine benutzerverwaltete Quelldatenbank in OCI. Weitere Informationen finden Sie unter [Account privileges for a self-managed Oracle source endpoint](#).
2. Erteilen Sie Kontoberechtigungen, die bei Verwendung von Binary Reader für den Zugriff auf die Redo-Protokolle erforderlich sind. Weitere Informationen finden Sie unter [Erforderliche Kontoberechtigungen bei Verwendung von Binary Reader](#).
3. Fügen Sie Kontoberechtigungen hinzu, die bei Verwendung von Binary Reader mit Oracle Automatic Storage Management (ASM) erforderlich sind. Weitere Informationen finden Sie unter [Erforderliche zusätzliche Kontoberechtigungen bei Verwendung von Binary Reader mit Oracle ASM](#).
4. Richten Sie die zusätzliche Protokollierung ein. Weitere Informationen finden Sie unter [Einrichten der zusätzlichen Protokollierung](#).

5. Richten Sie die TDE-Verschlüsselung ein. Weitere Informationen finden Sie unter [Verschlüsselungsmethoden bei Verwendung einer Oracle-Datenbank als Quellendpunkt](#).

Die folgenden Einschränkungen gelten für die Replikation von Daten aus einer Oracle-Quelldatenbank in Oracle Cloud Infrastructure (OCI).

Einschränkungen

- DMS unterstützt die Verwendung von Oracle für den Zugriff LogMiner auf die Redo-Logs nicht.
- DMS unterstützt Autonomous DB nicht.

Arbeiten mit einer AWS-verwalteten Oracle-Datenbank als Quelle für AWS DMS

Eine AWS verwaltete Datenbank ist eine Datenbank, die sich auf einem Amazon-Service wie Amazon RDS, Amazon Aurora oder Amazon S3 befindet. Im Folgenden finden Sie die Rechte und Konfigurationen, die Sie einrichten müssen, wenn Sie eine AWS-verwaltete Oracle-Datenbank mit verwenden. AWS DMS

Erforderliche Benutzerkontenberechtigungen für eine AWS-verwaltete Oracle-Quelle für AWS DMS

Erteilen Sie dem in der Oracle-Quellendpunktdefinition angegebenen Oracle-Benutzerkonto die folgenden Berechtigungen.

Important

Für alle Parameterwerte wie *db_user* und *any-replicated-table* geht Oracle davon aus, dass der Wert ausschließlich in Großbuchstaben geschrieben ist, es sei denn, Sie geben den Wert mit einer Kennung an, bei der Groß- und Kleinschreibung beachtet wird. Nehmen wir beispielsweise an, dass Sie einen *db_user*-Wert ohne Anführungszeichen erstellen, wie in `CREATE USER myuser` oder `CREATE USER MYUSER`. In diesem Fall identifiziert und speichert Oracle den Wert ausschließlich in Großbuchstaben (MYUSER). Wenn Sie Anführungszeichen verwenden, wie in `CREATE USER "MyUser"` oder `CREATE USER 'MyUser'`, identifiziert und speichert Oracle den angegebenen Wert unter Berücksichtigung der Groß- und Kleinschreibung (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.
```

Erteilen Sie außerdem die Berechtigungen SELECT und EXECUTE für SYS-Objekte mithilfe des Amazon-RDS-Verfahrens `rdsadmin.rdsadmin_util.grant_sys_object`, wie in der Abbildung gezeigt. Weitere Informationen finden Sie unter [Erteilen von SELECT- oder EXECUTE-Berechtigungen für SYS-Objekte](#).

```
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');
```

Weitere Informationen zur Verwendung von Amazon RDS Active Dataguard Standby (ADG) mit AWS DMS finden Sie unter [Verwenden von Amazon RDS Oracle Standby \(Lesereplikat\) als Quelle mit Binary Reader für CDC in AWS DMS](#).

Weitere Hinweise zur Verwendung von Oracle TDE mit finden Sie AWS DMS unter. [Unterstützte Verschlüsselungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS](#)

Voraussetzungen für die Verarbeitung offener Transaktionen für Oracle Standby

Wenn Sie AWS DMS Versionen 3.4.6 und höher verwenden, führen Sie die folgenden Schritte aus, um offene Transaktionen für Oracle Standby zu verarbeiten.

1. Erstellen Sie einen Datenbank-Link mit dem Namen AWSDMS_DBLINK in der Primärdatenbank. **DMS_USER** verwendet den Datenbank-Link, um eine Verbindung mit der Primärdatenbank herzustellen. Beachten Sie, dass der Datenbank-Link von der Standby-Instance aus ausgeführt wird, um die offenen Transaktionen abzufragen, die in der Primärdatenbank ausgeführt werden. Sehen Sie sich das folgende -Beispiel an.

```
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. Stellen Sie sicher, dass die Verbindung mit dem Datenbank-Link über *DMS_USER* hergestellt ist, wie im folgenden Beispiel dargestellt.

```
select 1 from dual@AWSDMS_DBLINK
```

Konfiguration einer AWS-verwalteten Oracle-Quelle für AWS DMS

Bevor Sie eine AWS-verwaltete Oracle-Datenbank als Quelle für verwenden AWS DMS, führen Sie die folgenden Aufgaben für die Oracle-Datenbank aus:

- Aktivieren Sie automatische Sicherungen. Weitere Informationen zum Aktivieren von automatischen Backups finden Sie unter [Aktivieren von automatisierten Backups](#) im Benutzerhandbuch für Amazon RDS.
- Richten Sie die zusätzliche Protokollierung ein.
- Richten Sie die Archivierung ein. Durch die Archivierung der Redo-Logs für Ihre Amazon RDS for Oracle DB-Instance können AWS DMS Sie die Protokollinformationen mit Oracle LogMiner oder Binary Reader abrufen.

So richten Sie die Archivierung ein:

1. Führen Sie den `rdsadmin.rdsadmin_util.set_configuration` Befehl aus, um die Archivierung einzurichten.

Führen Sie beispielsweise den folgenden Befehl aus, um die archivierten Redo-Protokolle 24 Stunden lang beizubehalten.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
commit;
```

Note

Das Commit muss durchgeführt werden, damit Änderungen wirksam werden.

2. Stellen Sie sicher, dass Ihr Speicherplatz für die archivierten Redo-Protokolle im angegebenen Zeitraum ausreicht. Wenn Ihr Speicherzeitraum beispielsweise 24 Stunden beträgt, berechnen Sie die Gesamtgröße Ihrer kumulierten archivierten Redo-Protokolle über eine typische Stunde Transaktionsverarbeitung und multiplizieren Sie diese Summe mit 24. Vergleichen Sie diese berechnete Summe für 24 Stunden mit Ihrem verfügbaren Speicherplatz und entscheiden Sie, ob Sie über genügend Speicherplatz für eine 24-Stunden-Transaktionsverarbeitung verfügen.

So richten Sie die zusätzliche Protokollierung ein:

1. Führen Sie den folgenden Befehl aus, um die zusätzliche Protokollierung auf Datenbankebene zu aktivieren.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');
```

2. Führen Sie den folgenden Befehl aus, um die zusätzliche Protokollierung für den Primärschlüssel zu aktivieren.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD','PRIMARY KEY');
```

3. (Optional) Aktivieren Sie die zusätzliche Schlüsselebenen-Protokollierung auf Tabellenebene.

Für die Quelldatenbank entsteht ein wenig Mehraufwand, wenn die zusätzliche Schlüsselebenen-Protokollierung aktiviert ist. Wenn Sie nur eine Teilmenge Ihrer Tabellen migrieren, sollten Sie daher die zusätzliche Schlüsselebenen-Protokollierung auf Tabellenebene aktivieren. Führen Sie den folgenden Befehl aus, um die zusätzliche Schlüsselebenen-Protokollierung auf Tabellenebene zu aktivieren.

```
alter table table_name add supplemental log data (PRIMARY KEY) columns;
```

Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS

Sie können den AWS DMS Zugriff auf die Redo-Logs der Amazon RDS for Oracle Oracle-Quellinstanz mithilfe von Binary Reader for CDC konfigurieren.

Note

Um Oracle zu verwenden LogMiner, sind die mindestens erforderlichen Benutzerkontoberechtigungen ausreichend. Weitere Informationen finden Sie unter [Erforderliche Benutzerkontoberechtigungen für eine AWS-verwaltete Oracle-Quelle für AWS DMS](#).

Um AWS DMS Binary Reader zu verwenden, geben Sie je nach AWS DMS Version zusätzliche Einstellungen und zusätzliche Verbindungsattribute für den Oracle-Quellendpunkt an.

Unterstützung für Binary Reader ist in den folgenden Versionen von Amazon RDS für Oracle verfügbar:

- Oracle 11.2 – Versionen 11.2.0.4V11 und höher
- Oracle 12.1 – Versionen 12.1.0.2.V7 und höher
- Oracle 12.2 – Alle Versionen
- Oracle 18.0 – Alle Versionen
- Oracle 19.0 – Alle Versionen

So konfigurieren Sie CDC mit dem Binary Reader:

1. Melden Sie sich als Hauptbenutzer bei Ihrer Quelldatenbank in Amazon RDS für Oracle an und führen Sie die folgenden gespeicherten Prozeduren aus, um die Verzeichnisse auf Serverebene zu erstellen.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;  
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

2. Erteilen Sie dem Oracle-Benutzerkonto, das für den Zugriff auf den Oracle-Quellendpunkt verwendet wird, die folgenden Berechtigungen.

```
GRANT READ ON DIRECTORY ONLINELOG_DIR TO db_user;  
GRANT READ ON DIRECTORY ARCHIVELOG_DIR TO db_user;
```

3. Legen Sie die folgenden zusätzlichen Verbindungsattribute für den Oracle-Quellendpunkt in Amazon RDS fest:

- Legen Sie für die RDS-Oracle-Versionen 11.2 und 12.1 Folgendes fest.

```
useLogminerReader=N;useBfile=Y;accessAlternateDirectly=false;useAlternateFolderForOnline=  
oraclePathPrefix=/rdsdbdata/db/{"$DATABASE_NAME"}_A/;usePathPrefix=/rdsdbdata/  
log/;replacePathPrefix=true;
```

- Legen Sie für die RDS-Oracle-Versionen 12.2, 18.0 und 19.0 Folgendes fest.

```
useLogminerReader=N;useBfile=Y;
```

Note

Stellen Sie sicher, dass auf das Semikolon (;) als Trennzeichen für mehrere Einstellungen kein Leerzeichen folgt (z. B. `oneSetting;thenAnother`).

Weitere Informationen zum Konfigurieren einer CDC-Aufgabe finden Sie unter [Konfiguration für CDC für eine Oracle-Quelldatenbank](#).

Verwenden von Amazon RDS Oracle Standby (Lesereplikat) als Quelle mit Binary Reader für CDC in AWS DMS

Überprüfen Sie die folgenden Voraussetzungen für die Verwendung von Amazon RDS für Oracle Standby als Quelle bei Verwendung von Binary Reader für CDC in AWS DMS:

- Verwenden Sie den Oracle-Hauptbenutzer, um Binary Reader einzurichten.
- Stellen Sie sicher, dass AWS DMS derzeit nur die Verwendung von Oracle Active Data Guard Standby unterstützt wird.

Verwenden Sie anschließend das folgende Verfahren, um RDS für Oracle Standby bei Verwendung von Binary Reader für CDC als Quelle zu verwenden.

So konfigurieren Sie RDS für Oracle Standby bei Verwendung von Binary Reader für CDC als Quelle

1. Melden Sie sich als Hauptbenutzer bei der primären RDS-für-Oracle-Instance an.
2. Führen Sie die folgenden gespeicherten Prozeduren wie im Benutzerhandbuch für Amazon RDS beschrieben aus, um die Verzeichnisse auf Serverebene zu erstellen.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

3. Identifizieren Sie die in Schritt 2 erstellten Verzeichnisse.

```
SELECT directory_name, directory_path FROM all_directories
WHERE directory_name LIKE ( 'ARCHIVELOG_DIR_%' )
      OR directory_name LIKE ( 'ONLINELOG_DIR_%' )
```

Mit dem obigen Code wird beispielsweise eine Liste von Verzeichnissen wie die folgende angezeigt.

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. Erteilen Sie dem Oracle-Benutzerkonto, das für den Zugriff auf Oracle Standby verwendet wird, die Berechtigung Read für die oben genannten Verzeichnisse.

```
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. Führen Sie einen Archivprotokoll-Wechsel auf der primären Instance durch. Dadurch wird sichergestellt, dass die Änderungen an ALL_DIRECTORIES auch in Oracle Standby portiert werden.
6. Führen Sie die Abfrage ALL_DIRECTORIES für Oracle Standby aus, um zu bestätigen, dass die Änderungen übernommen wurden.

- Erstellen Sie mithilfe der AWS DMS Management Console oder AWS Command Line Interface (AWS CLI) einen Quellendpunkt für den Oracle Standby. Geben Sie die folgenden zusätzlichen Verbindungsattribute an, während Sie den Endpunkt erstellen.

```
useLogminerReader=N;useBfile=Y;archivedLogDestId=1;additionalArchivedLogDestId=2
```

- Nachdem Sie den Endpunkt erstellt haben, verwenden Sie die Option Endpunktverbindung testen auf der Seite Endpunkt erstellen der Konsole oder verwenden Sie den AWS CLI `test-connection` Befehl, um zu überprüfen, ob die Konnektivität hergestellt ist.

Einschränkungen bei der Verwendung von Oracle als Quelle für AWS DMS

Die folgenden Einschränkungen gelten, wenn Sie eine Oracle-Datenbank als Quelle für AWS DMS verwenden:

- AWS DMS unterstützt Oracle Extended-Datentypen in AWS DMS Version 3.5.0 und höher.
- AWS DMS unterstützt keine langen Objektnamen (über 30 Byte).
- AWS DMS unterstützt keine funktionsbasierten Indizes.
- Wenn Sie die zusätzliche Protokollierung verwalten und Transformationen für eine der Spalten durchführen, stellen Sie sicher, dass die zusätzliche Protokollierung für alle Felder und Spalten aktiviert ist. Weitere Informationen zum Einrichten der zusätzlichen Protokollierung finden Sie in den folgenden Themen:
 - Informieren Sie sich im Fall einer selbstverwalteten Oracle-Quelldatenbank unter [Einrichten der zusätzlichen Protokollierung](#).
 - Informationen zu einer von AWS-verwalteten Oracle-Quelldatenbank finden Sie unter [Konfiguration einer AWS-verwalteten Oracle-Quelle für AWS DMS](#)
- AWS DMS unterstützt die Mehrmandanten-Container-Root-Datenbank (CDB\$ROOT) nicht. Es unterstützt eine PDB, die den Binary Reader verwendet.
- AWS DMS unterstützt keine verzögerten Einschränkungen.
- Sichere LOBs werden im vollständigen LOB-Modus nur unterstützt, wenn ein LOB-Lookup durchgeführt wird.
- AWS DMS unterstützt die `rename table table-name to new-table-name` Syntax für alle unterstützten Oracle-Versionen 11 und höher. Diese Syntax wird für Oracle-Quelldatenbanken der Version 10 nicht unterstützt.

- AWS DMS repliziert die Ergebnisse der DDL-Anweisung nicht. `ALTER TABLE ADD column data_type DEFAULT default_value` Anstatt `default_value` auf das Ziel zu replizieren, wird die neue Spalte auf NULL gesetzt.
- Wenn Sie AWS DMS Version 3.4.7 oder höher verwenden, um Änderungen zu replizieren, die sich aus Partitions- oder Unterpartitionsvorgängen ergeben, gehen Sie wie folgt vor, bevor Sie eine DMS-Aufgabe starten.
 - Erstellen Sie die partitionierte Tabellenstruktur (DDL) manuell;
 - Stellen Sie sicher, dass die DDL in der Oracle-Quelle und im Oracle-Ziel identisch ist;
 - Legen Sie das zusätzliche Verbindungsattribut `enableHomogenousPartitionOps=true` fest.

Mehr über `enableHomogenousPartitionOps` erfahren Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#). Beachten Sie außerdem, dass DMS bei FULL+CDC-Aufgaben keine Datenänderungen repliziert, die als Teil der zwischengespeicherten Änderungen erfasst wurden. Erstellen Sie in diesem Anwendungsfall die Tabellenstruktur im Oracle-Ziel neu und laden Sie die betreffenden Tabellen erneut.

Vor Version 3.4.7: AWS DMS

DMS repliziert keine Datenänderungen, die sich aus Partitions- oder Unterpartitionsoperationen (ADD, DROP, EXCHANGE und TRUNCATE) ergeben. Solche Aktualisierungen können während der Replikation folgende Fehler verursachen:

- Bei ADD-Operationen können Aktualisierungen und Löschungen für die hinzugefügten Daten die Warnung „0 rows affected (0 Zeilen betroffen)“ auslösen.
- Für DROP- und TRUNCATE-Operationen können neue Einfügungen „Duplicates (Duplikate)“-Fehler verursachen.
- EXCHANGE-Operationen können sowohl eine Warnung „0 rows affected (0 Zeilen betroffen)“, als auch „Duplicates (Duplikate)“-Fehler auslösen.

Um Änderungen zu replizieren, die sich aus Partitions- oder Unterpartitionsoperationen ergeben, laden Sie die betreffenden Tabellen neu. Nach dem Hinzufügen einer neuen leeren Partition werden Operationen für die neu hinzugefügte Partition wie gewohnt auf das Ziel repliziert.

- AWS DMS Versionen vor 3.4 unterstützen keine Datenänderungen auf dem Ziel, die sich aus der Ausführung der `CREATE TABLE AS` Anweisung auf der Quelle ergeben. Allerdings wird die neue Tabelle in der Zieldatenbank erstellt.
- AWS DMS erfasst keine Änderungen, die durch das `DBMS_REDEFINITION` Oracle-Paket vorgenommen wurden, z. B. die Tabellenmetadaten und das `OBJECT_ID` Feld.

- AWS DMS ordnet leere BLOB- und CLOB-Spalten NULL dem Ziel zu.
- Beim Erfassen von Änderungen mit Oracle 11 LogMiner geht eine Aktualisierung einer CLOB-Spalte mit einer Zeichenkettenlänge von mehr als 1982 verloren, und das Ziel wird nicht aktualisiert.
- Unterstützt während der Change Data Capture (CDC) AWS DMS keine Batch-Aktualisierungen von numerischen Spalten, die als Primärschlüssel definiert sind.
- AWS DMS unterstützt bestimmte UPDATE Befehle nicht. Das folgende Beispiel ist ein nicht unterstützter UPDATE-Befehl.

```
UPDATE TEST_TABLE SET KEY=KEY+1;
```

Hier ist TEST_TABLE der Tabellename und KEY ist eine numerische Spalte, die als Primärschlüssel definiert ist.

- AWS DMS unterstützt nicht den vollständigen LOB-Modus zum Laden von LONG- und LONG RAW-Spalten. Stattdessen können Sie den limitierten LOB-Modus verwenden, um diese Datentypen zu einem Oracle-Ziel zu migrieren. Im eingeschränkten LOB-Modus werden alle Daten, die Sie AWS DMS auf LONG- oder LONG-RAW-Spalten mit einer Länge von mehr als 64 KB festgelegt haben, auf 64 KB gekürzt.
- AWS DMS unterstützt den vollständigen LOB-Modus zum Laden von XMLTYPE-Spalten nicht. Stattdessen können Sie den limitierten LOB-Modus verwenden, um XMLTYPE-Spalten zu einem Oracle-Ziel zu migrieren. Im limitierten LOB-Modus kürzt DMS alle Daten, die größer als die benutzerdefinierte Variable „Maximale LOB-Größe“ sind. Der empfohlene Höchstwert für „Maximale LOB-Größe“ beträgt 100 MB.
- AWS DMS repliziert keine Tabellen, deren Namen Apostrophe enthalten.
- AWS DMS unterstützt CDC aus materialisierten Ansichten. DMS unterstützt CDC jedoch nicht aus anderen Ansichten.
- AWS DMS unterstützt CDC nicht für indexorganisierte Tabellen mit einem Überlaufsegment.
- AWS DMS unterstützt den Drop Partition Vorgang nicht für Tabellen, die nach Referenz partitioniert sind und auf gesetzt sind. `enableHomogenousPartitionOps true`
- Wenn Sie Oracle für den LogMiner Zugriff auf die Redo-Logs verwenden, gelten AWS DMS die folgenden Einschränkungen:
 - Nur für Oracle 12, repliziert AWS DMS keine Änderungen an LOB-Spalten.
 - Repliziert AWS DMS nicht für alle Oracle-Versionen das Ergebnis von UPDATE Operationen an XMLTYPE und LOB-Spalten.

- AWS DMS unterstützt keine XA-Transaktionen bei der Replikation bei der Verwendung von Oracle. LogMiner
- Oracle unterstützt LogMiner keine Verbindungen zu einer Pluggable Database (PDB). Um eine Verbindung mit einer PDB herzustellen, greifen Sie mit dem Binary Reader auf die Wiederholungsprotokolle zu.
- SHRINK SPACE-Operationen werden nicht unterstützt.
- Wenn Sie Binary Reader verwenden, gelten folgende Einschränkungen AWS DMS :
 - Es unterstützt keine Tabellencluster.
 - Es werden nur SHRINK SPACE-Operationen auf Tabellenebene unterstützt. Diese Ebene umfasst die vollständige Tabelle, Partitionen und Unterpartitionen.
 - Es unterstützt keine Änderungen an indexorganisierten Tabellen mit Schlüsselkomprimierung.
 - Die Implementierung von Online-Redo-Protokollen auf Raw-Devices wird nicht unterstützt.
 - Binary Reader unterstützt TDE nur für selbstverwaltete Oracle-Datenbanken, da RDS für Oracle das Abrufen von Wallet-Passwörtern für TDE-Verschlüsselungsschlüssel nicht unterstützt.
- AWS DMS unterstützt keine Verbindungen zu einer Amazon RDS-Oracle-Quelle, die einen Oracle Automatic Storage Management (ASM) -Proxy verwendet.
- AWS DMS unterstützt keine virtuellen Spalten.
- AWS DMS unterstützt weder den ROWID Datentyp noch materialisierte Ansichten, die auf einer ROWID-Spalte basieren.

AWS DMS unterstützt teilweise Oracle Materialized Views. Bei Volllastaufgaben kann DMS eine Vollstauskopie einer materialisierten Ansicht in Oracle erstellen. DMS kopiert die materialisierte Ansicht als Basistabelle in das Zielsystem und ignoriert alle ROWID-Spalten in der materialisierten Ansicht. Bei einer laufenden Replikation (CDC) versucht DMS, Änderungen an den Daten der materialisierten Ansicht zu replizieren. Jedoch sind die Ergebnisse möglicherweise nicht ideal. Insbesondere wenn die materialisierte Ansicht vollständig aktualisiert wird, repliziert DMS einzelne Löschvorgänge für alle Zeilen, gefolgt von einzelnen Einfügungen für alle Zeilen. Dieser Vorgang ist sehr ressourcenintensiv und funktioniert bei materialisierten Ansichten mit einer großen Anzahl von Zeilen möglicherweise schlecht. Bei einer laufenden Replikation, bei der die materialisierten Ansichten schnell aktualisiert werden, versucht DMS, die schnellen Datenänderungen zu verarbeiten und zu replizieren. In beiden Fällen überspringt DMS alle ROWID-Spalten in der materialisierten Ansicht.

- AWS DMS lädt oder erfasst keine globalen temporären Tabellen.

- Aktivieren Sie für S3-Ziele, die Replikation verwenden, die zusätzliche Protokollierung für jede Spalte, damit Quellzeilenaktualisierungen jeden Spaltenwert erfassen können. Zum Beispiel:
`alter table yourtablename add supplemental log data (all) columns;`
- Eine Aktualisierung für eine Zeile mit einem zusammengesetzten eindeutigen Schlüssel, der null enthält, kann am Ziel nicht repliziert werden.
- AWS DMS unterstützt nicht die Verwendung mehrerer Oracle TDE-Verschlüsselungsschlüssel auf demselben Quellendpunkt. Jeder Endpunkt kann nur ein Attribut für den TDE-Verschlüsselungsschlüsselnamen „securityDbEncryptionName“ und ein TDE-Passwort für diesen Schlüssel haben.
- Bei der Replikation von Amazon RDS for Oracle wird TDE nur mit verschlüsseltem Tablespace und unter Verwendung von Oracle unterstützt. LogMiner
- AWS DMS unterstützt nicht mehrere Operationen zur schnellen Umbenennung von Tabellen.
- Bei Verwendung von Oracle 19.0 als Quelle werden die folgenden Funktionen AWS DMS nicht unterstützt:
 - Data-Guard-DML-Umleitung
 - Partitionierte hybride Tabellen
 - Schema-only-Konten in Oracle
- AWS DMS unterstützt nicht die Migration von Tabellen oder Ansichten des Typs BIN\$ oder DR\$.
- Ab Oracle 18.x wird Change Data Capture (CDC) von Oracle Express Edition (Oracle Database XE) AWS DMS nicht unterstützt.
- Bei der Migration von Daten aus einer CHAR-Spalte kürzt DMS alle nachfolgenden Leerzeichen.
- AWS DMS unterstützt keine Replikation aus Anwendungscontainern.
- AWS DMS unterstützt nicht die Ausführung von Oracle Flashback Database und Restore Points, da diese Operationen die Konsistenz der Oracle Redo Log-Dateien beeinträchtigen.
- Das Verfahren INSERT zum direkten Laden mit der Option zur parallelen Ausführung wird in folgenden Fällen nicht unterstützt:
 - Unkomprimierte Tabellen mit mehr als 255 Spalten
 - Zeilengröße von mehr als 8K
 - Exadata-HCC-Tabellen
 - Datenbank, die auf der Big Endian-Plattform läuft
- Für eine Quelltable, die weder über einen Primärschlüssel noch über einen eindeutigen Schlüssel verfügt, muss die zusätzliche Protokollierung ALL COLUMN aktiviert sein. Dies führt zu mehr Redo-Protokollaktivitäten und kann die DMS-CDC-Latenz erhöhen.

- AWS DMS migriert keine Daten aus unsichtbaren Spalten in Ihrer Quelldatenbank. Verwenden Sie die Anweisung `ALTER TABLE`, um diese Spalten sichtbar zu machen und in den Migrationsbereich aufzunehmen.

SSL-Unterstützung für einen Oracle-Endpunkt

AWS DMS Oracle-Endgeräte unterstützen SSL V3 für die Modi `none` und `verify-ca` SSL. Um SSL mit einem Oracle-Endpunkt zu verwenden, müssen Sie das Oracle Wallet für den Endpunkt anstelle von `.pem`-Zertifikatdateien hochladen.

Themen

- [Verwenden eines vorhandenen Zertifikats für Oracle SSL](#)
- [Verwenden eines selbstsignierten Zertifikats für Oracle SSL](#)

Verwenden eines vorhandenen Zertifikats für Oracle SSL

Um eine vorhandene Oracle-Client-Installation zum Erstellen der Oracle Wallet-Datei aus der Zertifikatdatei der Zertifizierungsstelle zu erstellen, gehen Sie wie folgt vor.

So verwenden Sie eine vorhandene Oracle-Client-Installation für Oracle-SSL mit AWS DMS

1. Legen Sie die `ORACLE_HOME`-Systemvariable auf den Speicherort Ihres `dbhome_1`-Verzeichnisses fest, indem Sie den folgenden Befehl ausführen.

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

2. `$ORACLE_HOME/lib` an die `LD_LIBRARY_PATH`-Systemvariable anhängen.

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. Erstellen Sie ein Oracle Wallet unter `$ORACLE_HOME/ssl_wallet`.

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

- Legen Sie die CA-Zertifikatdatei `.pem` im Verzeichnis `ssl_wallet` ab. Wenn Sie Amazon RDS verwenden, können Sie die von Amazon RDS gehostete CA-Stammzertifikatdatei `rds-ca-2015-root.pem` verwenden. Weitere Informationen zum Herunterladen dieser Datei finden Sie unter [Verwenden von SSL/TLS für die Verschlüsselung einer Verbindung zu einer DB-Instance](#) im Benutzerhandbuch für Amazon RDS.
- Führen Sie die folgenden Befehle zur Oracle Wallet-Erstellung aus.

```
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
```

Wenn Sie die vorherigen Schritte ausgeführt haben, können Sie die Wallet-Datei mit dem `ImportCertificate-API`-Aufruf importieren, indem Sie den Parameter „certificate-wallet“ angeben. Sie können das importierte Wallet-Zertifikat dann verwenden, wenn Sie `verify-ca` als SSL-Modus beim Erstellen oder Ändern Ihres Oracle-Endpunkts auswählen.

Note

Oracle-Wallets sind Binärdateien. AWS DMS akzeptiert diese Dateien unverändert.

Verwenden eines selbstsignierten Zertifikats für Oracle SSL

Gehen Sie wie folgt vor, um ein selbstsigniertes Zertifikat für Oracle SSL zu verwenden. Dabei wird das Passwort `oracle123` für das Oracle-Wallet vorausgesetzt.

Um ein selbstsigniertes Zertifikat für Oracle SSL zu verwenden mit AWS DMS

- Erstellen Sie ein Verzeichnis für die Arbeit mit dem selbstsignierten Zertifikat.

```
mkdir -p /u01/app/oracle/self_signed_cert
```

- Wechseln Sie in das Verzeichnis, das Sie im vorherigen Schritt erstellt haben.

```
cd /u01/app/oracle/self_signed_cert
```

- Erstellen Sie einen Stammschlüssel.

```
openssl genrsa -out self-rootCA.key 2048
```

4. Signieren Sie selbst ein Stammzertifikat mithilfe des Stammschlüssels, den Sie im vorherigen Schritt erstellt haben.

```
openssl req -x509 -new -nodes -key self-rootCA.key  
-sha256 -days 3650 -out self-rootCA.pem
```

Verwenden Sie Eingabeparameter wie die folgenden:

- Country Name (2 letter code) [XX], z. B.: AU
 - State or Province Name (full name) [], z. B.: NSW
 - Locality Name (e.g., city) [Default City], z. B.: Sydney
 - Organization Name (e.g., company) [Default Company Ltd], z. B.: AmazonWebService
 - Organizational Unit Name (e.g., section) [], z. B.: DBeng
 - Common Name (e.g., your name or your server's hostname) [], z. B.: aws
 - Email Address [], z. B.: abcd.efgh@amazonwebservice.com
5. Erstellen Sie ein Oracle Wallet-Verzeichnis für die Oracle-Datenbank.

```
mkdir -p /u01/app/oracle/wallet
```

6. Erstellen Sie ein neues Oracle Wallet.

```
orapki wallet create -wallet "/u01/app/oracle/wallet" -pwd oracle123 -  
auto_login_local
```

7. Fügen Sie das Stammzertifikat dem Oracle Wallet hinzu.

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123 -trusted_cert  
-cert /u01/app/oracle/self_signed_cert/self-rootCA.pem
```

8. Listen Sie den Inhalt des Oracle Wallet auf. Die Liste sollte das Stammzertifikat enthalten.

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

Dies könnte beispielsweise wie folgt angezeigt werden.

```
Requested Certificates:  
User Certificates:  
Trusted Certificates:  
Subject:          CN=aws,OU=DBeng,O= AmazonWebService,L=Sydney,ST=NSW,C=AU
```

9. Generieren Sie die CSR (Certificate Signing Request) mithilfe des ORAPKI-Dienstprogramms.

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123  
-dn "CN=aws" -keysize 2048 -sign_alg sha256
```

10. Führen Sie den folgenden Befehl aus.

```
openssl pkcs12 -in /u01/app/oracle/wallet/ewallet.p12 -nodes -out /u01/app/oracle/  
wallet/nonoracle_wallet.pem
```

Es wird eine Ausgabe ähnlich der folgenden erstellt.

```
Enter Import Password:  
MAC verified OK  
Warning unsupported bag type: secretBag
```

11. Legen Sie "dms" als allgemeinen Namen fest.

```
openssl req -new -key /u01/app/oracle/wallet/nonoracle_wallet.pem -out certdms.csr
```

Verwenden Sie Eingabeparameter wie die folgenden:

- Country Name (2 letter code) [XX], z. B.: AU
- State or Province Name (full name) [], z. B.: NSW
- Locality Name (e.g., city) [Default City], z. B.: Sydney
- Organization Name (e.g., company) [Default Company Ltd], z. B.: AmazonWebService
- Organizational Unit Name (e.g., section) [], z. B.: aws
- Common Name (e.g., your name or your server's hostname) [], z. B.: aws
- Email Address [], z. B.: abcd.efgh@amazonwebservice.com

Stellen Sie sicher, dass dies nicht mit Schritt 4 identisch ist. Dazu können Sie beispielsweise den Namen der Organisationseinheit in einen anderen Namen ändern, wie in der Abbildung gezeigt.

Geben Sie die folgenden zusätzlichen Attribute ein, die zusammen mit Ihrer Zertifikatsanforderung gesendet werden sollen.

- A challenge password [], z. B.: oracle123
- An optional company name [], z. B.: aws

12. Rufen Sie die Zertifikatsignatur ab.

```
openssl req -noout -text -in certdms.csr | grep -i signature
```

Der Signaturschlüssel für diesen Beitrag lautet sha256WithRSAEncryption.

13. Führen Sie den folgenden Befehl aus, um die Zertifikatdatei (.crt) zu erstellen.

```
openssl x509 -req -in certdms.csr -CA self-rootCA.pem -CAkey self-rootCA.key  
-CAcreateserial -out certdms.crt -days 365 -sha256
```

Es wird eine Ausgabe ähnlich der folgenden erstellt.

```
Signature ok  
subject=/C=AU/ST=NSW/L=Sydney/O=awsweb/OU=DBGeng/CN=aws  
Getting CA Private Key
```

14. Fügen Sie das Zertifikat dem Wallet hinzu.

```
orapki wallet add -wallet /u01/app/oracle/wallet -pwd oracle123 -user_cert -cert  
certdms.crt
```

15. Sehen Sie sich das Wallet an. Es sollte zwei Einträge aufweisen. Siehe den folgenden Code.

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

16. Konfigurieren Sie die sqlnet.ora-Datei (\$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. Beenden Sie den Oracle Listener.

```
lsnrctl stop
```

18. Fügen Sie Einträge für SSL in der listener.ora-Datei (\$ORACLE_HOME/network/admin/listener.ora) hinzu.

```

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. Konfigurieren Sie die `tnsnames.ora`-Datei (`$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. Starten Sie den Oracle Listener neu.

```
lsnrctl start
```

21. Zeigen Sie den Status des Oracle Listener an.

```
lsnrctl status
```

22. Testen Sie die SSL-Verbindung zu der Datenbank von "localhost" mithilfe von `sqlplus` und dem SSL-Eintrag "tnsnames".

```
sqlplus -L ORACLE_USER@SID_ssl
```

23. Überprüfen Sie, dass Sie sich erfolgreich mithilfe von SSL verbunden haben.

```
SELECT SYS_CONTEXT('USERENV', 'network_protocol') FROM DUAL;

SYS_CONTEXT('USERENV', 'NETWORK_PROTOCOL')
-----
tcps
```

24. Wechseln Sie zu dem Verzeichnis mit dem selbstsignierten Zertifikat.

```
cd /u01/app/oracle/self_signed_cert
```

25. Erstellen Sie ein neues Client-Oracle-Wallet AWS DMS , das Sie verwenden können.

```
orapki wallet create -wallet ./ -auto_login_only
```

26. Fügen Sie das selbstsignierte Stammzertifikat dem Oracle Wallet hinzu.

```
orapki wallet add -wallet ./ -trusted_cert -cert self-rootCA.pem -auto_login_only
```

27. Listet den Inhalt des AWS DMS zu verwendenden Oracle-Wallets auf. Die Liste sollte das selbstsignierte Stammzertifikat enthalten.

```
orapki wallet display -wallet ./
```

Es wird eine Ausgabe ähnlich der folgenden erstellt.

```
Trusted Certificates:
Subject:          CN=aws,OU=DBeng,O=AmazonWebService,L=Sydney,ST=NSW,C=AU
```

28. Laden Sie das Oracle-Wallet hoch, das Sie gerade erstellt haben AWS DMS.

Unterstützte Verschlüsselungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS

In der folgenden Tabelle finden Sie die Methoden zur transparenten Datenverschlüsselung (TDE), die bei der Arbeit mit einer Oracle-Quelldatenbank AWS DMS unterstützt werden.

Zugriffsmethode für Wiederholungsprotokolle	TDE-Tabellenraum	TDE-Spalte
Oracle LogMiner	Ja	Ja
Binary Reader	Ja	Ja

AWS DMS unterstützt Oracle TDE, wenn Binary Reader verwendet wird, sowohl auf Spalten- als auch auf Tablespace-Ebene. Um die TDE-Verschlüsselung mit zu verwenden AWS DMS, identifizieren Sie zunächst den Oracle-Wallet-Speicherort, an dem der TDE-Verschlüsselungsschlüssel und das TDE-Passwort gespeichert sind. Identifizieren Sie dann den richtigen TDE-Verschlüsselungsschlüssel und das richtige Passwort für den Oracle-Quellendpunkt.

So identifizieren Sie den Verschlüsselungsschlüssel und das Passwort für die TDE-Verschlüsselung und geben diese an

1. Führen Sie die folgende Abfrage aus, um das Oracle-Verschlüsselungs-Wallet auf dem Oracle-Datenbank-Host zu ermitteln.

```
SQL> SELECT WRL_PARAMETER FROM V$ENCRYPTION_WALLET;

WRL_PARAMETER
-----
/u01/oracle/product/12.2.0/dbhome_1/data/wallet/
```

Hier ist `/u01/oracle/product/12.2.0/dbhome_1/data/wallet/` der Wallet-Speicherort.

2. Rufen Sie die Masterschlüssel-ID mit einer der folgenden Verschlüsselungsoptionen ab, je nachdem, welche diesen Wert zurückgibt.
 - a. Führen Sie für die Verschlüsselung auf Tabellen- oder Spaltenebene die folgenden Abfragen aus.


```
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+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Hier ist AWGDC9g1Sk8Xv+3bVveiVSg die Masterschlüssel-ID (MKEYID). Wenn Sie einen Wert für MKEYID erhalten, können Sie mit Schritt 3 fortfahren. Fahren Sie andernfalls mit Schritt 2.2 fort.

 Note

Die angehängten Zeichen 'A' in der Zeichenkette (AAA . . .) sind nicht Teil des Werts.

- b. Führen Sie für die Verschlüsselung auf Tablespace-Ebene die folgenden Abfragen aus.

```
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$tablespace t, x$kcbtek x
WHERE t.ts#=x.ts#)
WHERE name = 'TEST_ENCRYPT';

NAME                      MASTERKEYID_BASE64
-----
TEST_ENCRYPT              AWGDC9g1Sk8Xv+3bVveiVSg=
```

Hier ist AWGDC9g1Sk8Xv+3bVveiVSg die Masterschlüssel-ID (TEST_ENCRYPT). Wenn die Schritte 2.1 und 2.2 beide einen Wert zurückgeben, sind diese immer identisch.

Das angehängte Zeichen '=' ist nicht Teil des Werts.

3. Listen Sie über die Befehlszeile die Einträge des Verschlüsselungs-Wallets auf dem Oracle-Quelldatenbank-Host auf.

```
$ 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
```

Suchen Sie den Eintrag, der die Masterschlüssel-ID enthält, die Sie in Schritt 2 (AWGDC9g1Sk8Xv+3bVveiVSg) ermittelt haben. Dieser Eintrag ist der Name des TDE-Verschlüsselungsschlüssels.

4. Sehen Sie sich die Details des Eintrags an, den Sie im vorherigen Schritt ermittelt haben.

```
$ 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==
```

Geben Sie das Wallet-Passwort ein, um das Ergebnis anzuzeigen.

Hier ist der Wert rechts von '=' das TDE-Passwort.

5. Geben Sie den Namen des TDE-Verschlüsselungsschlüssels für den Oracle-Quellendpunkt an, indem Sie das zusätzliche Verbindungsattribut `securityDbEncryptionName` festlegen.

```
securityDbEncryptionName=ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9g1Sk8Xv
+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

6. Geben Sie das zugehörige TDE-Passwort für diesen Schlüssel in der Konsole als Teil des Passwort-Werts der Oracle-Quelle ein. Verwenden Sie die folgende Reihenfolge, um die durch Kommata getrennten Passwortwerte zu formatieren, endend mit dem Wert des TDE-Passworts.

```
Oracle_db_password,ASM_Password,AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3pJC  
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Geben Sie die Passwortwerte in dieser Reihenfolge an, unabhängig von Ihrer Oracle-Datenbankkonfiguration. Wenn Sie beispielsweise TDE verwenden, Ihre Oracle-Datenbank jedoch nicht ASM verwendet, geben Sie die Passwortwerte in der folgenden Reihenfolge an, durch Kommata getrennt.

```
Oracle_db_password, ,AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3pJC  
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Wenn die von Ihnen angegebenen TDE-Anmeldeinformationen falsch sind, schlägt die AWS DMS Migrationsaufgabe nicht fehl. Die Aufgabe liest jedoch auch keine laufenden Replikationsänderungen in der Zieldatenbank und wendet sie auch nicht an. Überwachen Sie nach dem Start der Aufgabe die Tabellenstatistik auf der Konsolenseite zur Migrationsaufgabe, um sicherzustellen, dass die Änderungen repliziert werden.

Wenn ein DBA die TDE-Anmeldeinformationen für die Oracle-Datenbank ändert, während die Aufgabe ausgeführt wird, schlägt die Aufgabe fehl. Die Fehlermeldung enthält den Namen des neuen TDE-Verschlüsselungsschlüssels. Gehen Sie wie oben beschrieben vor, um neue Werte anzugeben und die Aufgabe neu zu starten.

Important

Sie können ein TDE-Wallet, das an einem Oracle Automatic Storage Management (ASM)-Speicherort erstellt wurde, nicht bearbeiten, da Befehle auf Betriebssystemebene wie `cp`, `mv`, `orapki` und `mkstore` die in einem ASM-Speicherort gespeicherten Wallet-Dateien beschädigen. Diese Einschränkung gilt nur für TDE-Wallet-Dateien, die an einem ASM-Speicherort gespeichert sind. Sie gilt nicht für TDE-Wallet-Dateien, die in einem lokalen Betriebssystemverzeichnis gespeichert sind.

Wenn Sie ein in ASM gespeichertes TDE-Wallet mit Befehlen auf Betriebssystemebene bearbeiten möchten, erstellen Sie einen lokalen Keystore und führen Sie den ASM-Keystore wie folgt mit dem lokalen Keystore zusammen:

1. Erstellen Sie einen lokalen Keystore.

```
ADMINISTER KEY MANAGEMENT create keystore file system wallet location
identified by wallet password;
```

2. Führen Sie den ASM-Keystore mit dem lokalen Keystore zusammen.

```
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;
```

Führen Sie dann die Schritte 3 und 4 für den lokalen Keystore aus, um die Einträge des Verschlüsselungs-Wallets und das TDE-Passwort aufzulisten.

Unterstützte Komprimierungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS

In der folgenden Tabelle finden Sie, welche Komprimierungsmethoden bei der Arbeit mit einer Oracle-Quelldatenbank AWS DMS unterstützt werden. Wie die Tabelle zeigt, hängt die Komprimierungsunterstützung sowohl von Ihrer Oracle-Datenbankversion als auch davon ab, ob DMS so konfiguriert ist, dass Oracle für den LogMiner Zugriff auf die Redo-Logs verwendet wird.

Version	Basic	OLTP	HCC (von Oracle 11g R2 oder neuer)	Weitere
Oracle 10	Nein	N/A	N/A	Nein
Oracle 11 oder neuer — Oracle LogMiner	Ja	Ja	Ja	Ja — Jede von Oracle unterstützte Komprimierungsmethode LogMiner.

Version	Basic	OLTP	HCC (von Oracle 11g R2 oder neuer)	Weitere
Oracle 11 oder neuer – Binary Reader	Ja	Ja	Ja – weitere Informationen finden Sie im folgenden Hinweis.	Ja

Note

Wenn der Oracle-Quellendpunkt für die Verwendung von Binary Reader konfiguriert ist, wird die Query Low-Ebene der HCC-Komprimierungsmethode nur für Aufgaben mit Volllast unterstützt.

Replizieren verschachtelter Tabellen unter Verwendung von Oracle als Quelle für AWS DMS

AWS DMS unterstützt die Replikation von Oracle-Tabellen, die Spalten enthalten, bei denen es sich um verschachtelte Tabellen oder definierte Typen handelt. Fügen Sie dem Oracle-Quellendpunkt die folgende zusätzliche Verbindungsattributeinstellung hinzu, um diese Funktion zu aktivieren.

```
allowSelectNestedTables=true;
```

AWS DMS erstellt die Zieltabellen aus verschachtelten Oracle-Tabellen als reguläre über- und untergeordnete Tabellen auf dem Ziel ohne eindeutige Einschränkung. Um auf die richtigen Daten für das Ziel zuzugreifen, verbinden Sie die übergeordneten und untergeordneten Tabellen. Erstellen Sie dazu zunächst manuell einen nicht eindeutigen Index für die NESTED_TABLE_ID-Spalte in der untergeordneten Zieltabelle. Sie können dann die NESTED_TABLE_ID-Spalte in der Join ON-Klausel zusammen mit der übergeordneten Spalte verwenden, die dem Namen der untergeordneten Tabelle entspricht. Darüber hinaus verbessert die Erstellung eines solchen Index die Leistung, wenn die Daten der untergeordneten Zieltabelle aktualisiert oder gelöscht werden AWS DMS. Ein Beispiel finden Sie unter [Beispiel-Join für übergeordnete und untergeordnete Tabellen auf dem Ziel](#).

Es wird empfohlen, die Aufgabe so zu konfigurieren, dass sie beendet wird, nachdem eine vollständige Ladung abgeschlossen ist. Erstellen Sie dann diese nicht eindeutigen Indizes für alle replizierten untergeordneten Tabellen auf dem Ziel, und setzen Sie die Aufgabe fort.

Wenn eine erfasste verschachtelte Tabelle zu einer vorhandenen übergeordneten Tabelle hinzugefügt wird (erfasst oder nicht erfasst), wird sie korrekt AWS DMS behandelt. Der nicht eindeutige Index für die entsprechende Zieltabelle wird jedoch nicht erstellt. In diesem Fall kann die Leistung beeinträchtigt werden, wenn die untergeordnete Zieltabelle extrem groß wird. In einem solchen Fall empfehlen wir, die Aufgabe anzuhalten, den Index zu erstellen und dann die Aufgabe fortzusetzen.

Nachdem die verschachtelten Tabellen auf das Ziel repliziert wurden, muss der DBA einen Join für die übergeordneten und die entsprechenden untergeordneten Tabellen ausführen, um die Daten zu glätten.

Voraussetzungen für die Replikation von verschachtelten Oracle-Tabellen als Quelle

Stellen Sie sicher, dass Sie die übergeordneten Tabellen für alle replizierten verschachtelten Tabellen replizieren. Schließen Sie sowohl die übergeordneten Tabellen (die Tabellen, die die geschachtelte Tabellenspalte enthalten) als auch die untergeordneten (d. h. verschachtelten) Tabellen in die AWS DMS Tabellenzuordnungen ein.

Unterstützte verschachtelte Oracle-Tabellentypen als Quelle

AWS DMS unterstützt die folgenden verschachtelten Oracle-Tabellentypen als Quelle:

- Datentyp
- Benutzerdefiniertes Objekt

Einschränkungen der AWS DMS -Unterstützung für verschachtelte Oracle-Tabellen als Quelle

AWS DMS hat die folgenden Einschränkungen bei der Unterstützung von verschachtelten Oracle-Tabellen als Quelle:

- AWS DMS unterstützt nur eine Ebene der Tabellenverschachtelung.
- AWS DMS Bei der Tabellenzuordnung wird nicht überprüft, ob sowohl die übergeordnete als auch die untergeordnete (n) Tabelle (n) für die Replikation ausgewählt wurden. Dies bedeutet, dass es möglich ist, eine übergeordnete Tabelle ohne untergeordnete Tabelle oder eine untergeordnete Tabelle ohne übergeordnete Tabelle auszuwählen.

Wie AWS DMS verschachtelte Oracle-Tabellen als Quelle repliziert

AWS DMS repliziert übergeordnete und verschachtelte Tabellen wie folgt auf das Ziel:

- AWS DMS erstellt die Elterntabelle, die mit der Quelltable identisch ist. Anschließend definiert es die verschachtelte Spalte im übergeordneten Element als RAW(16) und schließt eine Referenz zu den verschachtelten Tabellen des übergeordneten Elements in seiner NESTED_TABLE_ID-Spalte ein.
- AWS DMS erstellt die untergeordnete Tabelle, die mit der verschachtelten Quelle identisch ist, jedoch mit einer zusätzlichen Spalte namens NESTED_TABLE_ID. Diese Spalte hat denselben Typ und denselben Wert wie die entsprechende übergeordnete verschachtelte Spalte und auch dieselbe Bedeutung.

Beispiel-Join für übergeordnete und untergeordnete Tabellen auf dem Ziel

Um die übergeordnete Tabelle zu glätten, führen Sie einen Join zwischen der übergeordneten und der untergeordneten Tabelle aus, wie im folgenden Beispiel gezeigt:

1. Erstellen Sie die Type-Tabelle.

```
CREATE OR REPLACE TYPE NESTED_TEST_T AS TABLE OF VARCHAR(50);
```

2. Erstellen Sie die übergeordnete Tabelle mit einer Spalte des Typs NESTED_TEST_T, wie oben definiert.

```
CREATE TABLE NESTED_PARENT_TEST (ID NUMBER(10,0) PRIMARY KEY, NAME NESTED_TEST_T)  
  NESTED TABLE NAME STORE AS NAME_KEY;
```

3. Glätten Sie die Tabelle NESTED_PARENT_TEST mit einem Join mit der untergeordneten Tabelle NAME_KEY, wobei CHILD.NESTED_TABLE_ID mit PARENT.NAME übereinstimmt.

```
SELECT ... FROM NESTED_PARENT_TEST PARENT, NAME_KEY CHILD WHERE CHILD.NESTED_  
TABLE_ID = PARENT.NAME;
```

REDO wird auf Oracle ASM gespeichert, wenn Oracle als Quelle für verwendet wird AWS DMS

Bei Oracle-Quellen mit hoher REDO-Generierung kann das Speichern von REDO in Oracle ASM die Leistung verbessern. Dies gilt insbesondere in einer RAC-Konfiguration, da Sie DMS so konfigurieren können, dass ASM-REDO-Lesevorgänge auf alle ASM-Knoten verteilt werden.

Verwenden Sie das Verbindungsattribut `asmServer`, um diese Konfiguration zu verwenden. Die folgende Verbindungszeichenfolge verteilt beispielsweise DMS-REDO-Lesevorgänge auf 3 ASM-Knoten:

```
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)))
```

Wenn Sie NFS zum Speichern von Oracle REDO verwenden, müssen Sie sicherstellen, dass die entsprechenden Client-Patches für DNFS (Direct NFS) installiert sind – insbesondere alle Patches für den Oracle-Fehler 25224242. Weitere Informationen finden Sie in der folgenden Oracle-Veröffentlichung zu Patches im Zusammenhang mit dem Client Direct NFS: [Recommended Patches for Direct NFS Client](#).

Zum Verbessern der NFS-Leseleistung empfehlen wir außerdem, den Wert `rsize` und `wsize` in `fstab` für das NFS-Volumen zu erhöhen, wie im folgenden Beispiel gezeigt.

```
NAS_name_here:/ora_DATA1_archive /u09/oradata/DATA1 nfs
rw,bg,hard,nointr,tcp,nfsvers=3,_netdev,
timeo=600,rsize=262144,wsize=262144
```

Passen Sie den Wert `tcp-max-xfer-size` außerdem wie folgt an:

```
vserver nfs modify -vserver vserver -tcp-max-xfer-size 262144
```


Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer Oracle-Quelldatenbank verwenden. Sie geben die Einstellungen an, wenn Sie den

Quellendpunkt mithilfe der AWS DMS Konsole oder mithilfe des `create-endpoint` Befehls in [AWS CLI](#), mit der `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax erstellen.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Oracle als Quelle verwenden können.


Name	Beschreibung
<code>AccessAlternateDirectly</code>	<p>Legen Sie dieses Attribut auf <code>false</code> fest, um mit Binary Reader Änderungsdaten für Amazon RDS für Oracle als Quelle zu erfassen. Dies weist die DMS-Instance an, dass sie nicht über einen bestimmten Pfadpräfix-Ersatz mittels direktem Dateizugriff auf Redo-Log-Dateien zugreifen darf. Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS.</p> <p>Standardwert: aktiviert</p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--oracle-settings '{"AccessAlternateDirectly": false}'</code></p>
<code>AdditionalArchivedLogDestId</code>	<p>Legen Sie dieses Attribut mit <code>ArchivedLogDestId</code> in einer primären/Standby-Einrichtung fest. Diese Einstellung ist bei einer Umstellung nützlich, wenn eine Datenbank in Oracle Data Guard als Quelle verwendet wird. In diesem Fall AWS DMS muss es wissen, von welchem Ziel die Archiv-Redo-Logs abgerufen werden sollen, um Änderungen zu lesen. Dies liegt daran, dass die vorherige primäre Instance nach der Umstellung eine Standby-Instance ist.</p> <p>AWS DMS unterstützt zwar die Verwendung der <code>RESETLOGS</code> Oracle-Option zum Öffnen der Datenbank, sollte aber <code>RESETLOGS</code> nur verwendet werden, wenn dies erforderlich ist. Weitere Informationen zu <code>RESETLOGS</code> finden Sie unter RMAN Data Repair Concepts im Benutzerhandbuch für Oracle® Database Backup and Recovery.</p> <p>Zulässige Werte: Archivierungsziel-IDs</p>

Name	Beschreibung
	Beispiel: <code>--oracle-settings '{"AdditionalArchivedLogDestId": 2}'</code>
AddSupplementalLogging	<p>Konfigurieren Sie dieses Attribut so, dass für die Oracle-Datenbank die zusätzliche Protokollierung auf Tabellenebene eingerichtet wird. Dieses Attribut aktiviert je nach Tabellenmetadaten eine der folgenden Optionen für alle Tabellen, die für eine Migrationsaufgabe ausgewählt wurden:</p> <ul style="list-style-type: none">• Zusätzliche PRIMARY KEY COLUMNS-Protokollierung• Zusätzliche UNIQUE KEY COLUMNS-Protokollierung• Zusätzliche ALL COLUMNS-Protokollierung <p>Standardwert: false</p> <p>Zulässige Werte: true/false</p> <p>Beispiel: <code>--oracle-settings '{"AddSupplementalLogging": false}'</code></p> <div data-bbox="461 1073 1507 1339" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Wenn Sie diese Option verwenden, müssen Sie dennoch die zusätzliche Protokollierung auf Datenbankebene wie zuvor erläutert aktivieren.</p></div>


Name	Beschreibung
AllowSelectNestedTables	<p>Setzen Sie dieses Attribut auf „true“, um die Replikation von Oracle-Tabellen zu aktivieren, die Spalten enthalten, die verschachtelte Tabellen oder definierte Typen sind. Weitere Informationen finden Sie unter Replizieren verschachtelter Tabellen unter Verwendung von Oracle als Quelle für AWS DMS.</p> <p>Standardwert: false</p> <p>Zulässige Werte: true/false</p> <p>Beispiel: <code>--oracle-settings '{"AllowSelectNestedTables": true}'</code></p>
ArchivedLogDestId	<p>Gibt die ID für die archivierten Redo-Protokolle an. Dieser Wert sollte mit einer Zahl in der Spalte <code>dest_id</code> der Ansicht „<code>v\$archived_log</code>“ identisch sein. Wenn Sie mit einem zusätzlichen Redo-Protokollziel arbeiten, empfehlen wir, das Attribut <code>AdditionalArchivedLogDestId</code> zum Angeben der zusätzlichen Ziel-ID zu verwenden. Dadurch verbessert sich die Performance, indem sichergestellt wird, dass von Anfang an auf die richtigen Protokolle zugegriffen wird.</p> <p>Standardwert: 1</p> <p>Zulässige Werte: Zahl</p> <p>Beispiel: <code>--oracle-settings '{"ArchivedLogDestId": 1}'</code></p>
ArchivedLogsOnly	<p>Wenn dieses Feld auf Y gesetzt ist, wird AWS DMS nur auf die archivierten Redo-Logs zugegriffen. Wenn die archivierten Redo-Logs nur in Oracle ASM gespeichert werden, müssen dem AWS DMS Benutzerkonto ASM-Rechte gewährt werden.</p> <p>Standardwert: N</p> <p>Zulässige Werte: Y/N</p> <p>Beispiel: <code>--oracle-settings '{"ArchivedLogsOnly": Y}'</code></p>

Name	Beschreibung
asmUsePLSQLArray (nur ECA)	<p>Verwenden Sie dieses zusätzliche Verbindungsattribut (ECA), wenn Sie Quelländerungen mit BinaryReader erfassen. Mit dieser Einstellung kann DMS 50 Lesevorgänge auf ASM-Ebene pro individuelm Lese-Thread puffern und gleichzeitig die Anzahl der Threads mit dem Attribut <code>parallelASMReadThread</code> kontrollieren. Wenn Sie dieses Attribut festlegen, verwendet der AWS DMS Binärleser einen anonymen PL/SQL-Block, um Redo-Daten zu erfassen und sie als großen Puffer an die Replikationsinstanz zurückzusenden. Dadurch wird die Anzahl der Round-Trips zur Quelle reduziert. Dies kann die Leistung bei der Erfassung an der Quelle erheblich verbessern, führt jedoch zu einem höheren PGA-Speicherverbrauch auf der ASM-Instance. Wenn das Speicherziel nicht ausreicht, können Stabilitätsprobleme auftreten. Sie können die folgende Formel verwenden, um den gesamten PGA-Speicherverbrauch der ASM-Instance durch eine einzelne DMS-Aufgabe zu schätzen: <code>number_of_redo_threads * parallelASMReadThreads * 7 MB</code></p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>ECA-Beispiel: <code>asmUsePLSQLArray=true;</code></p>
ConvertTimestampWithZoneToUTC	<p>Legen Sie dieses Attribut auf <code>true</code> fest, um den Zeitstempelwert der Spalten „TIMESTAMP WITH TIME ZONE“ und „TIMESTAMP WITH LOCAL TIME ZONE“ in UTC zu konvertieren. Standardmäßig ist der Wert dieses Attributs „<code>false</code>“ und die Daten werden unter Verwendung der Zeitzone der Quelldatenbank repliziert.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--oracle-settings '{"ConvertTimestampWithZoneToUTC": true}'</code></p>

Name	Beschreibung
EnableHomogenousPartitionOps	<p>Legen Sie dieses Attribut auf <code>true</code> fest, um die Replikation von Oracle-Partition- und -SubPartition-DDL-Operationen für die homogene Oracle-Migration zu aktivieren.</p> <p>Beachten Sie, dass diese Funktion und Erweiterung in AWS DMS Version 3.4.7 eingeführt wurden.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--oracle-settings '{"EnableHomogenousPartitionOps": true}'</code></p>
EnableHomogenousTablespace	<p>Setzen Sie dieses Attribut, um die homogene Tablespace-Replikation zu aktivieren und vorhandene Tabellen oder Indizes unter demselben Tablespace auf dem Ziel zu erstellen.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--oracle-settings '{"EnableHomogenousTablespace": true}'</code></p>

Name	Beschreibung
EscapeCharacter	<p>Legen Sie dieses Attribut auf ein Escape-Zeichen fest. Mit diesem Escape-Zeichen können Sie festlegen, dass sich ein einzelnes Platzhalt erzielen in Tabellenzuordnungsausdrücken wie ein normales Zeichen verhält. Weitere Informationen finden Sie unter Platzhalter bei der Tabellenzuordnung.</p> <p>Standardwert: Null</p> <p>Gültige Werte: alle Zeichen außer Platzhalterzeichen</p> <p>Beispiel: <code>--oracle-settings '{"EscapeCharacter": "#"}'</code></p> <div data-bbox="461 730 1508 999" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Sie können nur <code>escapeCharacter</code> für Tabellennamen verwenden. Es werden keine Zeichen aus Schemanamen oder Spaltennamen maskiert.</p></div>
ExposeViews	<p>Verwenden Sie dieses Attribut, um Daten einmal aus einer Ansicht abrufen; Sie können sie nicht für die laufende Replikation verwenden . Wenn Sie Daten aus einer Ansicht extrahieren, wird die Ansicht im Zielschema als Tabelle angezeigt.</p> <p>Standardwert: false</p> <p>Zulässige Werte: true/false</p> <p>Beispiel: <code>--oracle-settings '{"ExposeViews": true}'</code></p>

Name	Beschreibung
ExtraArchivedLogDestIds	<p>Gibt die IDs eines weiteren Ziels für ein oder mehrere archivierte Redo-Protokolle an. Diese IDs sind die Werte der Spalte <code>dest_id</code> in der Ansicht <code>v\$archived_log</code>. Verwenden Sie diese Einstellung zusammen mit dem <code>ArchivedLogDestId</code> zusätzlichen Verbindungsattribut in einem <code>primary-to-single</code> Setup oder einem <code>primary-to-multiple-standby</code> Setup.</p> <p>Diese Einstellung ist bei einem Switchover nützlich, wenn Sie eine Oracle-Data-Guard-Datenbank als Quelle verwenden. Benötigt in diesem Fall Informationen darüber, AWS DMS von welchem Ziel Archiv-Redo-Logs abgerufen werden sollen, um Änderungen zu lesen. AWS DMS benötigt dies, da die vorherige Primärinstanz nach dem Switchover eine Standby-Instanz ist.</p> <p>Zulässige Werte: Archivierungsziel-IDs</p> <p>Beispiel: <code>--oracle-settings '{"ExtraArchivedLogDestIds": 1}'</code></p>
FailTasksOnLobTruncation	<p>Mit der Einstellung <code>true</code> schlägt eine Aufgabe durch dieses Attribut fehl, wenn die tatsächliche Größe einer LOB-Spalte größer als der angegebene <code>LobMaxSize</code> -Wert ist.</p> <p>Wenn die Aufgabe auf den eingeschränkten LOB-Modus festgelegt ist und diese Option auf <code>true</code> festgelegt wurde, schlägt die Aufgabe fehl, anstatt LOB-Daten zu kürzen.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: Boolesch</p> <p>Beispiel: <code>--oracle-settings '{"FailTasksOnLobTruncation": true}'</code></p>

Name	Beschreibung
<code>filterTransactionsOfUser</code> (nur ECA)	<p>Verwenden Sie dieses zusätzliche Verbindungsattribut (ECA), damit DMS Transaktionen eines bestimmten Benutzers ignorieren kann, wenn Daten aus Oracle repliziert werden, wenn Sie LogMiner Sie können durch Kommata getrennte Benutzernamenwerte übergeben, diese müssen jedoch ausschließlich in GROSSBUCHSTABEN geschrieben sein.</p> <p>ECA-Beispiel: <code>filterTransactionsOfUser= <i>USERNAME</i>;</code></p>
<code>NumberDataTypeScale</code>	<p>Gibt den Skalierwert an. Sie können eine Skalierung von bis zu 38, -1 für FLOAT oder -2 für VARCHAR auswählen. Standardmäßig wird der Datentyp NUMBER in Präzision = 38 und Skalierung = 10 konvertiert.</p> <p>Standardwert: 10</p> <p>Zulässige Werte: -2 bis 38 (-2 für VARCHAR, -1 für FLOAT)</p> <p>Beispiel: <code>--oracle-settings '{"NumberDataTypeScale": 12}'</code></p> <div data-bbox="461 1037 1507 1497" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Wählen Sie eine Kombination aus Präzision und Skalierung, -1 (FLOAT) oder -2 (VARCHAR) aus. DMS unterstützt jede von Oracle unterstützte Kombination aus Präzision und Skalierung. Wählen Sie bei einer Präzision von 39 oder höher -2 (VARCHAR) aus. Die <code>NumberDataTypeScale</code> Einstellung für die Oracle-Datenbank wird nur für den NUMBER-Datentyp verwendet (ohne die explizite Präzisions- und Skalendefinition).</p> </div>
<code>OpenTransactionWindow</code>	<p>Gibt den Zeitrahmen in Minuten an, in dem nach offenen Transaktionen für reine CDC-Aufgaben gesucht wird.</p> <p>Standardwert: 0</p> <p>Gültige Werte: eine Ganzzahl von 0 bis 240</p> <p>Beispiel: <code>openTransactionWindow=15;</code></p>

Name	Beschreibung
OraclePathPrefix	<p>Legen Sie dieses Zeichenfolge-Attribut auf den erforderlichen Wert fest, um mit dem Binary Reader Änderungsdaten für ein Amazon RDS für Oracle als Quelle zu erfassen. Dieser Wert gibt das Standard-Oracle-Root an, das für den Zugriff auf die Redo-Log-Dateien verwendet wird. Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS.</p> <p>Standardwert: keiner</p> <p>Zulässiger Wert: /rdsdbdata/db/ORCL_A/</p> <p>Beispiel: <code>--oracle-settings '{"OraclePathPrefix": " /rdsdbdata/db/ORCL_A/"}'</code></p>
ParallelASMRReadThreads	<p>Legen Sie dieses Attribut fest, um die Anzahl der Threads zu ändern, die DMS konfiguriert, um CDC mit Oracle Automatic Storage Management (ASM) durchzuführen. Sie können einen Ganzzahlwert zwischen 2 (Standard) und 8 (maximal) angeben. Verwenden Sie dieses Attribut zusammen mit dem ReadAheadBlocks -Attribut. Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS.</p> <p>Standardwert: 2</p> <p>Gültige Werte: Eine Ganzzahl von 2 bis 8</p> <p>Beispiel: <code>--oracle-settings '{"ParallelASMRReadThreads": 6;}'</code></p>

Name	Beschreibung
ReadAheadBlocks	<p>Legen Sie dieses Attribut fest, um die Anzahl der Read-Ahead-Blöcke zu ändern, die DMS konfiguriert, um CDC mit Oracle Automatic Storage Management (ASM) und NAS-Speicher ohne ASM durchzuführen. Sie können einen Ganzzahlwert zwischen 1 000 (Standard) und 200 000 (maximal) angeben. Verwenden Sie dieses Attribut zusammen mit dem <code>ParallelASMReadThreads</code> -Attribut. Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS.</p> <p>Standardwert: 1000</p> <p>Gültige Werte: Eine Ganzzahl von 1000 bis 200 000</p> <p>Beispiel: <code>--oracle-settings '{"ReadAheadBlocks": 150000}'</code></p>
ReadTableSpaceName	<p>Bei der Einstellung <code>true</code> unterstützt dieses Attribut eine Replikation der Tabellenräume.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: Boolesch</p> <p>Beispiel: <code>--oracle-settings '{"ReadTableSpaceName": true}'</code></p>

Name	Beschreibung
ReplacePathPrefix	<p>Legen Sie dieses Attribut auf True fest, um mit dem Binary Reader Änderungsdaten für ein Amazon RDS für Oracle als Quelle zu erfassen. Diese Einstellung weist die DMS-Instance an, das Standard Oracle-Root durch die angegebene UsePathPrefix -Einstellung zu ersetzen, um auf die Redo-Log-Dateien zuzugreifen. Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS.</p> <p>Standardwert: false</p> <p>Zulässige Werte: true/false</p> <p>Beispiel: <code>--oracle-settings '{"ReplacePathPrefix": true}'</code></p>
RetryInterval	<p>Gibt die Anzahl der Sekunden an, die das System wartet, bevor eine Abfrage erneut gesendet wird.</p> <p>Standardwert: 5</p> <p>Zulässige Werte: Zahl ab 1</p> <p>Beispiel: <code>--oracle-settings '{"RetryInterval": 6}'</code></p>

Name	Beschreibung
SecurityDbEncryptionName	<p>Gibt den Namen eines Schlüssels an, der für die transparente Datenverschlüsselung (TDE) der Spalten und des Tablespace in der Oracle-Quelldatenbank verwendet wird. Weitere Hinweise zum Einstellen dieses Attributs und des zugehörigen Passworts auf dem Oracle-Quellendpunkt finden Sie unter Unterstützte Verschlüsselungsmethoden für die Verwendung von Oracle als Quelle für AWS DMS.</p> <p>Standardwert: ""</p> <p>Zulässige Werte: String</p> <p>Beispiel: <code>--oracle-settings '{"SecurityDbEncryptionName": "ORACLE.SECURITY.DB.ENCRYPTION.Adg8m2dhkU/0v/m5QUaaNJEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}'</code></p>
SpatialSdo2GeoJsonFunctionName	<p>Verwenden Sie für die Migration von Oracle-Version 12.1 oder früheren Quellen zu PostgreSQL-Zielen dieses Attribut, um SDO_GEOMETRY zum GEOJSON-Format zu konvertieren.</p> <p>AWS DMS ruft standardmäßig die <code>SDO2GEOJSON</code> benutzerdefinierte Funktion auf, die vorhanden und für den AWS DMS Benutzer zugänglich sein muss. Oder Sie können eine eigene benutzerdefinierte Funktion erstellen, die die Funktionsweise von <code>SDO2GEOJSON</code> nachahmt, und stattdessen <code>SpatialSdo2GeoJsonFunctionName</code> zum Aufruf einrichten.</p> <p>Standardwert: <code>SDO2GEOJSON</code></p> <p>Zulässige Werte: String</p> <p>Beispiel: <code>--oracle-settings '{"SpatialSdo2GeoJsonFunctionName": "myCustomSDO2GEOJSONFunction"}'</code></p>

Name	Beschreibung
StandbyDelayTime	<p>Mit diesem Attribut wird eine Zeit in Minuten für die Verzögerung bei der Standby-Synchronisierung festgelegt. Wenn die Quelle eine Standby-Datenbank von Active Data Guard ist, verwenden Sie dieses Attribut, um die Zeitverzögerung zwischen Primär- und Standby-Datenbanken anzugeben.</p> <p>In AWS DMS können Sie eine Oracle CDC-Aufgabe erstellen, die eine Active Data Guard-Standby-Instanz als Quelle für die Replikation laufender Änderungen verwendet. Somit ist es nicht mehr notwendig, eine Verbindung mit einer aktiven Datenbank herzustellen, die sich möglicherweise in der Produktion befindet.</p> <p>Standardwert: 0</p> <p>Zulässige Werte: Zahl</p> <p>Beispiel: <code>--oracle-settings '{"StandbyDelayTime": 1}'</code></p> <p>Hinweis: Bei Verwendung von DMS 3.4.6, 3.4.7 und höher ist die Verwendung dieser Verbindungseinstellung optional. In der neuesten Version von DMS 3.4.6 und Version 3.4.7 sollte <i>dms_user</i> über die Berechtigung <code>select</code> für <code>V_\$DATAGUARD_STATS</code> verfügen, damit DMS die Standby-Verzögerungszeit berechnen kann.</p>

Name	Beschreibung
<code>UseAlternateFolderForOnline</code>	<p>Legen Sie dieses Attribut auf <code>True</code> fest, um mit dem Binary Reader Änderungsdaten für ein Amazon RDS für Oracle als Quelle zu erfassen. Dies weist die DMS-Instance an, einen angegebenen Präfix-Ersatz zu verwenden, um auf alle Online-Redo-Log-Dateien zuzugreifen. Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--oracle-settings '{"UseAlternateFolderForOnline": true}'</code></p>
<code>UseBfile</code>	<p>Legen Sie für dieses Attribut <code>"Y"</code> fest, um Änderungen mit dem Dienstprogramm Binary Reader zu erfassen. Legen Sie <code>UseLogminerReader</code> auf <code>N</code> fest, um dieses Attribut auf <code>Y</code> festzulegen. Sie müssen zusätzliche Attribute festlegen, um Binary Reader mit Amazon RDS für Oracle als Quelle verwenden zu können. Weitere Informationen zu dieser Einstellung und der Verwendung von Oracle Automatic Storage Management (ASM) finden Sie unter Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC.</p> <p>Hinweis: Wenn Sie diesen Wert als zusätzliches Verbindungsattribut (ECA) festlegen, lauten die gültigen Werte „Y“ und „N“. Wenn Sie diesen Wert als Endpunkteinstellung festlegen, lauten die gültigen Werte <code>true</code> und <code>false</code>.</p> <p>Standardwert: <code>N</code></p> <p>Gültige Werte: <code>Y/N</code> (wenn dieser Wert als ECA festgelegt wird); <code>true/false</code> (wenn dieser Wert als Endpunkteinstellung festgelegt wird).</p> <p>Beispiel: <code>--oracle-settings '{"UseBfile": Y}'</code></p>

Name	Beschreibung
UseLogminerReader	<p>Setzen Sie dieses Attribut auf Y, um Änderungsdaten mit dem LogMiner Hilfsprogramm zu erfassen (Standard). Legen Sie für diese Option „N“ fest, wenn Sie möchten, dass AWS DMS auf die Wiederholungsprotokolle als Binärdatei zugreift. Wenn Sie diese Option auf N festlegen, fügen Sie auch die Einstellung useBfile=Y hinzu. Weitere Informationen zu dieser Einstellung und zur Verwendung von Oracle Automatic Storage Management (ASM) finden Sie unter Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC.</p> <p>Hinweis: Wenn Sie diesen Wert als zusätzliches Verbindungsattribut (ECA) festlegen, lauten die gültigen Werte „Y“ und „N“. Wenn Sie diesen Wert als Endpunkteinstellung festlegen, lauten die gültigen Werte true und false.</p> <p>Standardwert: Y</p> <p>Gültige Werte: Y/N (wenn dieser Wert als ECA festgelegt wird); true/false (wenn dieser Wert als Endpunkteinstellung festgelegt wird).</p> <p>Beispiel: <code>--oracle-settings '{"UseLogminerReader": Y}'</code></p>
UsePathPrefix	<p>Legen Sie dieses Zeichenfolge-Attribut auf den erforderlichen Wert fest, um mit dem Binary Reader Änderungsdaten für ein Amazon RDS für Oracle als Quelle zu erfassen. Dieser Wert gibt das Pfadpräfix an, das verwendet wird, um das Standard-Oracle-Root für den Zugriff auf die Redo-Log-Dateien zu ersetzen. Weitere Informationen finden Sie unter Konfiguration einer CDC-Aufgabe für die Verwendung von Binary Reader mit einer RDS for Oracle-Quelle für AWS DMS.</p> <p>Standardwert: keiner</p> <p>Zulässiger Wert: /rdsdbdata/log/</p> <p>Beispiel: <code>--oracle-settings '{"UsePathPrefix": " /rdsdbdata/log/ "'}</code></p>

Quelldatentypen für Oracle

Der Oracle-Endpunkt für AWS DMS unterstützt die meisten Oracle-Datentypen. Die folgende Tabelle zeigt die Oracle-Quelldatentypen, die bei der Verwendung unterstützt werden, AWS DMS und die Standardzuordnungen zu AWS DMS Datentypen.

Note

Mit Ausnahme der Datentypen LONG und LONG RAW sind bei der Replikation von einer Oracle-Quelle auf ein Oracle-Ziel (homogene Replikation) alle Quell- und Zieldatentypen identisch. Der Datentyp LONG wird jedoch CLOB zugeordnet, der Datentyp LONG RAW wird BLOB zugeordnet.

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

Oracle-Datentyp	AWS DMS Datentyp
BINARY_FLOAT	REAL4
BINARY_DOUBLE	REAL8
BINARY	BYTES
FLOAT (P)	Wenn die Präzision kleiner oder gleich 24 ist, verwenden Sie REAL4. Wenn die Präzision größer als 24 ist, verwenden Sie REAL8.
NUMBER (P,S)	Verwenden Sie bei einer Skalierung größer als 0 NUMERIC. Bei einer Skalierung von 0: <ul style="list-style-type: none"> • und Präzision kleiner als oder gleich 2, verwenden Sie INT1. • und Präzision größer als 2 und kleiner oder gleich 4, verwenden Sie INT2.

Oracle-Datentyp	AWS DMS Datentyp
	<ul style="list-style-type: none"> • und Präzision größer als 4 und kleiner oder gleich 9, verwenden Sie INT4. • und Präzision größer als 9, verwenden Sie NUMERIC. • und Präzision größer oder gleich der Skalierung, verwenden Sie NUMERIC. <p>Verwenden Sie bei einer Skalierung kleiner als 0 REAL8.</p>
DATUM	DATETIME
INTERVAL_YEAR TO MONTH	STRING (mit Angabe von "interval year_to_month")
INTERVAL_DAY TO SECOND	STRING (mit Angabe von "interval day_to_second")
TIMESTAMP (ZEITSTEMPEL)	DATETIME
TIMESTAMP WITH TIME ZONE	STRING (mit Angabe von "timestamp_with_timezone")
TIMESTAMP WITH LOCAL TIME ZONE	STRING (mit Angabe von "timestamp_with_local_timezone")
CHAR	STRING
VARCHAR2	STRING
NCHAR	WSTRING
NVARCHAR2	WSTRING
RAW	BYTES
REAL	REAL8

Oracle-Datentyp	AWS DMS Datentyp
BLOB	<p>BLOB</p> <p>Um diesen Datentyp mit verwenden zu können AWS DMS, müssen Sie die Verwendung von BLOB-Datentypen für eine bestimmte Aufgabe aktivieren. AWS DMS unterstützt BLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>
CLOB	<p>CLOB</p> <p>Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von CLOB-Datentypen für eine bestimmte Aufgabe aktivieren. AWS DMS Unterstützt während CDC CLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>
NCLOB	<p>NCLOB</p> <p>Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von NCLOB-Datentypen für eine bestimmte Aufgabe aktivieren. AWS DMS Unterstützt während CDC NCLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>


Oracle-Datentyp	AWS DMS Datentyp
LONG	<p data-bbox="542 226 634 260">CLOB</p> <p data-bbox="542 306 1484 390">Der LONG-Datentyp wird im stapeloptimierten Anwendungsmodus (CDC-Modus) nicht unterstützt. TurboStream</p> <p data-bbox="542 436 1463 520">Um diesen Datentyp mit zu verwenden AWS DMS, aktivieren Sie die Verwendung von LOBs für eine bestimmte Aufgabe.</p> <p data-bbox="542 562 1474 646">AWS DMS Unterstützt LOB-Datentypen bei CDC oder Volllast nur in Tabellen, die über einen Primärschlüssel verfügen.</p> <p data-bbox="542 688 1507 1108">Unterstützt auch AWS DMS nicht den vollständigen LOB-Modus zum Laden von LONG-Spalten. Stattdessen können Sie den limitierten LOB-Modus verwenden, um LONG-Spalten zu einem Oracle-Ziel zu migrieren. Im eingeschränkten LOB-Modus werden alle Daten, die Sie AWS DMS auf LONG-Spalten mit einer Länge von mehr als 64 KB festgelegt haben, auf 64 KB gekürzt. Weitere Informationen zur LOB-Unterstützung finden Sie unter AWS DMS <u>Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS</u></p>

Oracle-Datentyp	AWS DMS Datentyp
LONG RAW	<p>BLOB</p> <p>Der LONG RAW-Datentyp wird im stapeloptimierten Anwendungsmodus (TurboStream CDC-Modus) nicht unterstützt.</p> <p>Um diesen Datentyp mit zu verwenden AWS DMS, aktivieren Sie die Verwendung von LOBs für eine bestimmte Aufgabe.</p> <p>AWS DMS Unterstützt LOB-Datentypen bei CDC oder Volllast nur in Tabellen, die über einen Primärschlüssel verfügen.</p> <p>Unterstützt auch AWS DMS nicht den vollständigen LOB-Modus zum Laden von LONG RAW-Spalten. Stattdessen können Sie den limitierten LOB-Modus verwenden, um LONG RAW-Spalten zu einem Oracle-Ziel zu migrieren. Im eingeschränkten LOB-Modus werden alle Daten, die Sie AWS DMS auf LONG RAW-Spalten mit einer Länge von mehr als 64 KB festgelegt haben, auf 64 KB gekürzt. Weitere Informationen zur LOB-Unterstützung finden Sie unter AWS DMS <u>Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS</u></p>
XMLTYPE	CLOB
SDO_GEOMETRY	<p>BLOB (bei einer Migration von Oracle zu Oracle)</p> <p>CLOB (bei einer Migration von Oracle zu PostgreSQL)</p>

Oracle-Tabellen, die als Quelle mit Spalten der folgenden Datentypen verwendet werden, werden nicht unterstützt und können nicht repliziert werden. Das Replizieren von Spalten mit diesen Datentypen führt zu einer Null-Spalte.

- BFILE
- ROWID
- REF
- UROWID
- Benutzerdefinierte Datentypen

- ANYDATA
- VARRAY

 Note

Virtuelle Spalten werden nicht unterstützt.

Migrieren räumlicher Oracle-Datentypen

Räumliche Daten identifizieren die Geometrieinformationen für ein Objekt oder eine Position im Raum. In einer Oracle-Datenbank wird die geometrische Beschreibung eines räumlichen Objekts in einem Objekt des Typs SDO_GEOMETRY gespeichert. Innerhalb dieses Objekts wird die geometrische Beschreibung in einer einzelnen Zeile in einer einzelnen Spalte einer benutzerdefinierten Tabelle gespeichert.

AWS DMS unterstützt die Migration des Oracle-Typs SDO_GEOMETRY von einer Oracle-Quelle zu einem Oracle- oder PostgreSQL-Ziel.

Beachten Sie bei der Migration von Oracle-Geodatentypen mit folgenden Überlegungen AWS DMS:

- Stellen Sie bei der Migration zu einem Oracle-Ziel sicher, dass USER_SDO_GEOM_METADATA-Einträge, die Typinformationen enthalten, manuell übertragen werden.
- AWS DMS Erstellt bei der Migration von einem Oracle-Quellendpunkt zu einem PostgreSQL-Zielendpunkt Zielspalten. Diese Spalten verfügen über Standardgeometrie- und Geographie-Typinformationen mit einer 2D-Dimension und einer Raumbezugskenung (SRID) gleich Null (0). Ein Beispiel ist GEOMETRY, 2, 0.
- Konvertieren Sie bei Oracle-Version 12.1 oder früheren zu PostgreSQL-Zielen migrierten Quellen SDO_GEOMETRY-Objekte mithilfe der SD02GE0JSON-Funktion oder des zusätzlichen Verbindungsattributs `spatialSdo2GeoJsonFunctionName` in das GE0JSON-Format. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).
- AWS DMS unterstützt Oracle Spatial-Spaltenmigrationen nur für den vollständigen LOB-Modus. AWS DMS unterstützt die Modi Limited LOB oder Inline LOB nicht. Weitere Informationen zum LOB-Modus finden Sie unter [Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS](#).

- Da AWS DMS nur der vollständige LOB-Modus für die Migration von Oracle Spatial-Spalten unterstützt wird, benötigt die Tabelle der Spalten einen Primärschlüssel und einen eindeutigen Schlüssel. Wenn die Tabelle keinen Primärschlüssel und keinen eindeutigen Schlüssel aufweist, wird die Tabelle bei der Migration übersprungen.

Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS

Migrieren Sie Daten aus einer oder vielen Microsoft SQL Server-Datenbanken mit AWS DMS. Mit einer SQL Server-Datenbank als Quelle können Sie Daten zu einer anderen SQL Server-Datenbank oder zu einer der anderen AWS DMS unterstützten Datenbanken migrieren.

Informationen zu Versionen von SQL Server, die als Quelle AWS DMS unterstützt, finden Sie unter [Quellen für AWS DMS](#).

Die SQL Server-Quelldatenbank kann auf einem beliebigen Computer in Ihrem Netzwerk installiert sein. Ein SQL Server-Konto mit den entsprechenden Zugriffsberechtigungen für die Quelldatenbank für den Typ der ausgewählten Aufgabe ist für die Verwendung mit AWS DMS erforderlich. Dieses Konto muss über die Berechtigungen `view definition` und `view server state` verfügen. Sie fügen diese Berechtigung mit dem folgenden Befehl hinzu:

```
grant view definition to [user]
grant view server state to [user]
```

AWS DMS unterstützt die Migration von Daten von benannten Instances von SQL Server. Beim Erstellen des Quellendpunkts können Sie folgende Notation für den Servernamen verwenden.

```
IPAddress\InstanceName
```

Im folgenden Beispiel wird ein korrekter Servername für den Quellendpunkt angegeben. Der erste Teil des Namens ist die IP-Adresse des Servers und der zweite Teil ist der Name der SQL Server-Instance (hier "SQLTest").

```
10.0.0.25\SQLTest
```

Rufen Sie außerdem die Portnummer ab, die Ihre benannte Instance von SQL Server überwacht, und verwenden Sie sie, um Ihren AWS DMS Quellendpunkt zu konfigurieren.

Note

Port 1433 ist der Standard für Microsoft SQL Server. Jedoch werden dynamische Ports, die sich bei jedem Start von SQL Server ändern, und spezifische statische Portnummern, die zum Herstellen einer Verbindung mit SQL Server über eine Firewall genutzt werden, ebenfalls häufig verwendet. Sie möchten also die tatsächliche Portnummer Ihrer benannten Instance von SQL Server kennen, wenn Sie den AWS DMS Quellendpunkt erstellen.

Sie können SSL verwenden, um Verbindungen zwischen Ihrem SQL Server-Endpunkt und der Replikations-Instance zu verschlüsseln. Weitere Informationen zur Verwendung von SSL mit einem SQL Server-Endpunkt finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).

Weitere Informationen zum Arbeiten mit SQL Server-Quelldatenbanken und AWS DMS finden Sie im Folgenden.

Themen

- [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#)
- [Berechtigungen für reine Volllastaufgaben](#)
- [Voraussetzungen für die Verwendung der laufenden Replikation \(CDC\) von einer SQL-Server-Quelle aus](#)
- [Erfassen von Datenänderungen für selbstverwaltete SQL-Server-Quellen \(On-Premises oder in Amazon EC2\)](#)
- [Einrichten der laufenden Replikation auf einer SQL-Server-DB-Instance in der Cloud](#)
- [Empfohlene Einstellungen bei Verwendung von Amazon RDS für SQL Server als Quelle für AWS DMS](#)
- [Unterstützte Komprimierungsmethoden für SQL Server](#)
- [Arbeiten mit selbstverwalteten SQL Server AlwaysOn-Verfügbarkeitsgruppen](#)
- [Sicherheitsanforderungen bei Verwendung von SQL Server als Quelle für AWS Database Migration Service](#)
- [Endpunkteinstellungen bei Verwendung von SQL Server als Quelle für AWS DMS](#)
- [Quelldatentypen für SQL Server](#)

Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS

Die folgenden Einschränkungen gelten, wenn Sie eine SQL Server-Datenbank als Quelle für AWS DMS verwenden:

- Die Identitätseigenschaft für eine Spalte wird nicht zu einer Spalte in der Zieldatenbank migriert.
- Der SQL Server-Endpoint unterstützt nicht die Verwendung von Tabellen mit Sparse-Spalten.
- Windows-Authentifizierung wird nicht unterstützt.
- Änderungen an berechneten Feldern in einem SQL Server werden nicht repliziert.
- Temporäre Tabellen werden nicht unterstützt.
- SQL Server-Partition-Switching wird nicht unterstützt.
- Bei Verwendung der Dienstprogramme WRITETEXT und UPDATETEXT erfasst AWS DMS keine Ereignisse, die auf die Quelldatenbank angewendet werden.
- Das folgende Data Manipulation Language (DML)-Muster wird nicht unterstützt.

```
SELECT * INTO new_table FROM existing_table
```

- Bei der Verwendung von SQL Server als Quelle wird die Verschlüsselung auf Spaltenebene nicht unterstützt.
- AWS DMS unterstützt keine Audits auf Serverebene auf SQL Server 2008 oder SQL Server 2008 R2 als Quellen. Dies liegt an einem bekannten Problem mit SQL Server 2008 und 2008 R2. Wenn Sie beispielsweise den folgenden Befehl ausführen, schlägt AWS DMS fehl.

```
USE [master]
GO
ALTER SERVER AUDIT [my_audit_test-20140710] WITH (STATE=on)
GO
```

- Bei Verwendung von SQL Server als Quelle werden Geometriespalten im vollständigen LOB-Modus nicht unterstützt. Verwenden Sie stattdessen den limitierten LOB-Modus oder legen Sie die Aufgabeneinstellung `InLineLobMaxSize` so fest, dass der Inline-LOB-Modus verwendet wird.
- Wenn Sie eine Quelldatenbank in Microsoft SQL Server in einer Replikationsaufgabe verwenden, werden die Definitionen von SQL Server Replication Publisher nicht entfernt, falls Sie die Aufgabe entfernen. Ein Microsoft SQL Server-Systemadministrator muss diese Definitionen von Microsoft SQL Server löschen.
- Die Migration von Daten aus schemagebundenen non-schema-bound Ansichten wird nur für Volllastaufgaben unterstützt.

- Das Umbenennen von Tabellen mit `sp_rename` wird nicht unterstützt (z. B. `sp_rename 'Sales.SalesRegion', 'SalesReg';`)
- Das Umbenennen von Spalten mit `sp_rename` wird nicht unterstützt (z. B. `sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';`)
- AWS DMS unterstützt keine Änderungsverarbeitung zum Festlegen und Aufheben von Spaltenstandardwerten (mit der `-ALTER COLUMN SET DEFAULT`-Klausel mit `-ALTER TABLE`-Anweisungen).
- AWS DMS unterstützt keine Änderungsverarbeitung zum Festlegen der Spaltenlöschbarkeit (mit der `-ALTER COLUMN [SET|DROP] NOT NULL`-Klausel mit `-ALTER TABLE`-Anweisungen).
- Bei SQL Server 2012 und SQL Server 2014 kann die Verteilungsdatenbank bei Verwendung der DMS-Replikation mit Verfügbarkeitsgruppen nicht in einer Verfügbarkeitsgruppe platziert werden. SQL 2016 unterstützt das Platzieren der Verteilungsdatenbank in einer Verfügbarkeitsgruppe, mit Ausnahme von Verteilungsdatenbanken, die in Zusammenführungs-, bidirektionalen oder peer-to-peer Replikationstopologien verwendet werden.
- Für partitionierte Tabellen unterstützt keine AWS DMS unterschiedlichen Datenkomprimierungseinstellungen für jede Partition.
- Wenn Sie einen Wert in räumliche SQL-Server-Datentypen (GEOGRAPHY und GEOMETRY) einfügen, können Sie entweder die Eigenschaft Spatial Reference System Identifier (SRID) ignorieren oder eine andere Zahl angeben. Beim Replizieren von Tabellen mit räumlichen Datentypen ersetzt die SRID AWS DMS durch die Standard-SRID (0 für GEOMETRY und 4326 für GEOGRAPHY).
- Wenn Ihre Datenbank nicht für MS-REPLICATION oder MS-CDC konfiguriert ist, können Sie dennoch Tabellen erfassen, die keinen Primärschlüssel haben, es werden aber nur INSERT/DELETE DML-Ereignisse erfasst. Die Ereignisse UPDATE und TRUNCATE TABLE werden ignoriert.
- Columnstore-Indizes werden nicht unterstützt.
- Speicheroptimierte Tabellen (unter Verwendung von In-Memory OLTP) werden nicht unterstützt.
- Wenn Sie eine Tabelle mit einem Primärschlüssel replizieren, der aus mehreren Spalten besteht, wird das Aktualisieren der Primärschlüsselspalten während eines Volllastvorgangs nicht unterstützt.
- Verzögerte Haltbarkeit wird nicht unterstützt.
- Aufgrund der Art und Weise, wie RDS Backups durchführt, funktioniert die Endpunkteinstellung `readBackupOnly=Y` (zusätzliches Verbindungsattribut) auf Quell-Instances in RDS für SQL Server nicht.

- EXCLUSIVE_AUTOMATIC_TRUNCATION funktioniert auf Quell-Instances in Amazon RDS SQL Server nicht, da RDS-Benutzer keinen Zugriff zum Ausführen der gespeicherten SQL-Server-Prozedur sp_repldone haben.
- AWS DMS erfasst keine Kürzungsbefehle.
- AWS DMS unterstützt keine Replikation aus Datenbanken mit aktivierter beschleunigter Datenbankwiederherstellung (ADR).
- AWS DMS unterstützt nicht die Erfassung von Data Definition Language (DDL)- und Data Manipulation Language (DML)-Anweisungen innerhalb einer einzigen Transaktion.
- AWS DMS unterstützt nicht die Replikation von Anwendungspaketen auf Datenebene (DACPAC).
- UPDATE-Anweisungen, die Primärschlüssel oder eindeutige Indizes beinhalten und mehrere Datenzeilen aktualisieren, können zu Konflikten führen, wenn Sie Änderungen auf die Zieldatenbank anwenden. Dies kann beispielsweise der Fall sein, wenn die Zieldatenbank Aktualisierungen als INSERT- und DELETE-Anweisungen anstelle einer einzigen UPDATE-Anweisung anwendet. Im stapeloptimierten Anwendungsmodus wird die Tabelle möglicherweise ignoriert. Im transaktionalen Anwendungsmodus kann die UPDATE-Operation zu Verstößen gegen die Einschränkung führen. Laden Sie die entsprechende Tabelle neu, um dieses Problem zu vermeiden. Suchen Sie alternativ die problematischen Datensätze in der Kontrolltabelle „Ausnahmen anwenden“ (dmslogs.aws_dms_apply_exceptions) und bearbeiten Sie sie manuell in der Zieldatenbank. Weitere Informationen finden Sie unter [Einstellungen für die Optimierung der Verarbeitung von Änderungen](#).
- AWS DMS unterstützt nicht die Replikation von Tabellen und Schemata, wobei der Name ein Sonderzeichen aus dem folgenden Satz enthält.

```
\\ -- \n \" \b \r ' \t ;
```

- Die Datenmaskierung wird nicht unterstützt. AWS DMS migriert maskierte Daten ohne Maskierung.
- AWS DMS repliziert bis zu 32 767 Tabellen mit Primärschlüsseln und bis zu 1 000 Spalten pro Tabelle. Dies liegt daran, dass einen SQL Server-Replikationsartikel für jede replizierte Tabelle AWS DMS erstellt und SQL Server-Replikationsartikel diese Einschränkungen haben.
- Wenn Sie die Erfassung von Datenänderungen (Change Data Capture, CDC) verwenden, müssen Sie alle Spalten, die einen eindeutigen Index bilden, als NOT NULL definieren. Wenn diese Anforderung nicht erfüllt ist, führt dies zum SQL-Server-Systemfehler 22838.

Beim Zugriff auf die Sicherungstransaktionsprotokolle gelten die folgenden Einschränkungen:

- Verschlüsselte Sicherungen werden nicht unterstützt.

- Sicherungen, die unter einer URL oder unter Windows Azure gespeichert sind, werden nicht unterstützt.
- AWS DMS unterstützt keine direkte Verarbeitung von Transaktionsprotokoll-Backups auf Dateiebene aus alternativen freigegebenen Ordnern.

Berechtigungen für reine Volllastaufgaben

Die folgenden Berechtigungen sind zum Ausführen reiner Volllastaufgaben erforderlich. Beachten Sie, dass die `dms_user` Anmeldung AWS DMS nicht erstellt. Informationen zum Erstellen eines Login für SQL Server finden Sie unter [Erstellen eines Datenbankbenutzers mit 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;
```

Voraussetzungen für die Verwendung der laufenden Replikation (CDC) von einer SQL-Server-Quelle aus

Sie können die laufende Replikation (Erfassung von Datenänderungen (Change Data Capture, CDC)) für eine selbstverwaltete SQL-Server-Datenbank On-Premises oder in Amazon EC2 oder für eine Cloud-Datenbank wie Amazon RDS oder eine von Microsoft Azure SQL verwaltete Instance verwenden.

Die folgenden Anforderungen gelten speziell für die Verwendung der fortlaufenden Replikation mit einer SQL Server-Datenbank als Quelle für AWS DMS:

- SQL Server muss für vollständige Sicherungen konfiguriert werden und vor Beginn der Datenreplikation muss eine Sicherung ausgeführt werden.
- Das Wiederherstellungsmodell muss auf Bulk logged (Massenprotokollierung) oder Full (Vollständig) eingestellt sein.

- Die SQL Server-Sicherung auf mehrere Datenträger wird nicht unterstützt. Wenn das Backup so definiert ist, dass das Datenbank-Backup auf mehrere Dateien über verschiedene Datenträger geschrieben wird, AWS DMS kann die Daten nicht lesen und die AWS DMS Aufgabe schlägt fehl.
- Bei selbstverwalteten SQL-Server-Quellen werden Definitionen von SQL Server Replication Publisher für die in einer DMS-CDC-Aufgabe verwendete Quelle nicht entfernt, wenn die Aufgabe entfernt wird. Bei selbstverwalteten Quellen muss ein SQL Server-Systemadministrator diese Definitionen aus SQL Server löschen.
- Während CDC AWS DMS muss SQL Server-Transaktionsprotokoll-Backups nachschlagen, um Änderungen zu lesen. unterstützt AWS DMS keine SQL Server-Transaktionsprotokoll-Backups, die mit Backup-Software von Drittanbietern erstellt wurden, die nicht im nativen Format vorliegen. Um Backups des Transaktionsprotokolls zu unterstützen, die im nativen Format vorliegen und mit Backup-Software von Drittanbietern erstellt wurden, fügen Sie dem Quellendpunkt das Verbindungsattribut `use3rdPartyBackupDevice=Y` hinzu.
- Beachten Sie bei selbstverwalteten SQL Server-Quellen, dass SQL Server Änderungen an neu erstellten Tabellen erst erfasst, nachdem diese veröffentlicht wurden. Wenn Tabellen zu einer SQL Server-Quelle hinzugefügt werden, AWS DMS verwaltet die Erstellung der Veröffentlichung. Dieser Vorgang kann allerdings einige Minuten dauern. Während dieser Verzögerung vorgenommene Operationen in neu erstellten Tabellen werden nicht erfasst oder in der Zieldatenbank repliziert.
- AWS DMS Die Erfassung von Änderungsdaten erfordert, dass die vollständige Transaktionsprotokollierung in SQL Server aktiviert ist. Um die vollständige Transaktionsprotokollierung in SQL Server zu aktivieren, müssen Sie entweder MS-REPLICATION oder CHANGE DATA CAPTURE (CDC) aktivieren.
- SQL-Server-tlog-Einträge werden erst für die Wiederverwendung markiert, wenn der MS-CDC-Erfassungsauftrag diese Änderungen verarbeitet.
- CDC-Vorgänge werden auf speicheroptimierten Tabellen nicht unterstützt. Diese Einschränkung gilt für SQL Server 2014 (erstmalige Einführung des Features) und höher.
- AWS DMS Für die Erfassung von Datenänderungen ist standardmäßig eine Verteilungsdatenbank auf Amazon EC2- oder On-Premises-SQL-Servern als Quelle erforderlich. Stellen Sie daher sicher, dass Sie den Verteiler aktiviert haben, während Sie MS Replication für Tabellen mit Primärschlüsseln konfigurieren.

Erfassen von Datenänderungen für selbstverwaltete SQL-Server-Quellen (On-Premises oder in Amazon EC2)

Damit Änderungen aus einer Quelldatenbank in Microsoft SQL Server erfasst werden, stellen Sie sicher, dass die Datenbank für vollständige Backups konfiguriert ist. Konfigurieren Sie die Datenbank entweder im vollständigen Wiederherstellungsmodus oder im massenprotokollierten Modus.

Für eine selbstverwaltete SQL Server-Quelle AWS DMS verwendet Folgendes:

MS-Replication

Zum Erfassen von Änderungen für Tabellen mit Primärschlüsseln. Sie können dies automatisch konfigurieren, indem Sie dem AWS DMS Endpunktbenutzer auf der SQL Server-Quelle Instance Sysadmin-Berechtigungen erteilen. Oder Sie können die Schritte in diesem Abschnitt ausführen, um die Quelle vorzubereiten und einen Benutzer zu verwenden, der keine Sysadmin-Berechtigungen für den AWS DMS Endpunkt hat.

MS-CDC

Zum Erfassen von Änderungen für Tabellen ohne Primärschlüssel. Aktivieren Sie MS-CDC auf Datenbankebene und für alle Tabellen einzeln.

Beim Einrichten einer SQL-Server-Datenbank für die laufende Replikation (CDC) können Sie einen der folgenden Schritte ausführen:

- Einrichten der fortlaufenden Replikation mit der Sysadmin-Rolle.
- Einrichten der fortlaufenden Replikation ohne Verwendung der Sysadmin-Rolle.

Einrichten der laufenden Replikation auf einer selbstverwalteten SQL-Server-Quelle

Dieser Abschnitt enthält Informationen zum Einrichten der laufenden Replikation auf einer selbstverwalteten SQL-Server-Quelle mit oder ohne die Sysadmin-Rolle.

Themen

- [Einrichten der laufenden Replikation auf einer selbstverwalteten SQL-Server-Quelle: Verwenden der Sysadmin-Rolle](#)
- [Einrichten der laufenden Replikation auf einer eigenständigen SQL-Server-Quelle: Ohne Sysadmin-Rolle](#)

Einrichten der laufenden Replikation auf einer selbstverwalteten SQL-Server-Quelle: Verwenden der Sysadmin-Rolle

AWS DMS Die laufende Replikation für SQL Server verwendet die native SQL Server-Replikation für Tabellen mit Primärschlüsseln und die Erfassung von Datenänderungen (Change Data Capture, CDC) für Tabellen ohne Primärschlüssel.

Bevor Sie die laufende Replikation einrichten, informieren Sie sich unter [Voraussetzungen für die Verwendung der laufenden Replikation \(CDC\) von einer SQL-Server-Quelle aus](#).

Für Tabellen mit Primärschlüsseln AWS DMS kann im Allgemeinen die erforderlichen Artefakte für die Quelle konfigurieren. Bei selbstverwalteten SQL-Server-Quell-Instances müssen Sie die SQL-Server-Verteilung jedoch zunächst manuell konfigurieren. Danach können AWS DMS Quellbenutzer mit Sysadmin-Berechtigung die Veröffentlichung für Tabellen mit Primärschlüsseln automatisch erstellen.

Um zu überprüfen, ob die Verteilung bereits konfiguriert wurde, führen Sie den folgenden Befehl aus.

```
sp_get_distributor
```

Ist das Ergebnis für die Spaltenverteilung NULL, ist die Verteilung nicht konfiguriert. Sie können die Verteilung anhand des folgenden Verfahrens einrichten.

So richten Sie die Verteilung ein

1. Stellen Sie mithilfe des Tools SQL Server Management Studio (SSMS) eine Verbindung mit der SQL-Server-Quelldatenbank her.
2. Öffnen Sie das Kontextmenü (rechte Maustaste) für den Ordner Replikation und wählen Sie die Option Verteilung konfigurieren. Der Assistent zum Konfigurieren der Verteilung wird angezeigt.
3. Befolgen Sie die Anweisungen im Assistenten, um die Standardwerte einzugeben und die Verteilung zu erstellen.

So richten Sie CDC ein

AWS DMS Version 3.4.7 und höher kann MS CDC für Ihre Datenbank und alle Ihre Tabellen automatisch einrichten, wenn Sie kein schreibgeschütztes Replikat verwenden. Setzen Sie das `ECA SetupMsCdcForTables` auf `true`, um dieses Feature zu verwenden. Weitere Informationen zu ECAs finden Sie unter [Endpunkteinstellungen](#).

Führen Sie für Versionen AWS DMS vor 3.4.7 oder für ein schreibgeschütztes Replikat als Quelle die folgenden Schritte aus:

1. Richten Sie bei Tabellen ohne Primärschlüssel MS-CDC für die Datenbank ein. Verwenden Sie dazu ein Konto, dem die Sysadmin-Rolle zugewiesen ist, und führen Sie den folgenden Befehl aus.

```
use [DBname]
EXEC sys.sp_cdc_enable_db
```

2. Richten Sie anschließend MS-CDC für jede der Quelltabellen ein. Führen Sie die folgende Abfrage für jede Tabelle mit eindeutigen Schlüsseln, aber ohne Primärschlüssel aus, um MS-CDC einzurichten.

```
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. Führen Sie die folgende Abfrage für jede Tabelle ohne Primärschlüssel oder eindeutige Schlüssel aus, um MS-CDC einzurichten.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@role_name = NULL
GO
```

Weitere Informationen zum Einrichten von MS-CDC für bestimmte Tabellen finden Sie in der [SQL Server-Dokumentation](#).

Einrichten der laufenden Replikation auf einer eigenständigen SQL-Server-Quelle: Ohne Sysadmin-Rolle

Informationen zum Einrichten der laufenden Replikation auf einer eigenständigen SQL-Server-Quelle ohne die Sysadmin-Rolle finden Sie unter [Einrichtung der fortlaufenden Replikation auf einem eigenständigen SQL Server: Ohne Sysadmin-Rolle](#).

Einrichten der laufenden Replikation auf einer SQL-Server-DB-Instance in der Cloud

In diesem Abschnitt wird die Einrichtung von CDC für eine in der Cloud gehostete SQL-Server-Datenbank-Instance beschrieben. Eine in der Cloud gehostete SQL-Server-Instance ist eine Instance, die in Amazon RDS für SQL Server, einer von Azure SQL verwalteten Instance oder einer anderen verwalteten Cloud-SQL-Server-Instance ausgeführt wird. Informationen zu den Einschränkungen der laufenden Replikation für jeden Datenbanktyp finden Sie unter [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#).

Bevor Sie die laufende Replikation einrichten, informieren Sie sich unter [Voraussetzungen für die Verwendung der laufenden Replikation \(CDC\) von einer SQL-Server-Quelle aus](#).

Im Gegensatz zu selbstverwalteten Microsoft-SQL-Server-Quellen unterstützt Amazon RDS für SQL Server MS-Replication nicht. Daher AWS DMS muss MS-CDC für Tabellen mit oder ohne Primärschlüssel verwenden.

Amazon RDS gewährt keine Sysadmin-Berechtigungen zum Festlegen von Replikationsartefakten, die für laufende Änderungen in einer Quell-SQL-Server-Instance AWS DMS verwendet. Aktivieren Sie MS-CDC für die Amazon-RDS-Instance (unter Verwendung von Hauptbenutzer-Berechtigungen) wie im folgenden Verfahren gezeigt.

So aktivieren Sie MS-CDC für eine SQL-Server-DB-Instance in der Cloud

1. Führen Sie eine der folgenden Abfragen auf Datenbankebene aus.

Verwenden Sie diese Abfrage für eine DB-Instance von RDS für SQL Server.

```
exec msdb.dbo.rds_cdc_enable_db 'DB_name'
```

Verwenden Sie diese Abfrage für eine von Azure SQL verwaltete DB-Instance.

```
USE DB_name  
GO  
EXEC sys.sp_cdc_enable_db  
GO
```

2. Führen Sie die folgende Abfrage für jede Tabelle mit einem Primärschlüssel aus, um MS-CDC zu aktivieren.

```
exec sys.sp_cdc_enable_table
```

```
@source_schema = N'schema_name',  
@source_name = N'table_name',  
@role_name = NULL,  
@supports_net_changes = 1  
GO
```

Führen Sie die folgende Abfrage für jede Tabelle mit eindeutigen Schlüsseln, aber ohne Primärschlüssel aus, um MS-CDC zu aktivieren.

```
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
```

Führen Sie die folgende Abfrage für jede Tabelle ohne Primärschlüssel oder eindeutige Schlüssel aus, um MS-CDC zu aktivieren.

```
exec sys.sp_cdc_enable_table  
@source_schema = N'schema_name',  
@source_name = N'table_name',  
@role_name = NULL  
GO
```

3. Stellen Sie mit dem folgenden Befehl den Aufbewahrungszeitraum für Änderungen ein, die in der Quelle verfügbar sein sollen.

```
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'
```

Der Parameter `@pollinginterval` wird in Sekunden gemessen, wobei der empfohlene Wert auf 86 399 festgelegt ist. Dies bedeutet, dass das Transaktionsprotokoll Änderungen 86 399 Sekunden (einen Tag) lang speichert, wenn `@pollinginterval = 86399`. Die Prozedur `exec sp_cdc_start_job 'capture'` initiiert die Einstellungen.

Note

Bei einigen Versionen von SQL Server wird der Wert von `pollinginterval` auf den Standardwert von fünf Sekunden zurückgesetzt, wenn er auf mehr als 3 599 Sekunden festgelegt wurde. In diesem Fall werden T-Protokolleinträge gelöscht, bevor sie lesen AWS DMS kann. Informationen dazu, welche SQL-Server-Versionen von diesem bekannten Problem betroffen sind, finden Sie in [diesem Microsoft-KB-Artikel](#).

Wenn Sie Amazon RDS mit Multi-AZ verwenden, stellen Sie sicher, dass Sie auch Ihre sekundäre Availability Zone so einstellen, dass sie im Falle eines Failovers die richtigen Werte hat.

```
exec rdsadmin..rds_set_configuration 'cdc_capture_pollinginterval' , 86399
```

Wenn eine AWS DMS Replikationsaufgabe, die laufende Änderungen an Ihrer SQL Server-Quelle erfasst, länger als eine Stunde anhält, gehen Sie wie folgt vor.

So behalten Sie den Aufbewahrungszeitraum während einer AWS DMS Replikationsaufgabe bei

1. Halten Sie den Auftrag, der die Transaktionsprotokolle kürzt, mit dem folgenden Befehl an.

```
exec sp_cdc_stop_job 'capture'
```

2. Suchen Sie Ihre Aufgabe in der AWS DMS Konsole und setzen Sie die Aufgabe fort.
3. Wählen Sie die Registerkarte Überwachung und überprüfen Sie die Metrik `CDCLatencySource`.
4. Wenn die Metrik `CDCLatencySource` gleich 0 (Null) ist und diesen Wert beibehält, starten Sie den Auftrag, der die Transaktionsprotokolle kürzt, mit dem folgenden Befehl neu.

```
exec sp_cdc_start_job 'capture'
```

Denken Sie daran, den Auftrag zu starten, der die SQL-Server-Transaktionsprotokolle kürzt. Andernfalls könnte der Speicher auf Ihrer SQL-Server-Instance voll werden.

Einschränkungen bei der laufenden Replikation auf einer SQL-Server-DB-Instance in der Cloud

- AWS DMS unterstützt die fortlaufende Replikation (CDC) nur mit dem aktiven Transaktionsprotokoll. Sie können nicht das Backup-Protokoll für CDC verwenden.
- Ereignisse gehen möglicherweise verloren, wenn Sie sie aus dem aktiven Transaktionsprotokoll in das Backup-Protokoll verschieben oder sie im aktiven Transaktionsprotokoll kürzen.

Empfohlene Einstellungen bei Verwendung von Amazon RDS für SQL Server als Quelle für AWS DMS

Wenn Sie mit Amazon RDS für SQL Server als Quelle arbeiten, stützt sich der Erfassungsauftrag auf die Parameter `maxscans` und `maxtrans`. Diese Parameter bestimmen die maximale Anzahl von Scans, die bei der Erfassung im Transaktionsprotokoll durchgeführt werden, und die Anzahl der Transaktionen, die für jeden Scan verarbeitet werden.

Bei Datenbanken, bei denen die Anzahl der Transaktionen größer als `maxtrans*maxscans` ist, kann eine Erhöhung des Werts `polling_interval` zu einer Anhäufung von aktiven Transaktionsprotokoll-Datensätzen führen. Diese Anhäufung kann wiederum zu einer Vergrößerung des Transaktionsprotokolls führen.

Beachten Sie, dass AWS DMS nicht auf dem MS-CDC-Erfassungsauftrag basiert. Der MS-CDC-Erfassungsauftrag kennzeichnet die Transaktionsprotokolleinträge als verarbeitet. Dadurch kann der Backup-Auftrag für das Transaktionsprotokoll die Einträge aus dem Transaktionsprotokoll entfernen.

Es wird empfohlen, die Größe des Transaktionsprotokolls und den Erfolg der MS-CDC-Aufträge zu überwachen. Wenn die MS-CDC-Aufträge fehlschlagen, könnte das Transaktionsprotokoll übermäßig wachsen und AWS DMS zu Replikationsfehlern führen. Sie können Fehler bei MS-CDC-Erfassungsaufträgen mithilfe der dynamischen Verwaltungsansicht `sys.dm_cdc_errors` in der Quelldatenbank überwachen. Sie können die Größe des Transaktionsprotokolls mithilfe des Verwaltungsbefehls `DBCC SQLPERF (LOGSPACE)` überwachen.

So beheben Sie die von MS-CDC verursachte Vergrößerung des Transaktionsprotokolls

1. Überprüfen Sie die `Log Space Used %` für die Datenbank, die aus repliziert AWS DMS wird, und überprüfen Sie, ob sie kontinuierlich zunimmt.

```
DBCC SQLPERF(LOGSPACE)
```

2. Ermitteln Sie, wodurch der Backup-Vorgang des Transaktionsprotokolls blockiert wird.

```
Select log_reuse_wait, log_reuse_wait_desc, name from sys.databases where name = db_name();
```

Wenn der Wert `log_reuse_wait_desc` gleich `REPLICATION` ist, wird die Blockierung des Protokoll-Backups durch die Latenz in MS-CDC verursacht.

3. Erhöhen Sie die Anzahl der vom Erfassungsauftrag verarbeiteten Ereignisse, indem Sie die Parameterwerte `maxtrans` und `maxscans` erhöhen.

```
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'
```

Um dieses Problem zu beheben, legen Sie die Werte von `maxscans` und `maxtrans` so fest, dass der durchschnittlichen Anzahl von Ereignissen `maxtrans*maxscans` entspricht, die für Tabellen generiert werden, die für jeden Tag aus der Quelldatenbank AWS DMS repliziert werden.

Wenn Sie für diese Parameter einen höheren als den empfohlenen Wert festlegen, verarbeiten die Erfassungsaufträge alle Ereignisse in den Transaktionsprotokollen. Wenn Sie für diese Parameter einen niedrigeren als den empfohlenen Wert festlegen, erhöht sich die MS-CDC-Latenz und das Transaktionsprotokoll wird vergrößert.

Die Ermittlung geeigneter Werte für `maxscans` und `maxtrans` kann schwierig sein, da Änderungen des Workloads zu einer unterschiedlichen Anzahl von Ereignissen führen. In diesem Fall empfehlen wir, Überwachung für die MS-CDC-Latenz einzurichten. Weitere Informationen finden Sie unter [Monitor the process](#) in der Dokumentation zu SQL Server. Konfigurieren Sie dann `maxtrans` und `maxscans` dynamisch auf Grundlage der Überwachungsergebnisse.

Wenn die AWS DMS Aufgabe die Protokollsequenznummern (LSNs) nicht finden kann, die zum Fortsetzen oder Fortsetzen der Aufgabe erforderlich sind, schlägt die Aufgabe möglicherweise fehl und erfordert einen vollständigen erneuten Ladevorgang.

Note

Wenn Sie verwenden, AWS DMS um Daten aus einer RDS-für-SQL-Server-Quelle zu replizieren, können Fehler auftreten, wenn Sie versuchen, die Replikation nach einem Stopp-Start-Ereignis der Amazon-RDS-Instance fortzusetzen. Dies liegt daran, dass der SQL-

Server-Agent-Prozess den Erfassungsauftrag neu startet, wenn er nach dem Stopp-Start-Ereignis neu gestartet wird. Dadurch wird das MS-CDC-Abfrageintervall umgangen. Aus diesem Grund kann dies bei Datenbanken mit Transaktionsvolumen, die niedriger als die MS-CDC-Erfassungsauftragsverarbeitung sind, dazu führen, dass Daten verarbeitet oder als repliziert markiert und gesichert werden, bevor von dort aus fortgesetzt werden AWS DMS kann, wo es gestoppt wurde, was zu folgendem Fehler führt:

```
[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)
```

Legen Sie die Werte `maxtrans` und `maxscans` wie zuvor empfohlen fest, um dieses Problem zu beheben.

Unterstützte Komprimierungsmethoden für SQL Server

Beachten Sie in Bezug auf die Unterstützung von SQL-Server-Komprimierungsmethoden in AWS DMS Folgendes:

- AWS DMS unterstützt die Zeilen-/Seitenkomprimierung in SQL Server Version 2008 und höher.
- AWS DMS unterstützt das Vardecimal-Speicherformat nicht.
- AWS DMS unterstützt keine Sparse-Spalten und spaltenweise Strukturkomprimierung.

Arbeiten mit selbstverwalteten SQL Server AlwaysOn-Verfügbarkeitsgruppen

SQL-Server-AlwaysOn-Verfügbarkeitsgruppen bieten Hochverfügbarkeit und Notfallwiederherstellung als Alternative zur Datenbankspiegelung auf Unternehmensebene.

In können AWS DMSSie Änderungen von einem einzelnen Replikat einer primären oder sekundären Verfügbarkeitsgruppe migrieren.

Arbeiten mit dem Replikat der primären Verfügbarkeitsgruppe

Gehen Sie wie folgt vor AWS DMS, um die primäre Verfügbarkeitsgruppe als Quelle in zu verwenden:

1. Aktivieren Sie die Verteilungsoption für alle SQL-Server-Instances in Ihren Replikaten der Verfügbarkeitsgruppe. Weitere Informationen finden Sie unter [Einrichten der laufenden Replikation auf einer selbstverwalteten SQL-Server-Quelle](#).
2. Öffnen Sie in der - AWS DMS Konsole die SQL Server-Quelldatenbankeinstellungen. Geben Sie für Servername den Domain Name Service (DNS)-Namen oder die IP-Adresse ein, der/die für den Verfügbarkeitsgruppen-Listener konfiguriert wurde.

Wenn Sie eine - AWS DMS Aufgabe zum ersten Mal starten, kann es länger als gewöhnlich dauern, bis sie gestartet wird. Das liegt daran, dass die Erstellung der Tabellenartikel durch den Verfügbarkeitsgruppenserver dupliziert wird.

Arbeiten mit dem Replikat einer sekundären Verfügbarkeitsgruppe

Gehen Sie wie folgt vor AWS DMS, um eine sekundäre Verfügbarkeitsgruppe als Quelle in zu verwenden:

1. Verwenden Sie dieselben Anmeldeinformationen für die Verbindung mit einzelnen Replikaten wie die des AWS DMS Quellendpunktbenutzers.
2. Stellen Sie sicher, dass Ihre AWS DMS Replikations-Instance DNS-Namen für alle vorhandenen Replikate auflösen kann, und stellen Sie eine Verbindung zu ihnen her. Sie können die folgende SQL-Abfrage verwenden, um DNS-Namen für alle Ihre Replikate abzurufen.

```
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. Wenn Sie den Quellendpunkt erstellen, geben Sie den DNS-Namen des Verfügbarkeitsgruppen-Listeners als Servername des Endpunkts oder als Serveradresse des Endpunkt-Secrets an. Weitere Informationen zu Verfügbarkeitsgruppen-Listnern finden Sie unter [What is an availability group listener?](#) in der Dokumentation zu SQL Server.

Sie können entweder einen öffentlichen DNS-Server oder einen On-Premises-DNS-Server verwenden, um den Verfügbarkeitsgruppen-Listener, das primäre Replikat und die sekundären Replikate aufzulösen. Wenn Sie einen On-Premises-DNS-Server verwenden möchten, konfigurieren Sie den Amazon Route 53 Resolver. Weitere Informationen finden Sie unter [Verwenden Ihres eigenen Vor-Ort-Nameservers](#).

4. Fügen Sie Ihrem Quellendpunkt die folgenden zusätzlichen Verbindungsattribute hinzu.

Zusätzliches Verbindungsattribut	Wert	Hinweise
<code>applicationIntent</code>	<code>ReadOnly</code>	Ohne diese ODBC-Einstellung wird die Replikationsaufgabe an das Replikat der primären Verfügbarkeitsgruppe weitergeleitet. Weitere Informationen finden Sie unter SQL Server Native Client Support for High Availability, Disaster Recovery in der Dokumentation zu SQL Server.
<code>multiSubnetFailover</code>	<code>yes</code>	Weitere Informationen finden Sie unter SQL Server Native Client Support for High Availability, Disaster Recovery in der Dokumentation zu SQL Server.
<code>alwaysOnSharedBackupIsEnabled</code>	<code>false</code>	Weitere Informationen finden Sie unter Endpunkteinstellungen bei Verwendung von SQL Server als Quelle für AWS DMS .
<code>activateSafeguard</code>	<code>false</code>	Weitere Informationen finden Sie unter Einschränkungen .
<code>setUpMsDcForTables</code>	<code>false</code>	Weitere Informationen finden Sie unter Einschränkungen .

5. Aktivieren Sie die Verteilungsoption für alle Replikate in Ihrer Verfügbarkeitsgruppe. Fügen Sie der Verteilerliste alle Knoten hinzu. Weitere Informationen finden Sie unter [So richten Sie die Verteilung ein](#).
6. Führen Sie die folgende Abfrage für das primäre Lese-/Schreibreplikat aus, um die Veröffentlichung Ihrer Datenbank zu ermöglichen. Sie führen diese Abfrage nur einmal für Ihre Datenbank aus.

```
sp_replicationdboption @dbname = N'<source DB name>', @optname = N'publish', @value  
= N'true';
```

Einschränkungen

Im Folgenden werden Einschränkungen für die Arbeit mit einem Replikat einer sekundären Verfügbarkeitsgruppe aufgeführt:

- AWS DMS unterstützt Safeguard nicht, wenn ein schreibgeschütztes Verfügbarkeitsgruppenreplikat als Quelle verwendet wird. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von SQL Server als Quelle für AWS DMS](#).
- AWS DMS unterstützt das `setUpMsCdcForTables` zusätzliche Verbindungsattribut nicht, wenn ein schreibgeschütztes Verfügbarkeitsgruppenreplikat als Quelle verwendet wird. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von SQL Server als Quelle für AWS DMS](#).
- AWS DMS kann ab Version 3.4.7 ein selbstverwaltetes sekundäres Verfügbarkeitsgruppenreplikat als Quelldatenbank für die fortlaufende Replikation (Erfassung von Datenänderungen oder CDC) verwenden. Multi-AZ-Lesereplikate in Cloud SQL Server werden nicht unterstützt. Wenn Sie frühere Versionen von verwenden AWS DMS, stellen Sie sicher, dass Sie das Replikat der primären Verfügbarkeitsgruppe als Quelldatenbank für CDC verwenden.

Failover auf andere Knoten

Wenn Sie das `ApplicationIntent` zusätzliche Verbindungsattribut für Ihren Endpunkt auf `festlegenReadOnly` festlegen, stellt Ihre AWS DMS Aufgabe eine Verbindung zum schreibgeschützten Knoten mit der höchsten schreibgeschützten Routing-Priorität her. Anschließend erfolgt ein Failover zu anderen schreibgeschützten Knoten in Ihrer Verfügbarkeitsgruppe, wenn der schreibgeschützte Knoten mit der höchsten Priorität nicht verfügbar ist. Wenn Sie nicht `festlegenApplicationIntent` festlegen, stellt Ihre AWS DMS Aufgabe nur eine Verbindung zum primären (Lese-/Schreib-)Knoten in Ihrer Verfügbarkeitsgruppe her.

Sicherheitsanforderungen bei Verwendung von SQL Server als Quelle für AWS Database Migration Service

Das AWS DMS Benutzerkonto muss mindestens über die `db_owner` Benutzerrolle in der SQL Server-Quelldatenbank verfügen, mit der Sie eine Verbindung herstellen.

Endpunkteinstellungen bei Verwendung von SQL Server als Quelle für AWS DMS

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer SQL-Server-Quelldatenbank verwenden. Sie geben die Einstellungen an, wenn Sie den Quellendpunkt mithilfe der AWS DMS Konsole oder mithilfe des `-create-endpoint`-Befehls in der [AWS CLI](#) mit der `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax erstellen.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit SQL Server als Quelle verwenden können.

Name	Beschreibung
ActivateSafeguard	<p>Dieses Attribut aktiviert oder deaktiviert Safeguard . Weitere Informationen zu Safeguard finden Sie im Folgenden unter SafeguardPolicy .</p> <p>Standardwert: true</p> <p>Zulässige Werte: {false, true}</p> <p>Beispiel: '{"ActivateSafeguard": true}'</p>
AlwaysOnSharedSync hedBackupIsEnabled	<p>Dieses Attribut passt das Verhalten von AWS DMS bei der Migration von einer SQL Server-Quelldatenbank an, die als Teil eines AlwaysOn-Verfügbarkeitsgruppen-Clusters gehostet wird.</p> <p>AWS DMS bietet erweiterte Unterstützung für SQL Server-Quelldatenbanken, die für die Ausführung in einem AlwaysOn-Cluster konfiguriert sind. In diesem Fall versucht AWS DMS nachzuerfolgen, ob Transaktions-Backups von anderen Knoten im AlwaysOn-Cluster als dem Knoten, auf dem die Quelldatenbank-Instance gehostet wird, ausgeführt werden. Beim Start der Migrationsaufgabe AWS DMS versucht , eine Verbindung zu jedem Knoten im Cluster herzustellen, schlägt jedoch fehl, wenn keine Verbindung zu einem der Knoten hergestellt werden kann.</p>

Name	Beschreibung
	<p>Wenn Sie alle Knoten im AlwaysOn-Cluster nach Transaktionssicherungen AWS DMS abfragen müssen, setzen Sie dieses Attribut auf <code>false</code>.</p> <p>Standardwert: <code>true</code></p> <p>Gültige Werte: <code>true</code> oder <code>false</code>.</p> <p>Beispiel: <code>'{"AlwaysOnSharedSynchedBackupIsEnabled": false}'</code></p>
<code>"ApplicationIntent": "readonly"</code>	<p>Diese Einstellung des ODBC-Treiberattributs veranlasst SQL Server, Ihre Replikationsaufgabe an den schreibgeschützten Knoten mit der höchsten Priorität weiterzuleiten. Ohne diese Einstellung leitet SQL Server Ihre Replikationsaufgabe an den primären Lese-/Schreibknoten weiter.</p>
<code>EnableNonSysadminWrapper</code>	<p>Verwenden Sie diese Endpunkteinstellung, wenn Sie die laufende Replikation auf einer eigenständigen SQL-Server-Quelle ohne Sysadmin-Benutzer einrichten. Dieser Parameter wird auf AWS DMS Version 3.4.7 und höher unterstützt. Informationen zum Einrichten der laufenden Replikation auf einer eigenständigen SQL-Server-Quelle finden Sie unter Einrichten der laufenden Replikation auf einer eigenständigen SQL-Server-Quelle: Ohne Sysadmin-Rolle.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code></p> <p>Beispiel: <code>'{"EnableNonSysadminWrapper": true}'</code></p>

Name	Beschreibung
ExecuteTimeout	<p>Verwenden Sie dieses zusätzliche Verbindungsattribut (Extra Connection Attribute, ECA), um das Timeout für die Client-Anweisung für die SQL-Server-Instanz in Sekunden festzulegen. Der Standardwert liegt bei 60 Sekunden.</p> <p>Beispiel: <code>'{"ExecuteTimeout": 100}'</code></p>
FatalOnSimpleModel	<p>Ist diese Einstellung auf <code>true</code> gesetzt, wird ein schwerwiegender Fehler generiert, wenn das Wiederherstellungsmodell für SQL-Server-Datenbanken auf <code>simple</code> festgelegt wird.</p> <p>Standardwert: <code>false</code></p> <p>Gültige Werte: <code>true</code> oder <code>false</code>.</p> <p>Beispiel: <code>'{"FatalOnSimpleModel": true}'</code></p>
ForceLobLookup	<p>Erzwingt LOB-Lookup bei Inline-LOB.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code></p> <p>Beispiel: <code>'{"ForceLobLookup": false}'</code></p>
"MultiSubnetFailover": "Yes"	<p>Durch dieses ODBC-Treiberattribut kann DMS im Falle eines Failovers der Verfügbarkeitsgruppen eine Verbindung zur neuen primären Verfügbarkeitsgruppe herstellen. Dieses Attribut ist für Situationen konzipiert, in denen die Verbindung unterbrochen wird oder die Listener-IP-Adresse falsch ist. In diesen Situationen AWS DMS versucht, eine Verbindung zu allen IP-Adressen herzustellen, die dem Availability Group Listener zugeordnet sind.</p>

Name	Beschreibung
ReadBackupOnly	<p>Für die Verwendung dieses Attributs sind Sysadmin-Berechtigungen erforderlich. Wenn dieses Attribut auf gesetzt ist Y, AWS DMS werden während der laufenden Replikation nur Änderungen aus Transaktionsprotokoll-Backups gelesen und nicht aus der aktiven Transaktionsprotokolldatei gelesen. Wenn Sie diesen Parameter auf Y festlegen, können Sie steuern, wie stark die aktive Transaktionsprotokolldatei beim vollständigen Laden und fortlaufenden Replikationsaufgaben wächst. Dies kann bei fortlaufender Replikation jedoch zu einer gewissen Quelllatenz führen.</p> <p>Gültige Werte: N oder Y. Der Standardwert ist N.</p> <p>Beispiel: <code>'{"ReadBackupOnly": Y}'</code></p> <p>Hinweis: Aufgrund der Art und Weise, wie RDS Backups durchführt, funktioniert dieser Parameter nicht auf Quell-Instances von Amazon RDS SQL Server.</p>

Name	Beschreibung
SafeguardPolicy	<p>Für eine optimale Leistung AWS DMS versucht , alle ungelesenen Änderungen aus dem aktiven Transaktionsprotokoll (TLOG) zu erfassen. Manchmal enthält das aktive TLOG jedoch aufgrund von Kürzungen nicht alle ungelesenen Änderungen. In diesem Fall greift AWS DMS auf das Protokoll-Backup zu, um die fehlenden Änderungen zu erfassen. Um die Notwendigkeit für den Zugriff auf das Protokoll-Backup zu minimieren, AWS DMS vermeidet das Kürzen mit einer der folgenden Methoden:</p> <ol style="list-style-type: none">1. RELY_ON_SQL_SERVER_REPLICATION_AGENT (Transaktionen in der Datenbank starten): Dies ist die Standardeinstellung für AWS DMS. <p>Wenn Sie diese Einstellung verwenden, erfordert AWS DMS die Ausführung des SQL Server Log Reader Agent, damit AWS DMS für die Replikation markierte Transaktionen aus dem aktiven TLOG verschieben kann. Beachten Sie, dass das aktive TLOG voll werden kann, wenn der Log Reader Agent nicht ausgeführt wird. In diesem Fall wechselt die Quelldatenbank in den schreibgeschützten Modus, bis das Problem behoben wurde. Wenn Sie Microsoft Replication in Ihrer Datenbank für einen anderen Zweck als aktivieren müssen AWS DMS, müssen Sie diese Einstellung auswählen.</p> <p>Wenn Sie diese Einstellung verwenden, AWS DMS minimiert Protokoll-Backup-Lesevorgänge, indem eine Tabelle mit dem Namen erstellt wird, <code>awsdms_truncation_safeguard</code> und verhindert die TLOG-Kürzung, indem eine offene Transaktion in der Datenbank nachgeahmt wird. Dadurch wird verhindert, dass die Datenbank Ereignisse kürzt und sie fünf</p>

Name	Beschreibung
	<p>Minuten lang in das Backup-Protokoll verschiebt (standardmäßig). Stellen Sie sicher, dass die Tabelle in keinem Wartungsplan enthalten ist, da der Wartungsauftrag andernfalls möglicherweise fehlschlägt. Sie können die Tabelle problemlos löschen, wenn keine Aufgaben mit der Datenbankoption <code>Start Transactions</code> konfiguriert sind.</p> <p>2. <code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code> (Ausschließlich <code>sp_repldone</code> mit einer einzelnen Aufgabe verwenden): Wenn Sie diese Einstellung verwenden, AWS DMS hat die volle Kontrolle über den Replikationsagentenprozess, der Protokolleinträge als <code>ready for truncation</code> verwendend markiert <code>sp_repldone</code>. Bei dieser Einstellung verwendet keine Dummy AWS DMS -Transaktion wie bei der <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code> (Standard)-Einstellung. Sie können diese Einstellung nur verwenden, wenn MS Replication nur für AWS DMS die Quelldatenbank verwendet wird. Wenn Sie diese Einstellung verwenden, kann nur eine AWS DMS Aufgabe auf die Datenbank zugreifen. Wenn Sie parallele AWS DMS Aufgaben für dieselbe Datenbank ausführen müssen, verwenden Sie <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code>.</p> <ul style="list-style-type: none"> • Diese Einstellung erfordert, dass der Log Reader Agent in der Datenbank beendet wird. Wenn der Log Reader Agent ausgeführt wird, wenn die Aufgabe gestartet wird, erzwingt die AWS DMS Aufgabe das Beenden. Sie können den Log Reader Agent auch manuell beenden, bevor Sie die Aufgabe starten. • Wenn Sie diese Methode mit MS-CDC verwenden, sollten Sie die Aufträge zur MS-CDC-Erfassung und MS-CDC-Bereinigung beenden und deaktivieren.

Name	Beschreibung
	<ul style="list-style-type: none">• Sie können diese Einstellung nicht verwenden, wenn der Microsoft SQL Server-Migrationsauftrag auf einem Remote-Distributor-Computer ausgeführt wird, da keinen Zugriff auf den Remote-Computer hat.• <code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code> funktioniert auf Quell-Instances in Amazon RDS für SQL Server nicht, da Amazon-RDS-Benutzer keinen Zugriff zum Ausführen des gespeicherten Verfahrens <code>sp_repldone</code> haben.• Wenn Sie <code>SafeguardPolicy</code> auf <code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code> setzen, ohne die <code>Sysadmin</code>-Rolle zu verwenden, müssen Sie dem Benutzer <code>dmsuser</code> Berechtigungen für die Objekte <code>dbo.syscategories</code> und <code>dbo.sysjobs</code> erteilen. <p>Standardwert: <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code></p> <p>Zulässige Werte: <code>{EXCLUSIVE_AUTOMATIC_TRUNCATION , RELY_ON_SQL_SERVER_REPLICATION_AGENT }</code></p> <p>Beispiel: <code>'{"SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION"}'</code></p>

Name	Beschreibung
SetupMsCdcForTables	<p>Dieses Attribut aktiviert MS-CDC für die Quelldatenbank und für Tabellen in der Aufgabenzuordnung, für die MS-Replication nicht aktiviert ist. Wenn Sie diesen Wert auf <code>true</code> setzen, wird die gespeicherte Prozedur <code>sp_cdc_enable_db</code> in der Quelldatenbank und die gespeicherte Prozedur <code>sp_cdc_enable_table</code> für jede Tabelle in der Aufgabe ausgeführt, für die MS-Replication in der Quelldatenbank nicht aktiviert ist. Weitere Informationen zum Aktivieren der Verteilung finden Sie unter Einrichten der laufenden Replikation auf einer selbstverwalteten SQL-Server-Quelle.</p> <p>Zulässige Werte: {<code>true</code>, <code>false</code>}</p> <p>Beispiel: <code>'{"SetupMsCdcForTables": true}'</code></p>
TlogAccessMode	<p>Gibt den Modus an, der zum Abrufen von CDC-Daten verwendet wird.</p> <p>Standardwert: <code>PreferTlog</code></p> <p>Zulässige Werte: <code>BackupOnly</code> , <code>PreferBackup</code> , <code>PreferTlog</code> , <code>TlogOnly</code></p> <p>Beispiel: <code>'{"TlogAccessMode": "PreferTlog"}</code></p>
Use3rdPartyBackupDevice	<p>Wenn dieses Attribut auf <code>Y</code> gesetzt ist, verarbeitet AWS DMS Transaktionsprotokoll-Backups von Drittanbietern, falls sie im nativen Format erstellt wurden.</p>

Quelldatentypen für SQL Server

Datenmigration, die SQL Server als Quelle für verwendet, AWS DMS unterstützt die meisten SQL Server-Datentypen. Die folgende Tabelle zeigt die SQL Server-Quelldatentypen, die bei

der Verwendung von unterstützt werden, AWS DMS und die Standardzuordnung von AWS DMS Datentypen.

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

SQL Server-Datentypen	AWS DMS -Datentypen
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 und höher)	DATETIME
SMALLDATETIME	DATETIME
DATUM	DATUM
TIME	TIME

SQL Server-Datentypen	AWS DMS -Datentypen
DATETIMEOFFSET	WSTRING
CHAR	STRING
VARCHAR	STRING
VARCHAR (max)	<p>CLOB</p> <p>TEXT</p> <p>Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von CLOB-Datentypen für eine bestimmte Aufgabe aktivieren.</p> <p>AWS DMS Aktualisiert für SQL-Server-Tabelle n LOB-Spalten im Ziel auch für UPDATE-Anweisungen, die den Wert der LOB-Spalte in SQL Server nicht ändern.</p> <p>Während CDC AWS DMS unterstützt CLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>
NCHAR	WSTRING
NVARCHAR (Länge)	WSTRING

SQL Server-Datentypen	AWS DMS -Datentypen
NVARCHAR (max)	<p>NCLOB</p> <p>NTEXT</p> <p>Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von SupportLobs für eine bestimmte Aufgabe aktivieren. Weitere Informationen zum Aktivieren von Lob-Unterstützung finden Sie unter Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS.</p> <p>AWS DMS Aktualisiert für SQL-Server-Tabelle n LOB-Spalten im Ziel auch für UPDATE-Anweisungen, die den Wert der LOB-Spalte in SQL Server nicht ändern.</p> <p>Während CDC AWS DMS unterstützt CLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>
BINARY	BYTES
VARBINARY	BYTES

SQL Server-Datentypen	AWS DMS -Datentypen
VARBINARY (max)	<p data-bbox="833 226 922 258">BLOB</p> <p data-bbox="833 306 938 338">IMAGE</p> <p data-bbox="833 386 1495 562">AWS DMS Aktualisiert für SQL-Server-Tabelle n LOB-Spalten im Ziel auch für UPDATE-Anweisungen, die den Wert der LOB-Spalte in SQL Server nicht ändern.</p> <p data-bbox="833 611 1495 787">Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von BLOB-Datentypen für eine bestimmte Aufgabe aktivieren.</p> <p data-bbox="833 835 1495 911">AWS DMS unterstützt BLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>
TIMESTAMP (ZEITSTEMPEL)	BYTES
UNIQUEIDENTIFIER	STRING
HIERARCHYID	<p data-bbox="833 1121 1484 1251">Verwenden Sie HIERARCHYID bei der Replikation von Daten auf einem SQL Server-Zielendpunkt.</p> <p data-bbox="833 1299 1484 1375">Verwenden Sie WSTRING (250) bei der Replikation auf allen anderen Zielendpunkten.</p>

SQL Server-Datentypen	AWS DMS -Datentypen
XML	<p data-bbox="829 226 943 258">NCLOB</p> <p data-bbox="829 306 1490 485">AWS DMS Aktualisiert für SQL-Server-Tabellen LOB-Spalten im Ziel auch für UPDATE-Anweisungen, die den Wert der LOB-Spalte in SQL Server nicht ändern.</p> <p data-bbox="829 531 1450 709">Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von NCLOB-Datentypen für eine bestimmte Aufgabe aktivieren.</p> <p data-bbox="829 753 1490 879">Während CDC AWS DMS unterstützt NCLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>
GEOMETRY	<p data-bbox="829 930 1479 1056">Verwenden Sie GEOMETRY bei der Replikation auf Zielendpunkte, die diesen Datentyp unterstützen.</p> <p data-bbox="829 1102 1466 1228">Verwenden Sie CLOB bei der Replikation auf Zielendpunkte, die diesen Datentyp nicht unterstützen.</p>
GEOGRAPHY	<p data-bbox="829 1283 1503 1409">Verwenden Sie GEOGRAPHY bei der Replikation auf Zielendpunkte, die diesen Datentyp unterstützen.</p> <p data-bbox="829 1455 1466 1581">Verwenden Sie CLOB bei der Replikation auf Zielendpunkte, die diesen Datentyp nicht unterstützen.</p>

AWS DMS unterstützt keine Tabellen, die Felder mit den folgenden Datentypen enthalten.

- CURSOR
- SQL_VARIANT

- TABLE

Note

Benutzerdefinierte Datentypen werden abhängig von dem Typ, auf dem sie basieren, unterstützt. Beispielsweise wird ein benutzerdefinierter Datentyp basierend auf DATETIME als Datentyp DATETIME verarbeitet.

Verwenden einer Microsoft Azure SQL-Datenbank als Quelle für AWS DMS

Mit AWS DMS können Sie eine Microsoft-Azure-SQL-Datenbank ähnlich als Quelle verwenden wie SQL Server. AWS DMS unterstützt dieselben aufgelisteten Datenbankversionen als Quelle wie für SQL Server On-Premises oder auf einer Amazon-EC2-Instance.

Weitere Informationen finden Sie unter [Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS](#).

Note

AWS DMS unterstützt keine Change Data Capture-Operationen (CDC) mit einer Azure SQL-Datenbank.

Verwenden einer Microsoft-Azure-SQL-Managed-Instance als Quelle für AWS DMS

Mit AWS DMS können Sie eine Microsoft-Azure-SQL-Managed-Instance ähnlich als Quelle verwenden wie SQL Server. AWS DMS unterstützt dieselben aufgelisteten Datenbankversionen als Quelle wie für SQL Server bei Ausführung On-Premises oder auf einer Amazon-EC2-Instance.

Weitere Informationen finden Sie unter [Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS](#).

Verwendung des flexiblen Microsoft-Azure-Database-für-PostgreSQL-Servers als Quelle für AWS DMS

Mit AWS DMS können Sie eine Microsoft-Azure-SQL-Datenbank ähnlich als Quelle verwenden wie PostgreSQL.

Informationen zu Versionen des flexiblen Microsoft-Azure-Database-für-PostgreSQL-Servers, der AWS DMS als Quelle unterstützt, finden Sie unter [Quellen für AWS DMS](#).

Einrichtung eines flexiblen Microsoft-Azure-für-PostgreSQL-Servers für logische Replikation und Dekodierung

Sie können während der Datenbankmigration logische Replikations- und Dekodierungs-Features auf dem flexiblen Microsoft-Azure-Database-for-PostgreSQL-Server verwenden.

Für die logische Dekodierung verwendet DMS das Plugin `test_decoding` oder `pglogical`. Wenn das `pglogical`-Plugin in einer PostgreSQL-Quelldatenbank verfügbar ist, erstellt DMS einen Replikations-Slot mit `pglogical`, andernfalls wird das `test_decoding`-Plugin verwendet.

Gehen Sie wie folgt vor, um Ihren flexiblen Microsoft-Azure-für-PostgreSQL-Server als Quellendpunkt für DMS zu konfigurieren:

1. Öffnen Sie die Seite „Serverparameter“ im Portal.
2. Setzen Sie den `wal_level`-Serverparameter auf LOGICAL.
3. Wenn Sie die `pglogical`-Erweiterung verwenden möchten, setzen Sie die Parameter `shared_preload_libraries` und `azure.extensions` auf `pglogical`.
4. Setzen Sie den `max_replication_slots`-Parameter auf die maximale Anzahl von DMS-Aufgaben ein, die Sie gleichzeitig ausführen möchten. In Microsoft Azure ist der Standardwert für diesen Parameter 10. Der Maximalwert dieses Parameters hängt vom verfügbaren Speicher Ihrer PostgreSQL-Instance ab und ermöglicht zwischen 2 und 8 Replikations-Slots pro GB Speicher.
5. Legen Sie für den Parameter `max_wal_senders` einen Wert größer als 1 fest. Der Parameter `max_wal_senders` legt die Anzahl der Aufgaben fest, die gleichzeitig ausgeführt werden können. Der Standardwert lautet 10.
6. Setzen Sie den Parameterwert `max_worker_processes` auf mindestens 16. Andernfalls werden möglicherweise Fehler wie die folgenden angezeigt:

```
WARNING: out of background worker slots.
```

7. Speichern Sie die Änderungen. Starten Sie den Server neu, um die Änderungen zu übernehmen.
8. Vergewissern Sie sich, dass Ihre PostgreSQL-Instance Netzwerkverkehr von Ihrer Verbindungsressource zulässt.
9. Erteilen Sie einem vorhandenen Benutzer Replikationsberechtigungen oder erstellen Sie einen neuen Benutzer mit Replikationsberechtigungen, indem Sie die folgenden Befehle verwenden.
 - Mit folgendem Befehl gewähren Sie einem vorhandenen Benutzer Replikationsberechtigungen:

```
ALTER USER <existing_user> WITH REPLICATION;
```

- Mit folgendem Befehl erstellen Sie einen neuen Benutzer mit Replikationsberechtigungen:

```
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;
```

Weitere Informationen über die logische Replikation von PostgreSQL finden Sie in den folgenden Themen:

- [Aktivieren der Erfassung von Datenänderungen \(CDC\) mithilfe logischer Replikation](#)
- [Verwenden von nativen CDC-Startpunkten zum Einrichten einer CDC-Last einer PostgreSQL-Quelle](#)
- [Logische Replikation und logische Dekodierung in Azure Database für PostgreSQL – Flexibler Server](#) in der [Azure-Database-für PostgreSQL-Dokumentation](#).

Verwenden eines flexiblen Microsoft-Azure-Datenbank-für-MySQL-Servers als Quelle für AWS DMS

Mit AWS DMS können Sie einen flexiblen Microsoft-Azure-Datenbank-für-MySQL-Server ähnlich als Quelle verwenden wie MySQL.

Informationen zu den Versionen von flexiblen Microsoft-Azure-Datenbank-für-MySQL-Servern, die AWS DMS als Quelle unterstützen, finden Sie unter [Quellen für AWS DMS](#).

Weitere Hinweise zur Verwendung einer vom Kunden verwalteten MySQL-kompatiblen Datenbank mit AWS DMS finden Sie unter [Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#).

Einschränkungen bei der Verwendung von Azure MySQL als Quelle für AWS Database Migration Service

- Der Standardwert für die flexible Azure-MySQL-Serversystemvariable `sql_generate_invisible_primary_key` ist 0N und der Server fügt automatisch einen generierten unsichtbaren Primärschlüssel (GIPK) zu jeder Tabelle hinzu, die ohne einen expliziten Primärschlüssel erstellt wurde. AWS DMS unterstützt keine fortlaufende Replikation für MySQL-Tabellen mit GIPK-Einschränkungen.

Verwendung von OCI MySQL Heatwave als Quelle für AWS DMS

Mit AWS DMS können Sie OCI MySQL Heatwave auf die gleiche Weise wie MySQL als Quelle verwenden. Die Verwendung von OCI MySQL Heatwave als Quelle erfordert einige zusätzliche Konfigurationsänderungen.

Informationen zu den Versionen von OCI MySQL Heatwave, die AWS DMS als Quelle unterstützen, finden Sie unter [Quellen für AWS DMS](#).

Einrichtung von OCI MySQL Heatwave für die logische Replikation

Gehen Sie wie folgt vor, um Ihre OCI-MySQL-Heatwave-Instance als Quellendpunkt für DMS zu konfigurieren:

1. Melden Sie sich bei der OCI-Konsole an und öffnen Sie das „Hamburger“-Hauptmenü (≡) in der oberen linken Ecke.
2. Wählen Sie Datenbanken, DB-Systeme.
3. Öffnen Sie das Menü Konfigurationen.
4. Wählen Sie Create configuration (Konfiguration erstellen).
5. Geben Sie einen Konfigurationsnamen ein, z. B. **dms_configuration**.
6. Wählen Sie die Form Ihrer aktuellen OCI-MySQL-Heatwave-Instance. Sie finden die Form auf der Registerkarte „Eigenschaften“ der DB-Systemkonfiguration der Instance im Abschnitt DB-Systemkonfiguration:Form.

7. Wählen Sie im Abschnitt Benutzervariablen die Systemvariable `binlog_row_value_options` aus. Der Standardwert ist `PARTIAL_JSON`. Löschen Sie den Wert.
8. Wählen Sie die Schaltfläche Erstellen aus.
9. Öffnen Sie Ihre OCI-MySQLHeatwave-Instance und wählen Sie die Schaltfläche Bearbeiten aus.
10. Wählen Sie im Abschnitt Konfiguration die Schaltfläche Konfiguration ändern und wählen Sie die Form-Konfiguration aus, die Sie in Schritt 4 erstellt haben.
11. Sobald die Änderungen wirksam werden, ist Ihre Instance für die logische Replikation bereit.

Verwenden von Google Cloud for MySQL als Quelle für AWS DMS

Mit AWS DMS können Sie Google Cloud for MySQL auf die gleiche Weise wie MySQL als Quelle verwenden.

Informationen zu den Versionen von GCP MySQL, die AWS DMS als Quelle unterstützen, finden Sie unter [Quellen für AWS DMS](#).

Weitere Informationen finden Sie unter [Verwenden einer MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#).

Note

Unterstützung für GCP MySQL 8.0 als Quelle ist in AWS DMS Version 3.4.6 verfügbar. AWS DMS unterstützt den SSL-Modus `verify-full` für GCP für MySQL-Instances nicht. Die GCP-MySQL-Sicherheitseinstellung `Allow only SSL connections` wird nicht unterstützt, da sie sowohl die Überprüfung des Server- als auch des Client-Zertifikats erfordert. AWS DMS unterstützt nur die Überprüfung von Serverzertifikaten. AWS DMS unterstützt den Standardwert von GCP CloudSQL for MySQL `CRC32` für das `binlog_checksum`-Datenbank-Flag.

Verwenden von Google Cloud für PostgreSQL als Quelle für AWS DMS

Mit AWS DMS können Sie Google Cloud für PostgreSQL auf die gleiche Weise als Quelle verwenden wie selbstverwaltete PostgreSQL-Datenbanken.

Informationen zu den Versionen von GCP PostgreSQL, die AWS DMS als Quelle unterstützen, finden Sie unter [Quellen für AWS DMS](#).

Weitere Informationen finden Sie unter [Verwenden einer PostgreSQL-Datenbank als AWS DMS-Quelle](#).

Richten Sie Google Cloud für PostgreSQL für die logische Replikation und Dekodierung ein

Sie können während der Datenbankmigration Features für logische Replikation und Dekodierung in Google Cloud SQL for PostgreSQL verwenden.

Für die logische Dekodierung verwendet DMS eines der folgenden Plugins:

- `test_decoding`
- `pglogical`

Wenn das `pglogical`-Plugin in einer PostgreSQL-Quelldatenbank verfügbar ist, erstellt DMS einen Replikations-Slot mit `pglogical`, andernfalls wird das `test_decoding`-Plugin verwendet.

Beachten Sie Folgendes zur Verwendung der logischen Dekodierung mit AWS DMS:

1. Aktivieren Sie mit Google Cloud SQL für PostgreSQL die logische Dekodierung, indem Sie das Flag `cloudsql.logical_decoding` auf `on` setzen.
2. Um `pglogical` zu aktivieren, setzen Sie das `cloudsql.enable_pglogical`-Flag auf `on` und starten Sie die Datenbank neu.
3. Um logische Dekodierungs-Features zu verwenden, erstellen Sie einen PostgreSQL-Benutzer mit dem Attribut `REPLICATION`. Wenn Sie die `pglogical`-Erweiterung verwenden, muss der Benutzer die Rolle `cloudsqlsuperuser` haben. Gehen Sie wie folgt vor, um einen Benutzer mit der `cloudsqlsuperuser`-Rolle zu erstellen:

```
CREATE USER new_aws_dms_user WITH REPLICATION
IN ROLE cloudsqlsuperuser LOGIN PASSWORD 'new_aws_dms_user_password';
```

Um dieses Attribut für einen vorhandenen Benutzer festzulegen, führen Sie die folgenden Schritte aus:

```
ALTER USER existing_user WITH REPLICATION;
```

4. Setzen Sie den `max_replication_slots`-Parameter auf die maximale Anzahl von DMS-Aufgaben ein, die Sie gleichzeitig ausführen möchten. In Google Cloud SQL ist der Standardwert

für diesen Parameter 10. Der Maximalwert dieses Parameters hängt vom verfügbaren Speicher Ihrer PostgreSQL-Instance ab und ermöglicht zwischen 2 und 8 Replikations-Slots pro GB Speicher.

Weitere Informationen über die logische Replikation von PostgreSQL finden Sie in den folgenden Themen:

- [Aktivieren der Erfassung von Datenänderungen \(CDC\) mithilfe logischer Replikation](#)
- [Verwenden von nativen CDC-Startpunkten zum Einrichten einer CDC-Last einer PostgreSQL-Quelle](#)
- [Einrichten der logischen Replikation und Dekodierung](#) in der [Cloud-SQL-for-PostgreSQL-Dokumentation](#).

Verwenden einer PostgreSQL-Datenbank als AWS DMS-Quelle

Sie können Daten von einer oder vielen PostgreSQL-Datenbanken mithilfe von AWS DMS migrieren. Mit einer PostgreSQL-Datenbank als Quelle können Sie Daten zu einer anderen PostgreSQL-Datenbank oder einer der anderen unterstützten Datenbanken migrieren.

Informationen zu den Versionen von PostgreSQL, die AWS DMS als Quelle unterstützt, finden Sie unter [Quellen für AWS DMS](#).

AWS DMS unterstützt PostgreSQL für diese Datenbanktypen:

- Lokale Datenbanken
- Datenbanken auf einer Amazon-EC2-Instance
- Datenbanken auf einer DB-Instance von Amazon RDS
- Datenbanken auf einer DB-Instance basierend auf der Amazon-Aurora-PostgreSQL-kompatiblen Edition
- Datenbanken auf einer DB-Instance basierend auf der Amazon-Aurora-PostgreSQL-kompatiblen Serverless-Edition

Note

DMS unterstützt Amazon Aurora PostgreSQL – Serverless V1 nur als Quelle für Volllastaufgaben. Sie können jedoch Amazon Aurora PostgreSQL – Serverless V2 als Quelle für Volllast-, Volllast-und-CDC- und reine CDC-Aufgaben verwenden.

Zu verwendende AWS DMS-Version

Verwenden Sie eine beliebige verfügbare AWS DMS-Version.

Verwenden Sie AWS DMS Version 3.4.3 und höher.

Verwenden Sie AWS DMS Version 3.4.7 und höher.

Verwenden Sie AWS DMS Version 3.5.1 und höher.

Sie können Secure Sockets Layer (SSL) verwenden, um Verbindungen zwischen Ihrem PostgreSQL-Endpunkt und der Replikations-Instance zu verschlüsseln. Weitere Informationen zur Verwendung von SSL mit einem PostgreSQL-Endpunkt finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).

Eine zusätzliche Sicherheitsanforderung bei Verwendung von PostgreSQL als Quelle besteht darin, dass das angegebene Benutzerkonto ein registrierter Benutzer in der PostgreSQL-Datenbank sein muss.

Führen Sie die folgenden Schritte aus, um eine PostgreSQL-Datenbank als AWS DMS-Quellendpunkt zu konfigurieren:

- Erstellen Sie einen PostgreSQL-Benutzer mit entsprechenden Berechtigungen, um AWS DMS Zugriff auf Ihre PostgreSQL-Quelldatenbank zu gewähren.

Note

- Wenn es sich bei Ihrer PostgreSQL-Quelldatenbank um eine selbstverwaltete Datenbank handelt, informieren Sie sich unter [Arbeiten mit selbstverwalteten PostgreSQL-Datenbanken als Quelle in AWS DMS](#).
- Wenn es sich bei Ihrer PostgreSQL-Quelldatenbank um eine von Amazon RDS verwaltete Datenbank handelt, informieren Sie sich unter [Arbeiten mit von AWS verwalteten PostgreSQL-Datenbanken als DMS-Quelle](#).

- Erstellen Sie einen PostgreSQL-Quellendpunkt, der der von Ihnen ausgewählten PostgreSQL-Datenbankkonfiguration entspricht.
- Erstellen Sie eine Aufgabe oder eine Reihe von Aufgaben, um Ihre Tabellen zu migrieren.

Um eine full-load-only Aufgabe zu erstellen, ist keine weitere Endpunktconfiguration erforderlich.

Bevor Sie eine Aufgabe für die Erfassung von Datenänderungen erstellen (eine reine CDC-Aufgabe oder eine Volllast-und-CDC-Aufgabe), informieren Sie sich unter [Aktivieren von CDC mit einer selbstverwalteten PostgreSQL-Datenbank als AWS DMS-Quelle](#) oder [Aktivieren von CDC mit einer von AWS verwalteten PostgreSQL-DB-Instance mit AWS DMS](#).

Themen

- [Arbeiten mit selbstverwalteten PostgreSQL-Datenbanken als Quelle in AWS DMS](#)
- [Arbeiten mit von AWS verwalteten PostgreSQL-Datenbanken als DMS-Quelle](#)
- [Aktivieren der Erfassung von Datenänderungen \(CDC\) mithilfe logischer Replikation](#)
- [Verwenden von nativen CDC-Startpunkten zum Einrichten einer CDC-Last einer PostgreSQL-Quelle](#)
- [Migrieren von PostgreSQL auf PostgreSQL mit AWS DMS](#)
- [Entfernen von AWS DMS-Artefakten aus einer PostgreSQL-Quelldatenbank](#)
- [Zusätzliche Konfigurationseinstellungen bei Verwendung einer PostgreSQL-Datenbank als DMS-Quelle](#)
- [Verwenden der MapBooleanAsBoolean PostgreSQL-Endpunkteinstellung](#)
- [Endpunkteinstellungen und zusätzliche Verbindungsattribute \(ECAs\) bei Verwendung von PostgreSQL als DMS-Quelle](#)
- [Einschränkungen bei Verwendung einer PostgreSQL-Datenbank als DMS-Quelle](#)

- [Quelldatentypen für PostgreSQL](#)

Arbeiten mit selbstverwalteten PostgreSQL-Datenbanken als Quelle in AWS DMS

Mit einer selbstverwalteten PostgreSQL-Datenbank als Quelle können Sie Daten zu einer anderen PostgreSQL-Datenbank oder einer der anderen von AWS DMS unterstützten Datenbanken migrieren. Bei der Datenbankquelle kann es sich um eine On-Premises-Datenbank oder eine selbstverwaltete Engine handeln, die auf einer Amazon-EC2-Instance ausgeführt wird. Sie können eine DB-Instance sowohl für Volllast- als auch für CDC-Aufgaben verwenden.

Voraussetzungen für die Verwendung einer selbstverwalteten PostgreSQL-Datenbank als AWS DMS-Quelle

Führen Sie vor der Migration von Daten aus einer selbstverwalteten PostgreSQL-Quelldatenbank die folgenden Schritte aus:

- Stellen Sie sicher, dass Sie eine PostgreSQL-Datenbank der Version 9.4.x oder höher verwenden.
- Erteilen Sie für Volllast- und CDC- oder reine CDC-Aufgaben Superuser-Berechtigungen für das Benutzerkonto, das für die PostgreSQL-Quelldatenbank angegeben ist. Das Benutzerkonto erfordert Superuser-Berechtigungen, um auf replikationsspezifische Funktionen in der Quelle zugreifen zu können. Für reine Volllastaufgaben erfordert das Benutzerkonto SELECT-Berechtigungen für Tabellen, um diese migrieren zu können.
- Fügen Sie der `pg_hba.conf`-Konfigurationsdatei die IP-Adresse des AWS DMS-Replikationsservers hinzu, und aktivieren Sie Replikations- und Socketverbindungen. Ein Beispiel folgt.

```
# 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
```

Die Konfigurationsdatei `pg_hba.conf` von PostgreSQL steuert die Clientauthentifizierung. (HBA steht für hostbasierte Authentifizierung) Die Datei wird üblicherweise im Datenverzeichnis des Datenbankclusters gespeichert.

- Wenn Sie eine Datenbank als Quelle für die logische Replikation mit AWS DMS konfigurieren, informieren Sie sich unter [Aktivieren von CDC mit einer selbstverwalteten PostgreSQL-Datenbank als AWS DMS-Quelle](#).

Note

Einige AWS DMS-Transaktionen sind für einige Zeit im Leerlauf, bevor die DMS-Engine sie erneut nutzt. Über den Parameter `idle_in_transaction_session_timeout` in PostgreSQL-Versionen 9.6 und höher können Sie für Transaktionen im Leerlauf ein Timeout und Fehlschlagen veranlassen. Beenden Sie Transaktionen im Leerlauf nicht, wenn Sie AWS DMS verwenden.

Aktivieren von CDC mit einer selbstverwalteten PostgreSQL-Datenbank als AWS DMS-Quelle

AWS DMS unterstützt die Erfassung von Datenänderungen (CDC) mithilfe logischer Replikation. Legen Sie die folgenden Parameter und Werte in der Konfigurationsdatei `postgresql.conf` fest, um die logische Replikation einer selbstverwalteten PostgreSQL-Quelldatenbank zu aktivieren:

- Set `wal_level = logical`.
- Legen Sie für `max_replication_slots` einen Wert größer als 1 fest.

Legen Sie den Wert `max_replication_slots` gemäß der Anzahl der Aufgaben fest, die Sie ausführen möchten. Wenn Sie beispielsweise fünf Aufgaben ausführen möchten, legen Sie mindestens fünf Slots fest. Slots werden automatisch geöffnet, sobald eine Aufgabe gestartet wird und bleiben geöffnet, selbst wenn die Aufgabe nicht mehr ausgeführt wird. Löschen Sie offene Slots unbedingt manuell. Beachten Sie, dass DMS beim Löschen der Aufgabe automatisch Replikations-Slots löscht, sofern DMS den Slot erstellt hat.

- Legen Sie für `max_wal_senders` einen Wert größer als 1 fest.

Der Parameter `max_wal_senders` legt die Anzahl der Aufgaben fest, die gleichzeitig ausgeführt werden können.

- Der Parameter `wal_sender_timeout` beendet Replikationsverbindungen, die länger als die angegebene Anzahl von Millisekunden inaktiv sind. Der Standardwert für eine On-Premises-PostgreSQL-Datenbank beträgt 60 000 Millisekunden (60 Sekunden). Wenn Sie den Wert auf 0 (Null) festlegen, wird der Timeout-Mechanismus deaktiviert. Dies ist eine gültige Einstellung für DMS.

Wenn Sie den Wert `wal_sender_timeout` auf einen anderen Wert als Null festlegen, benötigt eine DMS-Aufgabe mit CDC mindestens 10 000 Millisekunden (10 Sekunden) und schlägt fehl, wenn der Wert unter 10 000 liegt. Halten Sie den Wert unter 5 Minuten, um eine Verzögerung während eines Multi-AZ-Failovers einer DMS-Replikations-Instance zu vermeiden.

Einige Parameter sind statisch und können nur beim Serverstart festgelegt werden. Änderungen an ihren Einträgen in der Konfigurationsdatei (für eine selbstverwaltete Datenbank) oder der DB-Parametergruppe (für eine Datenbank von RDS für PostgreSQL) werden ignoriert, bis der Server neu gestartet wird. Weitere Informationen finden Sie in der [PostgreSQL-Dokumentation](#).

Weitere Informationen zum Aktivieren von CDC finden Sie unter [Aktivieren der Erfassung von Datenänderungen \(CDC\) mithilfe logischer Replikation](#).

Arbeiten mit von AWS verwalteten PostgreSQL-Datenbanken als DMS-Quelle

Sie können eine von AWS verwaltete PostgreSQL-DB-Instance als Quelle für AWS DMS verwenden. Sie können sowohl Volllast- als auch CDC-Aufgaben mit einer von AWS verwalteten PostgreSQL-Quelle ausführen.

Voraussetzungen für die Verwendung einer von AWS verwalteten PostgreSQL-Datenbank als DMS-Quelle

Führen Sie vor der Migration von Daten aus einer von AWS verwalteten PostgreSQL-Quelldatenbank die folgenden Schritte aus:

- Wir empfehlen, ein AWS-Benutzerkonto mit den erforderlichen Mindestberechtigungen für die PostgreSQL-DB-Instance als Benutzerkonto für den PostgreSQL-Quellendpunkt für AWS DMS zu verwenden. Von der Verwendung des Hauptkontos wird abgeraten. Das Konto muss über die Rolle `rds_superuser` und die Rolle `rds_replication` verfügen. Die Rolle `rds_replication` erteilt Berechtigungen zur Verwaltung von logischen Slots und zum Streamen von Daten mithilfe von logischen Slots.

Erstellen Sie vom Hauptbenutzerkonto aus mehrere Objekte für das verwendete Konto.

Informationen dazu, wie Sie diese erstellen, finden Sie unter [Migrieren einer Datenbank von Amazon RDS für PostgreSQL ohne Verwendung des Hauptbenutzerkontos](#).

- Wenn sich Ihre Quelldatenbank in einer Virtual Private Cloud (VPC) befindet, wählen Sie die VPC-Sicherheitsgruppe aus, die Zugriff auf die DB-Instance bietet, in der sich die Datenbank befindet. Dies ist erforderlich, damit die DMS-Replikations-Instance eine Verbindung mit der Quell-

DB-Instance herstellen kann. Wenn sich die Datenbank und die DMS-Replikations-Instance in derselben VPC befinden, fügen Sie die entsprechende Sicherheitsgruppe ihren eigenen Regeln für eingehenden Datenverkehr hinzu.

Note

Einige AWS DMS-Transaktionen sind für einige Zeit im Leerlauf, bevor die DMS-Engine sie erneut nutzt. Über den Parameter `idle_in_transaction_session_timeout` in PostgreSQL-Versionen 9.6 und höher können Sie für Transaktionen im Leerlauf ein Timeout und Fehlschlagen veranlassen. Beenden Sie Transaktionen im Leerlauf nicht, wenn Sie AWS DMS verwenden.

Aktivieren von CDC mit einer von AWS verwalteten PostgreSQL-DB-Instance mit AWS DMS

AWS DMS unterstützt CDC für Amazon-RDS-PostgreSQL-Datenbanken, wenn die DB-Instance so konfiguriert ist, dass sie logische Replikation verwendet. Die folgende Tabelle enthält eine Zusammenfassung der Kompatibilität der einzelnen von AWS verwalteten PostgreSQL-Versionen mit logischer Replikation.

Sie können keine RDS-PostgreSQL-Lesereplikate für CDC (laufende Replikation) verwenden.

PostgreSQL-Version	Unterstützung für Volllastaufgaben mit AWS DMS	Unterstützung für CDC mit AWS DMS
Aurora PostgreSQL Version 2.1 mit Kompatibilität mit PostgreSQL 10.5 (oder niedriger)	Ja	Nein
Aurora PostgreSQL Version 2.2 mit Kompatibilität mit PostgreSQL 10.6 (oder höher)	Ja	Ja
RDS für PostgreSQL mit Kompatibilität mit PostgreSQL 10.21 (oder höher)	Ja	Ja

So aktivieren Sie die logische Replikation für eine RDS PostgreSQL DB-Instance

1. Verwenden Sie das AWS-Hauptbenutzerkonto für die PostgreSQL-DB-Instance als Benutzerkonto für den PostgreSQL-Quellendpunkt. Das Masterbenutzerkonto hat die erforderlichen Rollen für die CDC-Einrichtung.

Wenn Sie ein anderes als das Hauptbenutzerkonto verwenden, müssen Sie vom Hauptbenutzerkonto aus mehrere Objekte für das verwendete Konto erstellen. Weitere Informationen finden Sie unter [Migrieren einer Datenbank von Amazon RDS für PostgreSQL ohne Verwendung des Hauptbenutzerkontos](#).

2. Legen Sie den Parameter `rds.logical_replication` in der DB-CLUSTER-Parametergruppe auf 1 fest. Für das Wirksamwerden dieses statischen Parameters ist ein Neustart der DB-Instance erforderlich. Im Rahmen der Anwendung dieses Parameters legt AWS DMS die Parameter `wal_level`, `max_wal_senders`, `max_replication_slots` und `max_connections` fest. Diese Parameteränderungen können die WAL-Erzeugung (Write Ahead Log) erhöhen. Setzen Sie `rds.logical_replication` also nur, wenn Sie logische Replikationsslots verwenden.
3. Der Parameter `wal_sender_timeout` beendet Replikationsverbindungen, die länger als die angegebene Anzahl von Millisekunden inaktiv sind. Der Standardwert für eine von AWS verwaltete PostgreSQL-Datenbank beträgt 30 000 Millisekunden (30 Sekunden). Wenn Sie den Wert auf 0 (Null) festlegen, wird der Timeout-Mechanismus deaktiviert. Dies ist eine gültige Einstellung für DMS.

Wenn Sie den Wert `wal_sender_timeout` auf einen anderen Wert als Null festlegen, benötigt eine DMS-Aufgabe mit CDC mindestens 10 000 Millisekunden (10 Sekunden) und schlägt fehl, wenn der Wert zwischen 0 und 10 000 liegt. Halten Sie den Wert unter 5 Minuten, um eine Verzögerung während eines Multi-AZ-Failovers einer DMS-Replikations-Instance zu vermeiden.

4. Stellen Sie sicher, dass der Wert des Parameters `max_worker_processes` in Ihrer DB-Cluster-Parametergruppe gleich oder höher als die kombinierten Gesamtwerte von `max_logical_replication_workers`, `autovacuum_max_workers` und `max_parallel_workers` ist. Eine hohe Anzahl von Hintergrund-Worker-Prozessen kann sich auf die Anwendungs-Workloads auf kleinen Instances auswirken. Überwachen Sie daher die Leistung Ihrer Datenbank, wenn Sie für `max_worker_processes` einen höheren Wert als den Standardwert festlegen.

Migrieren einer Datenbank von Amazon RDS für PostgreSQL ohne Verwendung des Hauptbenutzerkontos

In Einzelfällen können Sie das Hauptbenutzerkonto für die DB-Instance von Amazon RDS PostgreSQL, die Sie als Quelle verwenden, nicht verwenden. In diesen Fällen erstellen Sie mehrere Objekte, um Data Definition Language (DDL)-Ereignisse zu erfassen. Sie erstellen diese Objekte in dem Konto, das nicht das Masterkonto ist, und dann erstellen Sie einen Auslöser im Masterbenutzerkonto.

Note

Wenn Sie die Endpunkteinstellung `captureDDLs` am Quellendpunkt auf `false` festlegen, müssen Sie folgende Tabelle/folgenden Auslöser in der Quelldatenbank nicht erstellen.

Gehen Sie wie folgt vor, um diese Objekte zu erstellen.

So erstellen Sie Objekte

1. Wählen Sie das Schema aus, in dem die Objekte erstellt werden sollen. Das Standardschema ist `public`. Stellen Sie sicher, dass das Schema vorhanden ist und dass das *OtherThanMaster*-Konto darauf zugreifen kann.
2. Melden Sie sich mit dem anderen Benutzerkonto (nicht dem Hauptkonto) (hier das Konto *OtherThanMaster*) bei der PostgreSQL-DB-Instance an.
3. Erstellen Sie die Tabelle `awsdms_ddl_audit`, indem Sie den folgenden Befehl ausführen. Ersetzen Sie dabei *objects_schema* im folgenden Code durch den Namen des zu verwendenden Schemas.

```
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_SCHEMA_NAME. For now - holds
  current_schema
```



```
c_ddlqry text          -- The DDL query associated with the current DDL event
);
```

- Erstellen Sie die Funktion `awsdms_intercept_ddl`, indem Sie den folgenden Befehl ausführen und dabei `objects_schema` im folgenden Code durch den Namen des zu verwendenden Schemas ersetzen.

```
CREATE OR REPLACE FUNCTION objects_schema.awsdms_intercept_ddl()
  RETURNS event_trigger
LANGUAGE plpgsql
SECURITY DEFINER
  AS $$
  declare _qry text;
BEGIN
  if (tg_tag='CREATE TABLE' or tg_tag='ALTER TABLE' or tg_tag='DROP TABLE' or
  tg_tag = 'CREATE TABLE AS') then
    SELECT current_query() into _qry;
    insert into objects_schema.awsdms_ddl_audit
    values
    (
      default,current_timestamp,current_user,cast(TXID_CURRENT()as
      varchar(16)),tg_tag,0,'',current_schema,_qry
    );
    delete from objects_schema.awsdms_ddl_audit;
  end if;
END;
$$;
```

- Melden Sie sich vom *OtherThanMaster*-Konto ab und melden Sie sich mit einem Konto an, dem die `rds_superuser`-Rolle zugewiesen ist.
- Erstellen Sie den Ereignisauslöser `awsdms_intercept_ddl`, indem Sie den folgenden Befehl ausführen.

```
CREATE EVENT TRIGGER awsdms_intercept_ddl ON ddl_command_end
EXECUTE PROCEDURE objects_schema.awsdms_intercept_ddl();
```

7. Stellen Sie sicher, dass alle Benutzer und Rollen, die auf diese Ereignisse zugreifen, über die erforderlichen DDL-Berechtigungen verfügen. Beispielsweise:

```
grant all on public.aws_dms_ddl_audit to public;  
grant all on public.aws_dms_ddl_audit_c_key_seq to public;
```

Wenn Sie das vorangegangene Verfahren ausgeführt haben, können Sie den AWS DMS-Quellendpunkt mithilfe des *OtherThanMaster*-Kontos erstellen.

Note

Diese Ereignisse werden durch die Anweisungen CREATE TABLE, ALTER TABLE und DROP TABLE ausgelöst.

Aktivieren der Erfassung von Datenänderungen (CDC) mithilfe logischer Replikation

Sie können das native PostgreSQL-Feature zur logischen Replikation verwenden, um CDC während einer Datenbankmigration für PostgreSQL-Quellen zu aktivieren. Sie können dieses Feature mit einer selbstverwalteten PostgreSQL- und einer DB-Instance von Amazon RDS für PostgreSQL verwenden. Dieser Ansatz reduziert Ausfallzeiten und stellt sicher, dass die Zieldatenbank mit der PostgreSQL-Quelldatenbank synchronisiert ist.

AWS DMS unterstützt CDC für PostgreSQL-Tabellen mit Primärschlüsseln. Wenn eine Tabelle keinen Primärschlüssel aufweist, enthalten die Write-Ahead (WAL)-Protokolle kein Vorher-Image der Datenbankzeile. In diesem Fall kann DMS die Tabelle nicht aktualisieren. Sie können dann zusätzliche Konfigurationseinstellungen verwenden und die Tabellenreplikidentität als Problemumgehung verwenden. Bei diesem Ansatz können jedoch zusätzliche Protokolle generiert werden. Es wird empfohlen, die Tabellenreplikidentität erst nach sorgfältigen Tests als Problemumgehung zu verwenden. Weitere Informationen finden Sie unter [Zusätzliche Konfigurationseinstellungen bei Verwendung einer PostgreSQL-Datenbank als DMS-Quelle](#).

Note

REPLICA IDENTITY FULL wird mit einem Plug-in für logische Dekodierung unterstützt, aber nicht mit dem Plug-in pglogical. Weitere Informationen finden Sie in der [Dokumentation zu pglogical](#).

AWS DMS verwendet bei Volllast-und-CDC- sowie reinen CDC-Aufgaben logische Replikations-Slots, um WAL-Protokolle für die Replikation beizubehalten, bis die Protokolle dekodiert werden. Beim Neustart (nicht bei der Fortsetzung) einer Volllast-und-CDC- oder einer reinen CDC-Aufgabe wird der Replikations-Slot neu erstellt.

Note

Für die logische Dekodierung verwendet DMS entweder das Plugin `test_decoding` oder `pglogical`. Wenn das Plug-in `pglogical` in einer PostgreSQL-Quelldatenbank verfügbar ist, erstellt DMS einen Replikations-Slot mithilfe von `pglogical`, andernfalls wird das Plug-in `test_decoding` verwendet. Weitere Informationen zum Plug-in `test_decoding` finden Sie in der [Dokumentation zu PostgreSQL](#).

Wenn der Datenbankparameter `max_slot_wal_keep_size` auf einen anderen Wert als den Standardwert festgelegt ist und die `restart_lsn` eines Replikations-Slots um mehr als diese Größe hinter der aktuellen LSN zurückbleibt, schlägt die DMS-Aufgabe aufgrund des Entfernens der erforderlichen WAL-Dateien fehl.

Konfigurieren des Plug-ins `pglogical`

Das als PostgreSQL-Erweiterung implementierte Plug-in `pglogical` ist ein logisches Replikationssystem und Modell für die selektive Datenreplikation. In der folgenden Tabelle sind die PostgreSQL-Quelldatenbankversionen aufgeführt, die das Plug-in `pglogical` unterstützen.

PostgreSQL-Quelle	Unterstützung für <code>pglogical</code>
Selbstverwaltetes PostgreSQL 9.4 oder höher	Ja
Amazon RDS PostgreSQL 9.5 oder niedriger	Nein
Amazon RDS PostgreSQL 9.6 oder höher	Ja
Aurora PostgreSQL 1.x bis 2.5.x	Nein
Aurora PostgreSQL 2.6.x oder höher	Ja
Aurora PostgreSQL 3.3.x oder höher	Ja

Bevor Sie `pglogical` für die Verwendung mit AWS DMS konfigurieren, aktivieren Sie zunächst die logische Replikation für die Erfassung von Datenänderungen (CDC) in Ihrer PostgreSQL-Quelldatenbank.

- Informationen zur Aktivierung der logischen Replikation für CDC auf selbstverwalteten PostgreSQL-Quelldatenbanken finden Sie unter [Aktivieren von CDC mit einer selbstverwalteten PostgreSQL-Datenbank als AWS DMS-Quelle](#).
- Informationen zur Aktivierung der logischen Replikation für CDC auf von AWS verwalteten PostgreSQL-Quelldatenbanken finden Sie unter [Aktivieren von CDC mit einer von AWS verwalteten PostgreSQL-DB-Instance mit AWS DMS](#).

Nachdem die logische Replikation in Ihrer PostgreSQL-Quelldatenbank aktiviert wurde, führen Sie die folgenden Schritte aus, um `pglogical` für die Verwendung mit DMS zu konfigurieren.

So verwenden Sie das Plug-in `pglogical` für die logische Replikation in einer PostgreSQL-Quelldatenbank mit AWS DMS

1. Erstellen Sie eine `pglogical`-Erweiterung für Ihre PostgreSQL-Quelldatenbank:
 - a. Legen Sie den richtigen Parameter fest:
 - Legen Sie für selbstverwaltete PostgreSQL-Datenbanken den Datenbankparameter `shared_preload_libraries= 'pglogical'` fest.
 - Legen Sie für PostgreSQL-Datenbanken in Amazon RDS und Datenbanken der Amazon-Aurora-PostgreSQL-kompatiblen Edition den Parameter `shared_preload_libraries` in derselben RDS-Parametergruppe auf `pglogical` fest.
 - b. Starten Sie Ihre PostgreSQL-Quelldatenbank neu.
 - c. Führen Sie in der PostgreSQL-Datenbank den Befehl `create extension pglogical;` aus.
2. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob `pglogical` erfolgreich installiert wurde:

```
select * FROM pg_catalog.pg_extension
```

Sie können jetzt eine AWS DMS-Aufgabe erstellen, die die Erfassung von Datenänderungen für Ihren PostgreSQL-Quelldatenbank-Endpoint durchführt.

Note

Wenn Sie `pglogical` in Ihrer PostgreSQL-Quelldatenbank nicht aktivieren, verwendet AWS DMS standardmäßig das Plug-in `test_decoding`. Wenn `pglogical` für die logische Dekodierung aktiviert ist, verwendet AWS DMS standardmäßig `pglogical`. Sie können jedoch das zusätzliche Verbindungsattribut `PluginName` festlegen, um stattdessen das Plug-in `test_decoding` zu verwenden.

Verwenden von nativen CDC-Startpunkten zum Einrichten einer CDC-Last einer PostgreSQL-Quelle

Legen Sie zum Aktivieren von nativen CDC-Startpunkten mit PostgreSQL als Quelle das zusätzliche Verbindungsattribut `slotName` auf den Namen eines vorhandenen logischen Replikations-Slots fest, wenn Sie den Endpunkt erstellen. Dieser logische Replikations-Slot enthält laufende Änderungen ab dem Zeitpunkt der Erstellung des Endpunkts und unterstützt somit die Replikation ab einem früheren Zeitpunkt.

PostgreSQL schreibt die Datenbankänderungen in WAL-Dateien, die nur verworfen werden, nachdem ein AWS DMS erfolgreich Änderungen aus dem logischen Replikations-Slot gelesen hat. Durch die Verwendung von logischen Replikations-Slots können protokollierte Änderungen vor dem Löschen geschützt werden, bevor sie vom Replikationsmodul verwendet werden.

Je nach Änderungsrate und Verbrauch können Änderungen in einem logischen Replikations-Slot jedoch zu einer erhöhten Festplattennutzung führen. Es wird empfohlen, Speicherplatznutzungsalarme in der PostgreSQL-Quell-Instance festzulegen, wenn Sie logische Replikations-Slots verwenden. Weitere Informationen zur Einstellung des zusätzlichen Verbindungsattributs `slotName` finden Sie unter [Endpunkteinstellungen und zusätzliche Verbindungsattribute \(ECAs\) bei Verwendung von PostgreSQL als DMS-Quelle](#).

Im folgenden Verfahren wird dieser Ansatz näher erläutert.

So verwenden Sie einen systemeigenen CDC-Startpunkt zum Einrichten einer CDC-Last eines PostgreSQL-Quellendpunkts:

1. Identifizieren Sie den logischen Replikations-Slot, der von einer früheren Replikationsaufgabe (einer übergeordneten Aufgabe) verwendet wird, die Sie als Startpunkt verwenden möchten. Fragen Sie dann die `pg_replication_slots`-Ansicht der Quelldatenbank ab, um

sicherzustellen, dass dieser Slot über keine aktiven Verbindungen verfügt. Wenn dies doch der Fall ist, lösen Sie sie auf und beenden Sie sie, bevor Sie fortfahren.

Für die folgenden Schritte gehen Sie davon aus, dass Ihr logischer Replikations-Slot `abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef` ist.

- Erstellen Sie einen neuen Quellendpunkt, der die folgenden zusätzlichen Verbindungsattributeinstellungen enthält.

```
slotName=abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef;
```

- Erstellen Sie eine neue reine CDC-Aufgabe mithilfe der Konsole, der AWS CLI oder der AWS DMS-API. Wenn Sie die CLI verwenden, können Sie beispielsweise den folgenden Befehl `create-replication-task` ausführen.

```
aws dms create-replication-task --replication-task-identifizier 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"
```

Im vorhergehenden Befehl werden die folgenden Optionen festgelegt:

- Die Option `source-endpoint-arn` ist auf den neuen Wert festgelegt, den Sie in Schritt 2 erstellt haben.
- Die Option `replication-instance-arn` ist auf den gleichen Wert wie für die übergeordnete Aufgabe aus Schritt 1 festgelegt.
- Die Optionen `table-mappings` und `replication-task-settings` sind auf die gleichen Werte wie für die übergeordnete Aufgabe aus Schritt 1 festgelegt.
- Die Option `cdc-start-position` ist auf einen Startpositions Wert festgelegt. Um diese Startposition zu finden, fragen Sie entweder die `pg_replication_slots`-Ansicht Ihrer Quelldatenbank ab oder sehen Sie sich die Konsolendetails für die übergeordnete Aufgabe an.

in Schritt 1 an. Weitere Informationen finden Sie unter [Bestimmen eines nativen CDC-Startpunkts](#).

Führen Sie die folgenden Schritte aus, um den benutzerdefinierten CDC-Startmodus zu aktivieren, wenn Sie eine neue reine CDC-Aufgabe mithilfe der AWS DMS-Konsole erstellen:

- Wählen Sie im Abschnitt Aufgabeneinstellungen für CDC-Startmodus für Quelltransaktionen die Option Benutzerdefinierten CDC-Startmodus aktivieren aus.
- Wählen Sie für Benutzerdefinierter CDC-Startpunkt für Quelltransaktionen die Option Protokollsequenznummer angeben aus. Geben Sie die Systemänderungsnummer an oder wählen Sie Wiederherstellungsprüfpunkt angeben aus und geben Sie einen Wiederherstellungsprüfpunkt an.

Wenn diese CDC-Aufgabe ausgeführt wird und der angegebene logische Replikations-Slot nicht vorhanden ist, löst AWS DMS einen Fehler aus. Außerdem wird ein Fehler ausgelöst, wenn die Aufgabe nicht mit einer gültigen Einstellung für `cdc-start-position` erstellt wurde.

Wenn Sie native CDC-Startpunkte mit dem Plug-in `pglogical` verwenden und einen neuen Replikations-Slot verwenden möchten, führen Sie die folgenden Einrichtungsschritte durch, bevor Sie eine CDC-Aufgabe erstellen.

So verwenden Sie einen neuen Replikations-Slot, der nicht zuvor als Teil einer anderen DMS-Aufgabe erstellt wurde

1. Erstellen Sie einen Replikations-Slot wie im Folgenden dargestellt:

```
SELECT * FROM pg_create_logical_replication_slot('replication_slot_name',  
'pglogical');
```

2. Nachdem die Datenbank den Replikations-Slot erstellt hat, rufen Sie die Werte `restart_lsn` und `confirmed_flush_lsn` für den Slot ab und notieren Sie sich diese:

```
select * from pg_replication_slots where slot_name like 'replication_slot_name';
```

Beachten Sie, dass die native CDC-Startposition für eine CDC-Aufgabe, die nach dem Replikations-Slot erstellt wurde, nicht älter sein darf als der Wert `confirmed_flush_lsn`.

Informationen zu den Werten `restart_lsn` und `confirmed_flush_lsn` finden Sie unter [pg_replication_slots](#)

- Erstellen Sie einen `pglogical`-Knoten.

```
SELECT pglogical.create_node(node_name := 'node_name', dsn := 'your_dsn_name');
```

- Erstellen Sie mit der Funktion `pglogical.create_replication_set` zwei Replikationssätze. Der erste Replikationssatz verfolgt Aktualisierungen und Löschungen für Tabellen mit Primärschlüsseln nach. Der zweite Replikationssatz verfolgt nur Einfügungen nach und hat den gleichen Namen wie der erste Replikationssatz mit dem zusätzlichen Präfix „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);
```

- Fügen Sie dem Replikationssatz eine Tabelle hinzu.

```
SELECT pglogical.replication_set_add_table('replication_slot_name',
'schemaname.tablename', true);
SELECT pglogical.replication_set_add_table('ireplication_slot_name',
'schemaname.tablename', true);
```

- Legen Sie das zusätzliche Verbindungsattribut (Extra Connection Attribute, ECA) wie folgt fest, wenn Sie Ihren Quellendpunkt erstellen.

```
PluginName=PGLOGICAL;slotName=slot_name;
```

Sie können jetzt mithilfe des neuen Replikations-Slots eine reine CDC-Aufgabe mit einem nativen PostgreSQL-Startpunkt erstellen. Weitere Informationen zum Plug-in `pglogical` finden Sie in der [Dokumentation zu `pglogical` 3.7](#).

Migrieren von PostgreSQL auf PostgreSQL mit AWS DMS

Wenn Sie von einer anderen Datenbank-Engine als PostgreSQL auf eine PostgreSQL-Datenbank migrieren, eignet sich AWS DMS fast immer als Migrationstool. Wenn Sie jedoch aus einer

PostgreSQL-Datenbank auf eine PostgreSQL-Datenbank migrieren, können native PostgreSQL-Tools effektiver sein.

Verwenden nativer PostgreSQL-Tools zur Datenmigration

Unter den folgenden Bedingungen empfehlen wir die Verwendung von PostgreSQL-Datenbank-Migrationstools wie `pg_dump`:

- Sie möchten eine homogene Migration durchführen, bei der die PostgreSQL-Ausgangsdatenbank auf eine PostgreSQL-Zielfdatenbank migriert werden soll.
- Sie möchten eine komplette Datenbank migrieren.
- Die nativen Tools erlauben Ihnen, Ihre Daten mit einer minimalen Ausfallzeit zu migrieren.

Das Dienstprogramm `pg_dump` verwendet den Befehl `COPY`, um ein Schema sowie eine Daten-Dumpdatei einer PostgreSQL-Datenbank zu erstellen. Das durch `pg_dump` erzeugte Dump-Skript lädt Daten in eine Datenbank mit dem gleichen Namen und erstellt die Tabellen, Indizes und Fremdschlüssel neu. Verwenden Sie den Befehl `pg_restore` und den Parameter `-d`, um die Daten in einer Datenbank mit einem anderen Namen wiederherzustellen.

Wenn Sie Daten aus einer PostgreSQL-Quelldatenbank, die auf EC2 ausgeführt wird, auf ein Ziel in Amazon RDS für PostgreSQL migrieren, können Sie das Plug-in `pglogical` verwenden.

Weitere Informationen zum Importieren einer PostgreSQL-Datenbank in Amazon RDS für PostgreSQL oder die Amazon-Aurora-PostgreSQL-kompatible Edition finden Sie unter <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/PostgreSQL.Procedural.Importing.html>.

Migration von Daten von PostgreSQL auf PostgreSQL mit DMS

AWS DMS kann Daten beispielsweise von einer On-Premises-PostgreSQL-Quelldatenbank auf eine Ziel-Instance in Amazon RDS für PostgreSQL oder Aurora PostgreSQL migrieren. Kern- oder grundlegende PostgreSQL-Datentypen werden meist erfolgreich migriert.

Note

Wenn Sie partitionierte Tabellen aus einer PostgreSQL-Quelle in ein PostgreSQL-Ziel replizieren, müssen Sie die übergeordnete Tabelle nicht als Teil der Auswahlkriterien in der DMS-Aufgabe angeben. Wenn Sie die übergeordnete Tabelle angeben, werden Daten in untergeordneten Tabellen auf dem Ziel dupliziert, was möglicherweise eine PK-Verletzung

verursacht. Wenn in den Auswahlkriterien für die Tabellenzuordnung nur untergeordnete Tabellen ausgewählt werden, wird die übergeordnete Tabelle automatisch aufgefüllt.

Datentypen, die zwar in der Quelldatenbank, nicht aber in der Zieldatenbank unterstützt werden, werden möglicherweise nicht erfolgreich migriert. AWS DMS streamt einige Datentypen als Zeichenfolgen, wenn der Datentyp unbekannt ist. Bei einigen Datentypen, wie z. B. XML und JSON, können kleine Dateien erfolgreich migriert werden, bei großen Dokumenten schlägt die Migration jedoch fehl.

Beachten Sie bei der Datentypmigration Folgendes:

- In Einzelfällen gibt der PostgreSQL-Datentyp NUMERIC(p,s) keine Präzision und Skalierung an. Für DMS-Versionen 3.4.2 und früher verwendet DMS standardmäßig eine Präzision von 28 und eine Skalierung von 6, NUMERIC(28,6). Beispielsweise wird der Wert 0,611111104488373 von der Quelle in 0,611111 auf dem PostgreSQL-Ziel konvertiert.
- Eine Tabelle mit einem ARRAY-Datentyp muss über einen Primärschlüssel verfügen. Eine Tabelle mit einem ARRAY-Datentyp ohne Primärschlüssel wird während Volllastvorgängen angehalten.

Die folgende Tabelle enthält PostgreSQL-Quelldatentypen sowie Erläuterungen dazu, ob diese erfolgreich migriert werden können:

Datentyp	Wird erfolgreich migriert	Wird teilweise migriert	Wird nicht migriert	Kommentare
INTEGER	X			
SMALLINT	X			
BIGINT	X			
NUMERIC/DECIMAL(p,s)		X		Wobei $0 < p < 39$ und $0 < s$
NUMERIC/DECIMAL		X		Wobei $p > 38$ oder $p=s=0$

Datentyp	Wird erfolgreich migriert	Wird teilweise migriert	Wird nicht migriert	Kommentare
REAL	X			
DOUBLE	X			
SMALLSERIAL	X			
SERIAL	X			
BIGSERIAL	X			
MONEY	X			
CHAR		X		Ohne angegebene Genauigkeit
CHAR(n)	X			
VARCHAR		X		Ohne angegebene Genauigkeit
VARCHAR (n)	X			
TEXT	X			
BYTEA	X			

Datentyp	Wird erfolgreich migriert	Wird teilweise migriert	Wird nicht migriert	Kommentare
TIMESTAMP (ZEITSTEMPEL)	X			Positive und negative Unendlichkeitswerte werden auf „9999-12-31 23:59:59“ bzw. „4713-01-01 00:00:00 BC“ gekürzt.
TIMESTAMP WITH TIME ZONE		X		
DATUM	X			
TIME	X			
TIME WITH TIME ZONE		X		
INTERVAL		X		
BOOLEAN	X			
ENUM			X	
CIDR	X			
INET			X	
MACADDR			X	
TSVECTOR			X	
TSQUERY			X	

Datentyp	Wird erfolgreich migriert	Wird teilweise migriert	Wird nicht migriert	Kommentare
XML		X		
POINT	X			PostGIS Spatial Data-Typ
LINE			X	
LSEG			X	
BOX			X	
PATH			X	
POLYGON	X			PostGIS Spatial Data-Typ
CIRCLE			X	
JSON		X		
ARRAY	X			Primärschlüssel erforderlich
COMPOSITE			X	
RANGE			X	
LINestring	X			PostGIS Spatial Data-Typ
MULTIPOINT	X			PostGIS Spatial Data-Typ

Datentyp	Wird erfolgreich migriert	Wird teilweise migriert	Wird nicht migriert	Kommentare
MULTILINESTRING	X			PostGIS Spatial Data-Typ
MULTIPOLYGON	X			PostGIS Spatial Data-Typ
GEOMETRYCOLLECTION	X			PostGIS Spatial Data-Typ

Migrieren räumlicher PostGIS-Datentypen

Räumliche Daten identifizieren die Geometrieinformationen eines Objekts oder einer Position im Raum. Objektrelationale PostgreSQL-Datenbanken unterstützen räumliche PostGIS-Datentypen.

Stellen Sie vor der Migration räumlicher PostgreSQL-Datenobjekte sicher, dass das Plug-in PostGIS auf globaler Ebene aktiviert ist. Dadurch wird sichergestellt, dass AWS DMS die genauen Quellspalten der räumlichen Daten für die PostgreSQL-Ziel-DB-Instance erstellt.

Bei homogenen Migrationen von PostgreSQL zu PostgreSQL unterstützt AWS DMS die Migration von geometrischen und geographischen (geodätischen Koordinaten) PostGIS-Datenobjekttypen und -subtypen wie den folgenden:

- POINT
- LINESTRING
- POLYGON
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION

Entfernen von AWS DMS-Artefakten aus einer PostgreSQL-Quelldatenbank

Um DDL-Ereignisse zu erfassen, erstellt AWS DMS in der PostgreSQL-Datenbank verschiedene Artefakte, wenn eine Migrationsaufgabe gestartet wird. Wenn die Aufgabe abgeschlossen ist, können Sie diese Artefakte entfernen.

Entfernen Sie die Artefakte, indem Sie die folgenden Anweisungen ausgeben (in der Reihenfolge, in der sie angezeigt werden), wobei {AmazonRDSMigration} das Schema ist, in dem die Artefakte erstellt wurden. Ein Schema sollte nur mit äußerster Vorsicht gelöscht werden. Löschen Sie nie ein Schema, das in Betrieb ist, besonders kein öffentliches.

```
drop event trigger awsdms_intercept_ddl;
```

Der Ereignisauslöser gehört nicht zu einem bestimmten Schema.

```
drop function {AmazonRDSMigration}.awsdms_intercept_ddl()  
drop table {AmazonRDSMigration}.awsdms_ddl_audit  
drop schema {AmazonRDSMigration}
```

Zusätzliche Konfigurationseinstellungen bei Verwendung einer PostgreSQL-Datenbank als DMS-Quelle

Sie können zusätzliche Konfigurationseinstellungen für die Migration von Daten aus einer PostgreSQL-Datenbank auf zwei Weisen hinzufügen:

- Sie können dem Attribut "Extra Connection" Werte hinzufügen, um DDL-Ereignisse zu erfassen und um das Schema anzugeben, in dem die betrieblichen DDL-Datenbankartefakte erstellt werden. Weitere Informationen finden Sie unter [Endpunkteinstellungen und zusätzliche Verbindungsattribute \(ECAs\) bei Verwendung von PostgreSQL als DMS-Quelle](#).
- Sie können Verbindungszeichenfolgenparameter außer Kraft setzen. Wählen Sie diese Option aus, um einen der folgenden Schritte auszuführen:
 - Interne AWS DMS-Parameter angeben. Solche Parameter sind selten erforderlich und werden daher nicht auf der Benutzeroberfläche angezeigt.
 - Geben Sie Pass-Through-Werte (passthru) für den spezifischen Datenbank-Client an. AWS DMS fügt Pass-Through-Parameter in die Verbindungszeichenfolge ein, die an den Datenbank-Client übergeben wird.
- Mit dem Parameter REPLICA IDENTITY auf Tabellenebene in PostgreSQL-Versionen 9.4 und höher können Sie kontrollieren, welche Informationen in Write-Ahead-Protokollen (WALs)

geschrieben werden. Dies gilt insbesondere für WALs, die Zeilen identifizieren, die aktualisiert oder gelöscht werden. `REPLICA IDENTITY FULL` zeichnet die alten Werte aller Spalten in der Zeile auf. Verwenden Sie `REPLICA IDENTITY FULL` sorgfältig für jede Tabelle, da `FULL` eine zusätzliche Anzahl an WAL generiert, die möglicherweise nicht erforderlich ist. Weitere Informationen finden Sie unter [ALTER TABLE-REPLICA IDENTITY](#).

Verwenden der `MapBooleanAsBoolean` PostgreSQL-Endpunkteinstellung

Sie können PostgreSQL-Endpunkteinstellungen verwenden, um einen booleschen Wert aus Ihrer PostgreSQL-Quelle einem Amazon-Redshift-Ziel als booleschen Wert zuzuordnen. Standardmäßig wird ein `BOOLEAN`-Typ als `varchar(5)` migriert. Sie können `MapBooleanAsBoolean` angeben, damit PostgreSQL den booleschen Wert als booleschen Wert migriert, wie im folgenden Beispiel gezeigt.

```
--postgres-sql-settings '{"MapBooleanAsBoolean": true}'
```

Beachten Sie, dass Sie diese Einstellung sowohl am Quell- als auch am Zielpunkt festlegen müssen, damit sie wirksam wird.

Da MySQL keinen `BOOLEAN`-Typ hat, sollten Sie bei der Migration von `BOOLEAN`-Daten nach MySQL eine Transformationsregel anstelle dieser Einstellung verwenden.

Endpunkteinstellungen und zusätzliche Verbindungsattribute (ECAs) bei Verwendung von PostgreSQL als DMS-Quelle

Sie können Endpunkteinstellungen und zusätzliche Verbindungsattribute (ECAs) verwenden, um Ihre PostgreSQL-Quelldatenbank zu konfigurieren. Sie geben Endpunkteinstellungen an, wenn Sie den Quellendpunkt mithilfe der AWS DMS Konsole oder mithilfe des Befehls `create-endpoint` in der [AWS CLI](#) mit der `--postgres-sql-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax erstellen.

Die folgende Tabelle zeigt die Endpunkteinstellungen und ECAs, die Sie mit PostgreSQL als Quelle verwenden können.


Attributname	Beschreibung
<code>CaptureDDLs</code>	Um DDL-Ereignisse zu erfassen, erstellt AWS DMS in der PostgreSQL-Datenbank verschiedene Artefakte, wenn die Aufgabe gestartet wird. Sie können diese

Attributname	Beschreibung
	<p>Artefakte zu einem späteren Zeitpunkt entfernen, wie im Abschnitt Entfernen von AWS DMS-Artefakten aus einer PostgreSQL-Quelldatenbank beschrieben.</p> <p>Wenn dieser Wert auf <code>false</code> gesetzt wird, müssen Sie für die Quelldatenbank keine Tabellen oder Auslöser erstellen.</p> <p>Gestreamte DDL-Ereignisse werden erfasst.</p> <p>Standardwert: <code>true</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--postgres-sql-settings '{"CaptureDDLs": true}'</code></p>
ConsumeMonotonicEvents	<p>Wird verwendet, um zu steuern, wie monolithische Transaktionen mit doppelten Protokollsequenznummern (Log Sequence Numbers, LSNs) repliziert werden. Wenn dieser Parameter auf <code>false</code> gesetzt ist, werden Ereignisse mit doppelten LSNs verarbeitet und auf dem Ziel repliziert. Wenn dieser Parameter auf <code>true</code> gesetzt ist, wird nur das erste Ereignis repliziert, während Ereignisse mit doppelten LSNs weder verarbeitet noch auf dem Ziel repliziert werden.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>false/true</code></p> <p>Beispiel: <code>--postgres-sql-settings '{"ConsumeMonotonicEvents": true}'</code></p>

Attributname	Beschreibung
DdlArtifactsSchema	<p>Legt das Schema fest, in dem die operativen DDL-Daten bankartefakte erstellt werden.</p> <p>Standardwert: öffentlich</p> <p>Zulässige Werte: String</p> <p>Beispiel: <code>--postgre-sql-settings '{"DdlArtifactsSchema": "xyzddlSchema"}'</code></p>
ExecuteTimeout	<p>Stellt die Zeitbeschränkung der Client-Anweisung für die PostgreSQL-Instance in Sekunden ein. Der Standardwert liegt bei 60 Sekunden.</p> <p>Beispiel: <code>--postgre-sql-settings '{"ExecuteTimeout": 100}'</code></p>
FailTasksOnLobTruncation	<p>Mit der Einstellung <code>true</code> bewirkt dieser Wert, dass eine Aufgabe fehlschlägt, wenn die tatsächliche Größe einer LOB-Spalte größer als der angegebene <code>LobMaxSize</code> - Wert ist.</p> <p>Wenn die Aufgabe auf den eingeschränkten LOB-Modus festgelegt ist und diese Option auf <code>"true"</code> festgelegt wurde, schlägt die Aufgabe fehl, anstatt LOB-Daten zu kürzen.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: Boolesch</p> <p>Beispiel: <code>--postgre-sql-settings '{"FailTasksOnLobTruncation": true}'</code></p>

Attributname	Beschreibung
fetchCacheSize	<p>Dieses zusätzliche Verbindungsattribut (ECA) legt die Anzahl der Zeilen fest, die der Cursor während des Volllastvorgangs abruft. Abhängig von den in der Replikations-Instance verfügbaren Ressourcen können Sie einen höheren oder niedrigeren Wert festlegen.</p> <p>Standardwert: 10000</p> <p>Zulässige Werte: Zahl</p> <p>ECA-Beispiel: <code>fetchCacheSize=10000;</code></p>
HeartbeatFrequency	<p>Stellt die WAL-Heartbeat-Frequenz ein (in Minuten).</p> <p>Standardwert: 5</p> <p>Zulässige Werte: Zahl</p> <p>Beispiel: <code>--postgres-sql-settings '{"HeartbeatFrequency": 1}'</code></p>
HeartbeatSchema	<p>Legt das Schema fest, in dem die Heartbeat-Artefakte erstellt werden.</p> <p>Standardwert: public</p> <p>Zulässige Werte: String</p> <p>Beispiel: <code>--postgres-sql-settings '{"HeartbeatSchema": "xyzheartbeatSchema"}'</code></p>
MapJsonbAsClob	<p>Standardmäßig ordnet AWS DMS JSONB NCLOB zu. Sie können MapJsonbAsClob angeben, damit PostgreSQL den JSONB-Typ als CLOB migriert.</p> <p>Beispiel: <code>--postgres-sql-settings='{"MapJsonbAsClob": "true"}'</code></p>

Attributname	Beschreibung
MapLongVarcharAs	<p>Standardmäßig ordnet AWS DMS VARCHAR WSTRING zu. Sie können MapLongVarcharAs angeben, damit PostgreSQL den Typ VARCHAR(N) (wobei N größer als 16 387 ist) zu den folgenden Typen migriert:</p> <ul style="list-style-type: none">• WSTRING• CLOB• NCLOB <p>Beispiel: <code>--postgre-sql-settings='{\"MapLongVarcharAs\": \"CLOB\"}'</code></p>

Attributname	Beschreibung
MapUnboundedNumericAsString	<p>Dieser Parameter behandelt Spalten mit unbegrenzten NUMERIC-Datentypen als STRING, um eine erfolgreiche Migration ohne Verlust der Präzision des numerischen Werts zu ermöglichen. Verwenden Sie diesen Parameter nur für die Replikation von einer PostgreSQL-Quelle zu einem PostgreSQL-Ziel oder für Datenbanken mit PostgreSQL-Kompatibilität.</p> <p>Standardwert: false</p> <p>Zulässige Werte: false/true</p> <p>Beispiel: <code>--postgre-sql-settings '{"MapUnboundedNumericAsString": true}'</code></p> <p>Die Verwendung dieses Parameters kann zu einer gewissen Beeinträchtigung der Replikationsleistung führen, da eine Transformation von numerischen Werten in Zeichenfolgen und zurück in numerische Werte erfolgt. Dieser Parameter wird für DMS-Version 3.4.4 und höher unterstützt</p> <div data-bbox="688 1194 1508 1761" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Verwenden Sie <code>MapUnboundedNumericAsString</code> nur in PostgreSQL-Quell- und -Zielendpunkten gemeinsam.</p><p>Bei Verwendung von <code>MapUnboundedNumericAsString</code> für PostgreSQL-Quellendpunkte wird die Präzision während CDC auf 28 beschränkt. Bei Verwendung von <code>MapUnboundedNumericAsString</code> für Zielendpunkte werden Daten mit einer Präzision von 28 und einer Skalierung von 6 migriert.</p></div>

Attributname	Beschreibung
	<p>Verwenden Sie <code>MapUnboundedNumericAsString</code> nicht mit Nicht-PostgreSQL-Zielen.</p>
PluginName	<p>Gibt das Plugin an, das zum Erstellen eines Replikations-Slots verwendet werden soll.</p> <p>Zulässige Werte: <code>pglogical</code> , <code>test_decoding</code></p> <p>Beispiel: <code>--postgre-sql-settings '{"Plugin Name": "test_decoding"}'</code></p>

Attributname	Beschreibung
SlotName	<p>Legt den Namen eines zuvor erstellten logischen Replikations-Slots für eine CDC-Last der PostgreSQL-Quell-Instance fest.</p> <p>Bei Verwendung mit dem <code>AWS DMS-APICdcStartPosition</code>-Anforderungsparameter ermöglicht dieses Attribut auch die Verwendung nativer CDC-Startpunkte. Das DMS überprüft, ob der angegebene logische Replikations-Slot vorhanden ist, bevor die Aufgabe für das CDC-Laden gestartet wird. Es wird auch überprüft, ob die Aufgabe mit einer gültigen Einstellung von <code>CdcStartPosition</code> erstellt wurde. Wenn der angegebene Slot nicht existiert oder die Aufgabe keine gültige <code>CdcStartPosition</code>-Einstellung besitzt, löst DMS einen Fehler aus.</p> <p>Weitere Informationen zu den Einstellungen für den <code>CdcStartPosition</code>-Anforderungsparameter finden Sie unter Bestimmen eines nativen CDC-Startpunkts. Weitere Informationen über <code>CdcStartPosition</code> finden Sie in der Dokumentation zu den API-Operationen <code>CreateReplicationTask</code>, <code>StartReplicationTask</code> und <code>ModifyReplicationTask</code> in der API-Referenz zum AWS Database Migration Service.</p> <p>Zulässige Werte: String</p> <p>Beispiel: <code>--postgres-sql-settings '{"SlotName": "abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef"}'</code></p>
unboundedVarcharMaxSize	<p>Dieses zusätzliche Verbindungsattribut (Extra Connection Attribute, ECA) definiert die maximale Größe einer Datenspalte, die als Typ <code>VarChar</code> ohne Angabe der maximalen Länge definiert ist. Der Standardwert ist 8 000 Byte. Der maximale Wert ist 10485760 Byte.</p>

Einschränkungen bei Verwendung einer PostgreSQL-Datenbank als DMS-Quelle

Die folgenden Einschränkungen gelten bei der Verwendung von PostgreSQL als Quelle für AWS DMS:

- AWS DMS funktioniert weder mit Amazon RDS für PostgreSQL 10.4 noch mit Amazon Aurora PostgreSQL 10.4 als Quelle oder Ziel.
- Eine erfasste Tabelle muss über einen Primärschlüssel verfügen. Wenn eine Tabelle nicht über einen Primärschlüssel verfügt, ignoriert AWS DMS DELETE- und UPDATE-Vorgänge für Datensätze in dieser Tabelle. Informationen zur Problemumgehung finden Sie unter [Aktivieren der Erfassung von Datenänderungen \(CDC\) mithilfe logischer Replikation](#).

Hinweis: Wir raten von Migrationen ohne Primärschlüssel/eindeutigen Index ab. Andernfalls gelten zusätzliche Einschränkungen, z. B. KEINE Möglichkeit zur Stapelanwendung, vollständige LOB-Fähigkeit, Datenvalidierung und keine Möglichkeit, effizient auf das Redshift-Ziel zu replizieren.

- AWS DMS ignoriert einen Versuch, ein Segment eines Primärschlüssels zu aktualisieren. In diesen Fällen identifiziert die Zieldatenbank die Aktualisierung als eine, die keine Zeilen aktualisiert hat. Da die Ergebnisse der Aktualisierung eines Primärschlüssels in PostgreSQL jedoch unvorhersehbar sind, werden keine Datensätze in die Ausnahmetabelle geschrieben.
- AWS DMS unterstützt die Ausführungsoption Start Process Changes from Timestamp (Prozessänderungen von Zeitstempel starten) nicht.
- AWS DMS repliziert keine Änderungen, die sich aus Partitions- oder Unterpartitionsoperationen (ADD, DROP oder TRUNCATE) ergeben.
- Replikation mehrerer Tabellen mit demselben Namen, wobei sich jedoch jeder Name in Groß-/Kleinschreibung unterscheidet (z. B. tabelle1, TABELLE1 und Tabelle1), kann unvorhersehbares Verhalten verursachen. Aufgrund dieses Problems unterstützt AWS DMS diese Art der Replikation nicht.
- In den meisten Fällen unterstützt AWS DMS Änderungsverarbeitung von CREATE-, ALTER- und DROP DDL-Anweisungen für Tabellen. AWS DMS unterstützt diese Änderungsverarbeitung nicht, wenn die Tabellen in einer inneren Funktion oder einem Procedure-Body-Block oder in anderen verschachtelten Konstrukten gehalten werden.

Die folgende Änderung wird beispielsweise nicht erfasst:

```
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;
$$;
```

- Derzeit werden boolean-Datentypen in einer PostgreSQL-Quelle als bit-Datentyp mit inkonsistenten Werten zu einem SQL-Server-Ziel migriert. Erstellen Sie als Problemumgehung die Tabelle vorab mit einem VARCHAR(1)-Datentyp für die Spalte (oder lassen Sie AWS DMS die Tabelle erstellen). Dann lassen Sie die nachgeschaltete Verarbeitung ein „F“ als „False“ und ein „T“ als „True“ behandeln.
- AWS DMS unterstützt die Änderungsverarbeitung von TRUNCATE-Operationen nicht.
- Der OID LOB-Datentyp wird nicht auf das Ziel migriert.
- AWS DMS unterstützt den PostGIS-Datentyp nur für homogene Migrationen.
- Wenn es sich bei Ihrer Quelle um eine PostgreSQL-Datenbank handelt, die On-Premises oder auf einer Amazon-EC2-Instance ausgeführt wird, stellen Sie sicher, dass das Ausgabe-Plug-in test_decoding auf dem Quellendpunkt installiert ist. Sie finden dieses Plug-in im PostgreSQL-Paket contrib. Weitere Informationen zum Test-Decodier-Plugin finden Sie in der [PostgreSQL-Dokumentation](#).
- AWS DMS unterstützt keine Änderungsverarbeitung zum Festlegen und Aufheben von Spaltenstandardwerten (mit der Klausel ALTER COLUMN SET DEFAULT in ALTER TABLE-Anweisungen).
- AWS DMS unterstützt keine Änderungsverarbeitung zum Festlegen von NULL-Zulässigkeit der Spalte (mit der Klausel ALTER COLUMN [SET|DROP] NOT NULL in ALTER TABLE-Anweisungen).
- Wenn die logische Replikation aktiviert ist, beträgt die maximale Anzahl von Änderungen, die pro Transaktion im Arbeitsspeicher gespeichert werden, 4 MB. Danach werden die Änderungen auf den Datenträger übertragen. Daher nimmt der Wert für ReplicationSlotDiskUsage zu und restart_lsn schreitet erst voran, wenn die Transaktion abgeschlossen oder angehalten und das Rollback beendet ist. Da es sich um eine lange Transaktion handelt, kann das Rollback lange dauern. Vermeiden Sie daher lang laufende Transaktionen oder viele Untertransaktionen, wenn die logische Replikation aktiviert ist. Teilen Sie die Transaktion stattdessen in mehrere kleinere Transaktionen auf.

In Aurora PostgreSQL Version 13 und höher können Sie den `logical_decoding_work_mem` Parameter so einstellen, dass gesteuert wird, wann DMS-Spills Daten auf die Festplatte ändern. Weitere Informationen finden Sie unter [Überlaufdateien in Aurora PostgreSQL](#).

- Eine Tabelle mit einem ARRAY-Datentyp muss über einen Primärschlüssel verfügen. Eine Tabelle mit einem ARRAY-Datentyp ohne Primärschlüssel wird während Volllastvorgängen angehalten.
- AWS DMS unterstützt die Replikation partitionierter Tabellen nicht. Wenn eine partitionierte Tabelle erkannt wird, geschieht Folgendes:
 - Der Endpunkt meldet eine Liste von über- und untergeordneten Tabellen.
 - AWS DMS erstellt die Tabelle auf der Zieldatenbank als eine reguläre Tabelle mit den gleichen Eigenschaften wie die ausgewählten Tabellen.
 - Wenn die übergeordnete Tabelle in der Quelldatenbank denselben Primärschlüsselwert wie ihre untergeordneten Tabellen hat, wird ein "Duplikatschlüssel" generiert.
- Um partitionierte Tabellen aus einer PostgreSQL-Quelle auf ein PostgreSQL-Ziel zu replizieren, müssen Sie zunächst die über- und untergeordneten Tabellen im Ziel manuell erstellen. Definieren Sie anschließend eine separate Aufgabe, um auf diese Tabellen zu replizieren. Legen Sie die Aufgabenkonfiguration in solchen Fällen auf Vor dem Laden kürzen fest.
- Der PostgreSQL-Datentyp NUMERIC verfügt über keine feste Größe. Wenn Sie Daten mit dem Datentyp NUMERIC, jedoch ohne Präzision und Skalierung übertagen, verwendet DMS standardmäßig NUMERIC(28, 6) (mit 28 als Genauigkeit und 6 als Skalierung). Beispielsweise wird der Wert 0,611111104488373 von der Quelle auf dem PostgreSQL-Ziel in 0,611111 konvertiert.
- AWS DMS unterstützt Aurora PostgreSQL Serverless V1 nur als Quelle für Volllastaufgaben. AWS DMS unterstützt Aurora PostgreSQL Serverless V2 als Quelle für Volllast-, Volllast-und-CDC- und reine CDC-Aufgaben.
- AWS DMS unterstützt nicht die Replikation einer Tabelle mit einem eindeutigen Index, der mit einer coalesce-Funktion erstellt wurde.
- Bei Verwendung des LOB-Modus müssen die Quelltable und die entsprechende Zieltabelle einen identischen Primärschlüssel aufweisen. Wenn eine der Tabellen keinen Primärschlüssel aufweist, ist das Ergebnis der DELETE- und UPDATE-Datensatzoperationen nicht vorhersehbar.
- Bei Verwendung des Features für paralleles Laden wird die Tabellensegmentierung nach Partitionen oder Unterpartitionen nicht unterstützt. Weitere Informationen zum parallelen Laden finden Sie unter [Verwendung des parallelen Ladens für ausgewählte Tabellen und Ansichten](#).
- AWS DMS unterstützt keine verzögerten Einschränkungen.
- AWS DMS Version 3.4.7 unterstützt PostgreSQL 14.x als Quelle mit diesen Einschränkungen:
 - AWS DMS unterstützt keine Änderungsverarbeitung von zweiphasigen Commits.
 - AWS DMS unterstützt keine logische Replikation zum Streamen langer laufender Transaktionen.

- AWS DMS unterstützt CDC für Amazon-RDS-Proxy nicht für PostgreSQL als Quelle.
- Bei der Verwendung von [Quellfiltern](#), die keine Primärschlüsselspalte enthalten, werden DELETE-Operationen nicht erfasst.
- Wenn Ihre Quelldatenbank auch ein Ziel für ein anderes Replikationssystem eines Drittanbieters ist, werden DDL-Änderungen während CDC möglicherweise nicht migriert. Diese Situation kann das Auslösen des Ereignisauslösers `awsdms_intercept_ddl` verhindern. Ändern Sie diesen Auslöser in der Quelldatenbank wie folgt, um das Problem zu umgehen:

```
alter event trigger awsdms_intercept_ddl enable always;
```

- AWS DMS unterstützt CDC für Datenbank-Cluster von Amazon RDS Multi-AZ nicht für PostgreSQL als Quelle, da Datenbank-Cluster von RDS für PostgreSQL Multi-AZ keine logische Replikation unterstützen.

Quelldatentypen für PostgreSQL

In der folgenden Tabelle sind die PostgreSQL-Quelldatentypen, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung zu AWS DMS-Datentypen aufgeführt.

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

PostgreSQL-Datentypen	DMS-Datentypen
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
NUMERIC (p,s)	Wenn die Präzision 0 bis 38 ist, verwenden Sie NUMERIC. Wenn die Präzision 39 oder größer ist, verwenden Sie STRING.

PostgreSQL-Datentypen	DMS-Datentypen
DECIMAL(P,S)	Wenn die Präzision 0 bis 38 ist, verwenden Sie NUMERIC. Wenn die Präzision 39 oder größer ist, verwenden Sie STRING.
REAL	REAL4
DOUBLE	REAL8
SMALLSERIAL	INT2
SERIAL	INT4
BIGSERIAL	INT8
MONEY	NUMERIC(38,4) Der Datentyp MONEY ist in SQL Server FLOAT zugeordnet.
CHAR	WSTRING (1)
CHAR(N)	WSTRING (n)
VARCHAR(N)	WSTRING (n)
TEXT	NCLOB
BYTEA	BLOB
TIMESTAMP (ZEITSTEMPEL)	DATETIME
TIMESTAMP WITH TIME ZONE	DATETIME
DATUM	DATUM
TIME	TIME
TIME WITH TIME ZONE	TIME

PostgreSQL-Datentypen	DMS-Datentypen
INTERVAL	STRING (128) – 1 JAHR, 2 MONATE, 3 TAGE, 4 STUNDEN, 5 MINUTEN, 6 SEKUNDEN
BOOLEAN	CHAR (5) "false" oder "true"
ENUM	STRING (64)
CIDR	STRING (50)
INET	STRING (50)
MACADDR	STRING (18)
BIT (n)	STRING (n)
BIT VARYING (n)	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

PostgreSQL-Datentypen	DMS-Datentypen
JSONB	NCLOB
ARRAY	NCLOB
COMPOSITE	NCLOB
HSTORE	NCLOB
INT4RANGE	STRING (255)
INT8RANGE	STRING (255)
NUMRANGE	STRING (255)
STRRANGE	STRING (255)

Arbeiten mit LOB-Quelldatentypen für PostgreSQL

PostgreSQL-Spaltengrößen wirken sich auf die Konvertierung von PostgreSQL-LOB-Datentypen in AWS DMS-Datentypen aus. Um damit zu arbeiten, führen Sie für die folgenden AWS DMS-Datentypen folgende Schritte aus:

- BLOB – Legen Sie bei der Aufgabenerstellung für LOB-Grenze begrenzen auf den Wert Maximale LOB-Größe (KB) fest.
- CLOB – Die Replikation verarbeitet jedes Zeichen als UTF8-Zeichen. Suchen Sie daher nach der Länge des längsten Zeichentextes in der Spalte, hier als `max_num_chars_text` gezeigt. Nutzen Sie diese Länge, um den Wert für LOB-Größe begrenzen auf anzugeben. Wenn die Daten 4-Byte-Zeichen enthalten, multiplizieren Sie diese mit 2, um den Wert Limit LOB size to (LOB-Größe begrenzen auf) festzulegen, da dieser in Bytes angegeben wird. In diesem Fall ist Limit LOB size to (LOB-Größe begrenzen auf) gleich `max_num_chars_text`, multipliziert mit 2.
- NCLOB – Die Replikation verarbeitet jedes Zeichen als Doppelbyte-Zeichen. Suchen Sie daher nach der Länge des längsten Zeichentextes in der Spalte (`max_num_chars_text`) und multiplizieren ihn mit 2. Auf diese Weise geben Sie den Wert für LOB-Größe begrenzen auf an. In diesem Fall ist Limit LOB size to (LOB-Größe begrenzen auf) gleich `max_num_chars_text`, multipliziert mit 2. Wenn die Daten 4-Byte-Zeichen enthalten,

multiplizieren Sie diese erneut mit 2. In diesem Fall ist Limit LOB size to (LOB-Größe begrenzen auf) gleich `max_num_chars_text`, multipliziert mit 4.

Verwenden einer MySQL-kompatiblen Datenbank als Quelle für AWS DMS

Mit dem Database Migration Service können Sie Daten aus jeder MySQL-kompatiblen Datenbank (MySQL, MariaDB oder Amazon Aurora MySQL) migrieren. AWS

Informationen zu den Versionen von MySQL, die AWS DMS als Quelle unterstützt, finden Sie unter [Quellen für AWS DMS](#).

Sie können mit SSL Verbindungen zwischen Ihrem MySQL-kompatiblen Endpunkt und der Replikations-Instance verschlüsseln. Weitere Informationen zur Verwendung von SSL mit einem MySQL-kompatiblen Endpunkt finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).

In den folgenden Abschnitten bezieht sich der Begriff „selbstverwaltet“ auf Datenbanken, die entweder On-Premises oder in Amazon EC2 installiert sind. Der Begriff „von AWS verwaltet“ bezieht sich auf Datenbanken in Amazon RDS, Amazon Aurora oder Amazon S3.

Weitere Informationen zur Arbeit mit MySQL-kompatiblen Datenbanken und finden Sie in den AWS DMS folgenden Abschnitten.

Themen

- [Migrieren von MySQL auf MySQL mit AWS DMS](#)
- [Verwendung einer beliebigen MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#)
- [Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#)
- [Verwendung einer AWS-verwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#)
- [Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS](#)
- [Unterstützung für XA-Transaktionen](#)
- [Endpunkteinstellungen bei Verwendung von MySQL als Quelle für AWS DMS](#)
- [Quelldatentypen für MySQL](#)

Migrieren von MySQL auf MySQL mit AWS DMS

Für eine heterogene Migration, bei der Sie von einer anderen Datenbank-Engine als MySQL zu einer MySQL-Datenbank migrieren, AWS DMS ist dies fast immer das beste Migrationstool. Für eine

homogene Migration, bei der Sie von einer MySQL-Datenbank auf eine MySQL-Datenbank migrieren, empfehlen wir jedoch, ein Migrationsprojekt für homogene Datenmigrationen zu verwenden. Bei homogenen Datenmigrationen werden native Datenbanktools verwendet, um im Vergleich zu AWS DMS eine bessere Leistung und Genauigkeit bei der Datenmigration zu erzielen.

Verwendung einer beliebigen MySQL-kompatiblen Datenbank als Quelle für AWS DMS

Bevor Sie beginnen, mit einer MySQL-Datenbank als Quelle für zu arbeiten AWS DMS, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen. Diese Voraussetzungen gelten entweder für selbstverwaltete oder für AWS-verwaltete Quellen.

Sie müssen über ein Konto verfügen AWS DMS , das die Rolle des Replikationsadministrators innehat. Die Rolle benötigt die folgenden Berechtigungen:

- REPLICATION CLIENT – Diese Berechtigung ist nur für CDC-Aufgaben erforderlich. Mit anderen Worten, für full-load-only Aufgaben ist dieses Recht nicht erforderlich.
- REPLICATION SLAVE – Diese Berechtigung ist nur für CDC-Aufgaben erforderlich. Mit anderen Worten, für full-load-only Aufgaben ist dieses Privileg nicht erforderlich.
- SUPER – Diese Berechtigung ist nur für MySQL-Versionen vor 5.6.6 erforderlich.

Der AWS DMS Benutzer muss außerdem über SELECT-Rechte für die Quelltabellen verfügen, die für die Replikation vorgesehen sind.

Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS

Sie können die folgenden selbstverwalteten MySQL-kompatiblen Datenbanken als Quellen für AWS DMS verwenden:

- MySQL Community Edition
- MySQL Standard Edition
- MySQL Enterprise Edition
- MySQL Cluster Carrier Grade Edition
- MariaDB Community Edition
- MariaDB Enterprise Edition
- MariaDB Column Store

Stellen Sie sicher, dass die binäre Protokollierung aktiviert ist, damit Sie CDC verwenden können. Um die binäre Protokollierung zu aktivieren, müssen die folgenden Parameter in der MySQL-Datei `my.ini` (Windows) oder `my.cnf` (UNIX) konfiguriert werden.

Parameter	Wert
<code>server_id</code>	Legen Sie diesen Parameter auf einen Wert von 1 oder größer fest.
<code>log-bin</code>	Legen Sie den Pfad zur binären Protokolldatei fest, z. B. <code>log-bin=E:\MySQL_Logs\BinLog</code> . Schließen Sie nicht die Dateierweiterung ein.
<code>binlog_format</code>	Legen Sie diesen Parameter auf <code>ROW</code> fest. Wir empfehlen, diese Einstellung während der Replikation zu verwenden, da es in gewissen Fällen zu Inkonsistenzen bei der Replikation von Daten auf das Ziel kommen kann, wenn <code>binlog_format</code> auf <code>STATEMENT</code> festgelegt ist. Die Datenbank-Engine schreibt auch ähnlich inkonsistente Daten in das Ziel, wenn <code>binlog_format</code> auf <code>MIXED</code> festgelegt ist, da die Datenbank-Engine automatisch zur <code>STATEMENT</code> -basierten Protokollierung wechselt. Dies kann dazu führen, dass inkonsistente Daten in die Zieldatenbank geschrieben werden.
<code>expire_logs_days</code>	Legen Sie diesen Parameter auf einen Wert von 1 oder größer fest. Um zu verhindern, dass zu viel Speicherplatz verwendet wird, empfehlen wir, nicht den Standardwert 0 zu verwenden.
<code>binlog_checksum</code>	Setzen Sie diesen Parameter <code>NONE</code> für DMS-Version 3.4.7 oder früher auf.
<code>binlog_row_image</code>	Legen Sie diesen Parameter auf <code>FULL</code> fest.
<code>log_slave_updates</code>	Setzen Sie diesen Parameter auf <code>TRUE</code> , wenn Sie eine Read Replica von MySQL oder MariaDB als Quelle verwenden.

Sofern Ihre Quelle die (geclusterte) NDB-Datenbank-Engine nutzt, müssen die folgenden Parameter konfiguriert werden, um CDC auf Tabellen zu aktivieren, die diese Speicher-Engine verwenden. Fügen Sie diese Änderungen in der MySQL-Datei `my.ini` (Windows) oder `my.cnf` (UNIX) hinzu.

Parameter	Wert
<code>ndb_log_bin</code>	Legen Sie diesen Parameter auf ON fest. Dieser Wert stellt sicher, dass Änderungen in geclusterten Tabellen im binären Protokoll protokolliert werden.
<code>ndb_log_u pdate_as_write</code>	Legen Sie diesen Parameter auf OFF fest. Dieser Wert verhindert das Schreiben von UPDATE-Anweisungen als INSERT-Anweisungen in das Binärprotokoll.
<code>ndb_log_u pdated_only</code>	Legen Sie diesen Parameter auf OFF fest. Dieser Wert stellt sicher, dass das Binärprotokoll die gesamte Zeile enthält und nicht nur die geänderten Spalten.

Verwendung einer AWS-verwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS

Sie können die folgenden AWS verwalteten MySQL-kompatiblen Datenbanken als Quellen für verwenden: AWS DMS

- MySQL Community Edition
- MariaDB Community Edition
- Amazon Aurora MySQL-Compatible Edition

Wenn Sie eine von AWS-managed MySQL-kompatible Datenbank als Quelle für verwenden, stellen Sie sicher AWS DMS, dass Sie die folgenden Voraussetzungen für CDC erfüllen:

- Aktivieren Sie automatische Backups auf Instance-Ebene, um Binärprotokolle für RDS für MySQL und für RDS für MariaDB zu aktivieren. Ändern Sie die Variable `binlog_format` in der Parametergruppe, um Binärprotokolle für einen Aurora-MySQL-Cluster zu aktivieren.

Weitere Informationen zum Einrichten von automatischen Backups finden Sie unter [Working with automated backups](#) im Benutzerhandbuch für Amazon RDS.

Weitere Informationen zum Einrichten der Binärprotokollierung für eine Datenbank von Amazon RDS für MySQL finden Sie unter [Setting the binary logging format](#) im Benutzerhandbuch für Amazon RDS.

Weitere Informationen zum Einrichten der Binärprotokollierung für einen Aurora-MySQL-Cluster finden Sie unter [Wie aktiviere ich die Binärprotokollierung für meinen Amazon-Aurora-MySQL-kompatiblen Cluster?](#).

- Wenn Sie CDC verwenden möchten, aktivieren Sie die Binärprotokollierung. Weitere Informationen zum Einrichten der Binärprotokollierung für eine Datenbank von Amazon RDS für MySQL finden Sie unter [Setting the binary logging format](#) im Benutzerhandbuch für Amazon RDS.
- Stellen Sie sicher, dass die Binärprotokolle für verfügbar sind. AWS DMS Da AWS-managed MySQL-kompatible Datenbanken die Binärprotokolle so schnell wie möglich löschen, sollten Sie die Dauer erhöhen, für die die Protokolle verfügbar bleiben. Um z. B. die Aufbewahrungszeit der Protokolle auf 24 Stunden zu verlängern, führen Sie den folgenden Befehl aus.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Stellen Sie den Parameter `binlog_format` auf "ROW" ein.

Note

In MySQL oder MariaDB ist `binlog_format` ein dynamischer Parameter, sodass kein Neustart erforderlich ist, damit der neue Wert wirksam wird. Der neue Wert gilt jedoch nur für neue Sitzungen. Wenn Sie `binlog_format` zu Replikationszwecken in ROW ändern, kann Ihre Datenbank weiterhin nachfolgende Binärprotokolle mit dem Format MIXED erstellen, sofern diese Sitzungen gestartet wurden, bevor der Wert geändert wurde. Dadurch wird möglicherweise AWS DMS verhindert, dass alle Änderungen in der Quelldatenbank ordnungsgemäß erfasst werden. Wenn Sie die Einstellung `binlog_format` für eine MariaDB- oder MySQL-Datenbank ändern, starten Sie die Datenbank unbedingt neu, um alle vorhandenen Sitzungen zu beenden, oder starten Sie alle Anwendungen neu, die Data Manipulation Language (DML)-Operationen ausführen. Wenn Sie Ihre Datenbank zwingen, alle Sitzungen neu zu starten, nachdem Sie den `binlog_format` Parameter geändert haben, ROW wird sichergestellt, dass Ihre Datenbank alle nachfolgenden Änderungen an der Quelldatenbank im richtigen Format schreibt, sodass diese Änderungen ordnungsgemäß erfasst werden AWS DMS können.

- Stellen Sie den Parameter `binlog_row_image` auf "Full" ein.

- Setzen Sie den `binlog_checksum` Parameter "NONE" für DMS-Version 3.4.7 oder früher auf. Weitere Informationen zum Festlegen von Parametern in Amazon RDS MySQL finden Sie unter [Working with automated backups](#) im Benutzerhandbuch für Amazon RDS.
- Wenn Sie ein Lesereplikat von Amazon RDS MySQL oder Amazon RDS MariaDB als Quelle verwenden, aktivieren Sie für das Lesereplikat Backups und stellen Sie sicher, dass der Parameter `log_slave_updates` auf TRUE gesetzt ist.

Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS

Wenn Sie eine MySQL-Datenbank als Quelle verwenden, sollten Sie Folgendes beachten:

- Die Erfassung von Datenänderungen (Change Data Capture, CDC) wird für Amazon RDS MySQL 5.5 oder frühere Versionen nicht unterstützt. Für Amazon RDS MySQL müssen Sie die Version 5.6, 5.7 oder 8.0 verwenden, um CDC zu aktivieren. CDC wird für selbstverwaltete MySQL-5.5-Quellen unterstützt.
- Für CDC werden `CREATE TABLE`, `ADD COLUMN` und `DROP COLUMN` zur Änderung des Spaltentyps und `renaming a column` unterstützt. `DROP TABLE`, `RENAME TABLE` und Änderungen an anderen Attributen wie beispielsweise Standard-Spaltenwert, Nullfähigkeit für Spalten, Zeichensatz usw. werden jedoch nicht unterstützt.
- Wenn Sie für partitionierte Tabellen auf der Quelle den Target-Tabellenvorbereitungsmodus auf Tabellen auf Ziel löschen setzen, AWS DMS wird eine einfache Tabelle ohne Partitionen auf dem MySQL-Ziel erstellt. Wenn Sie partitionierte Tabellen in eine partitionierte Tabelle auf dem Ziel migrieren möchten, erstellen Sie zunächst die partitionierten Tabellen in der MySQL-Zieldatenbank.
- Die Verwendung der Anweisung `ALTER TABLE table_name ADD COLUMN column_name` zum Hinzufügen von Spalten zum Anfang (FIRST) oder zur Mitte einer Tabelle (AFTER) wird nicht unterstützt. Spalten werden stets am Ende der Tabelle hinzugefügt.
- CDC wird nicht unterstützt, wenn ein Tabellename Groß- und Kleinbuchstaben enthält und die Quell-Engine auf einem Betriebssystem mit Dateinamen ohne Berücksichtigung von Groß-/ Kleinschreibung gehostet wird. Ein Beispiel dafür ist Microsoft Windows oder OS X mit HFS+.
- Sie können Aurora MySQL-Compatible Edition Serverless v1 für Volllast verwenden, aber Sie können sie nicht für CDC verwenden. Dies liegt daran, dass Sie die Voraussetzungen für MySQL nicht aktivieren können. Weitere Informationen finden Sie unter [Parameter groups and Aurora Serverless v1](#).

Aurora MySQL-Compatible Edition Serverless v2 unterstützt CDC.

- Das Attribut `AUTO_INCREMENT` auf einer Spalte wird nicht zu einer Spalte in der Zieldatenbank migriert.
- Das Erfassen von Änderungen, wenn die Binärprotokolle nicht auf Standardblockspeicher gespeichert sind, wird nicht unterstützt. Beispielsweise funktioniert CDC nicht, wenn die Binärprotokolle in Amazon S3 gespeichert werden.
- AWS DMS erstellt standardmäßig Zieltabellen mit der InnoDB-Speicher-Engine. Falls Sie eine andere Speicher-Engine als InnoDB benötigen, müssen Sie die Tabelle manuell erstellen und dann die Daten mit dem Modus [do nothing](#) migrieren.
- Sie können Aurora MySQL-Repliken nicht als Quelle verwenden, AWS DMS es sei denn, Ihr DMS-Migrationsaufgabenmodus lautet Vorhandene Daten migrieren — nur Volllast.
- Im Falle, dass die MySQL-kompatible Quelle während des vollständigen Ladevorgangs gestoppt wird, bricht die AWS DMS -Aufgabe nicht mit einem Fehler ab. Die Aufgabe wird erfolgreich abgeschlossen, aber das Ziel wurde möglicherweise nicht mit der Quelle synchronisiert. Wenn dies geschieht, starten Sie entweder die Aufgabe neu oder Sie laden die betroffenen Tabellen neu.
- Indizes, die auf einem Teil des Spaltenwerts erstellt wurden, werden nicht migriert. Beispielsweise wird der Index `CREATE INDEX first_ten_chars ON customer (name(10))` nicht auf dem Ziel erstellt.
- In einigen Fällen ist die Aufgabe so konfiguriert, dass LOBs nicht repliziert werden ("SupportLobs" ist in den Aufgabeneinstellungen falsch oder LOB-Spalten nicht einbeziehen ist in der Task-Konsole ausgewählt). In diesen Fällen werden AWS DMS keine MEDIUMBLOB-, LONGBLOB-, MEDIUMTEXT- und LONGTEXT-Spalten zum Ziel migriert.

Spalten des Typs BLOB, TINYBLOB, TEXT und TINYTEXT sind davon nicht betroffen und werden zum Ziel migriert.

- Temporäre Datentabellen oder systemversionierte Tabellen werden in MariaDB-Quell- und Zieldatenbanken nicht unterstützt.
- Bei der Migration zwischen zwei Clustern von Amazon RDS Aurora MySQL muss es sich beim Quellendpunkt von RDS Aurora MySQL um eine Lese-/Schreib-Instance handeln, nicht um eine Replikat-Instance.
- AWS DMS unterstützt derzeit keine Migration von Ansichten für MariaDB.
- AWS DMS unterstützt keine DDL-Änderungen für partitionierte Tabellen für MySQL. Setzen Sie `skipTableSuspensionForPartitionDdl` auf `true`, um das Sperren von Tabellen bei DDL-Änderungen an Partitionen während CDC zu überspringen.
- AWS DMS unterstützt nur XA-Transaktionen in Version 3.5.0 und höher. Frühere Versionen unterstützen keine XA-Transaktionen. AWS DMS unterstützt keine XA-Transaktionen in MariaDB

Version 10.6. Weitere Informationen finden Sie unter [the section called “Unterstützung für XA-Transaktionen”](#).

- AWS DMS verwendet keine GTIDs für die Replikation, auch wenn die Quelldaten sie enthalten.
- AWS DMS unterstützt keine Komprimierung von binären Protokolltransaktionen.
- AWS DMS verbreitet die Ereignisse ON DELETE CASCADE und ON UPDATE CASCADE nicht für MySQL-Datenbanken, die die InnoDB-Speicher-Engine verwenden. Für diese Ereignisse generiert MySQL keine Binlog-Ereignisse, um die kaskadierten Operationen in den untergeordneten Tabellen widerzuspiegeln. Folglich AWS DMS können die entsprechenden Änderungen nicht in die untergeordneten Tabellen repliziert werden. Weitere Informationen finden Sie unter [Indizes, Fremdschlüssel oder kaskadierende Aktualisierungen oder Löschungen wurden nicht migriert](#).
- AWS DMS erfasst keine Änderungen an berechneten (VIRTUAL und GENERATED ALWAYS) Spalten. Gehen Sie wie folgt vor, um diese Einschränkung zu umgehen:
 - Erstellen Sie die Zieltabelle vorab in der Zieldatenbank und erstellen Sie die AWS DMS - Aufgabe mit der Einstellung DO_NOTHING oder TRUNCATE_BEFORE_LOAD für Aufgaben mit vollständigem Laden.
 - Fügen Sie eine Transformationsregel hinzu, um die berechnete Spalte aus dem Aufgabenbereich zu entfernen. Informationen zu Transformationsregeln finden Sie unter [Transformationsregeln und Aktionen](#).

Unterstützung für XA-Transaktionen

Eine Extended Architecture (XA)-Transaktion ist eine Transaktion, die verwendet werden kann, um eine Reihe von Operationen aus mehreren Transaktionsressourcen in einer einzigen, zuverlässigen globalen Transaktion zusammenzufassen. Eine XA-Transaktion verwendet ein zweiphasiges Commit-Protokoll. Im Allgemeinen kann die Erfassung von Änderungen während offener XA-Transaktionen zu Datenverlust führen. Wenn Ihre Datenbank keine XA-Transaktionen verwendet, können Sie diese Berechtigung und die Konfiguration IgnoreOpenXaTransactionsCheck ignorieren, indem Sie den Standardwert TRUE verwenden. Gehen Sie wie folgt vor, um mit der Replikation aus einer Quelle zu beginnen, die XA-Transaktionen enthält:

- Stellen Sie sicher, dass der AWS DMS Endpunktbenutzer über die folgenden Berechtigungen verfügt:

```
grant XA_RECOVER_ADMIN on *.* to 'userName'@'%';
```

- Setzen Sie die Endpunkteinstellung IgnoreOpenXaTransactionsCheck auf false.

Note

AWS DMS unterstützt keine XA-Transaktionen auf MariaDB Source DB Version 10.6.

Endpunkteinstellungen bei Verwendung von MySQL als Quelle für AWS DMS

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer MySQL-Quelldatenbank verwenden. Sie geben die Einstellungen an, wenn Sie den Quellendpunkt mithilfe der AWS DMS Konsole oder mithilfe des `create-endpoint` Befehls in [AWS CLI](#), mit der `--my-sql-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax erstellen.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit MySQL als Quelle verwenden können.

Name	Beschreibung
<code>EventsPollInterval</code>	<p>Gibt an, wie oft das binäre Protokolle für neue Änderungen/Ereignisse überprüft wird, wenn die Datenbank inaktiv ist.</p> <p>Standardwert: 5</p> <p>Zulässige Werte: 1 bis 60</p> <p>Beispiel: <code>--my-sql-settings '{"EventsPollInterval": 5}'</code></p> <p>In diesem Beispiel wird AWS DMS alle fünf Sekunden nach Änderungen in den Binärprotokollen gesucht.</p>
<code>ExecuteTimeout</code>	<p>Legt für AWS DMS Versionen 3.4.7 und höher den Timeout der Client-Anweisung für einen MySQL-Quellendpunkt in Sekunden fest.</p> <p>Standardwert: 60</p>

Name	Beschreibung
	Beispiel: <code>--my-sql-settings '{"ExecuteTimeout": 1500}'</code>
ServerTimezone	<p>Gibt die Zeitzone für die MySQL-Quelldatenbank an.</p> <p>Beispiel: <code>--my-sql-settings '{"ServerTimezone": " <i>US/Pacific</i> "'}</code></p>
AfterConnectScript	<p>Gibt ein Skript an, das unmittelbar nach der AWS DMS Verbindung mit dem Endpunkt ausgeführt werden soll. Die Migrationsaufgabe wird weiter ausgeführt, unabhängig davon, ob die SQL-Anweisung erfolgreich ist oder fehlschlägt.</p> <p>Zulässige Werte: Eine oder mehrere gültige SQL-Anweisungen, getrennt durch ein Semikolon.</p> <p>Beispiel: <code>--my-sql-settings '{"AfterConnectScript": "ALTER SESSION SET CURRENT_SCHEMA=system"}'</code></p>
CleanSrcMetadataOnMismatch	<p>Bereinigt und erstellt Tabellenmetadaten-Informationen auf der Replikations-Instance, wenn eine fehlende Übereinstimmung auftritt. Zum Beispiel in einer Situation , in der das Ausführen einer alter DDL auf der Tabelle zu unterschiedlichen Informationen über die in der Replikations-Instance zwischengespeicherte Tabelle führen könnte. Boolesch.</p> <p>Standardwert: <code>false</code></p> <p>Beispiel: <code>--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'</code></p>

Name	Beschreibung
<code>skipTableSuspensionForPartitionDdl</code>	<p>AWS DMS unterstützt keine DDL-Änderungen für partitionierte Tabellen für MySQL. Bei den AWS DMS Versionen 3.4.6 und höher wird durch die Einstellung dieser Einstellung die Tabellensperre für Partitions-DDL-Änderungen während <code>CDC true</code> übersprungen. AWS DMS ignoriert <code>partitioned-table-related DDL</code> und setzt die Verarbeitung weiterer Änderungen im Binärprotokoll fort.</p> <p>Standardwert: <code>false</code></p> <p>Beispiel: <code>--my-sql-settings '{"skipTableSuspensionForPartitionDdl": true}'</code></p>
<code>IgnoreOpenXaTransactionsCheck</code>	<p>Gibt für AWS DMS Versionen 3.5.0 und höher an, ob Aufgaben offene XA-Transaktionen beim Start ignorieren sollen. Setzen Sie dies auf <code>false</code>, wenn Ihre Quelle XA-Transaktionen enthält.</p> <p>Standardwert: <code>true</code></p> <p>Beispiel: <code>--my-sql-settings '{"IgnoreOpenXaTransactionsCheck": false}'</code></p>

Quelldatentypen für MySQL

Die folgende Tabelle zeigt die Quelldatentypen der MySQL-Datenbank, die bei der Verwendung unterstützt werden, AWS DMS sowie die Standardzuweisung von AWS DMS Datentypen.

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Hinweise zu AWS DMS Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

MySQL-Datentypen	AWS DMS Datentypen
INT	INT4
BIGINT	INT8
MEDIUMINT	INT4
TINYINT	INT1
SMALLINT	INT2
UNSIGNED TINYINT	UINT1
UNSIGNED SMALLINT	UINT2
UNSIGNED MEDIUMINT	UINT4
UNSIGNED INT	UINT4
UNSIGNED BIGINT	UINT8
DECIMAL(10)	NUMERIC (10,0)
BINARY	BYTES(1)
BIT	BOOLEAN
BIT(64)	BYTES(8)
BLOB	BYTES(65535)
LOB	BLOB
MEDIUMBLOB	BLOB
TINYBLOB	BYTES(255)
DATUM	DATUM
DATETIME	DATETIME

MySQL-Datentypen	AWS DMS Datentypen
	<p>DATETIME ohne einen Wert in Klammern wird ohne Millisekunden repliziert. DATETIME mit einem Wert zwischen 1 und 5 in Klammern (z. B. DATETIME(5)) wird mit Millisekunden repliziert.</p> <p>Bei der Replikation einer DATETIME-Spalte bleibt die Uhrzeit im Ziel gleich. Sie wird nicht in UTC konvertiert.</p>
TIME	STRING
TIMESTAMP (ZEITSTEMPEL)	<p>DATETIME</p> <p>Bei der Replikation einer TIMESTAMP-Spalte wird die Uhrzeit im Ziel in UTC konvertiert.</p>
JAHR	INT2
DOUBLE	REAL8
FLOAT	<p>REAL(DOUBLE)</p> <p>Wenn die FLOAT-Werte nicht im folgenden Bereich liegen, verwenden Sie eine Transformation, um FLOAT auf STRING abzubilden. Weitere Informationen zu Umwandlungen finden Sie unter Transformationsregeln und Aktionen.</p> <p>Der unterstützte FLOAT-Bereich ist -1,79E+308 bis -2,23E-308, 0 und 2,23E-308 bis 1,79E+308</p> <p>.</p>
VARCHAR (45)	WSTRING (45)
VARCHAR (2000)	WSTRING (2000)

MySQL-Datentypen	AWS DMS Datentypen
VARCHAR (4000)	WSTRING (4000)
VARBINARY (4000)	BYTES (4000)
VARBINARY (2000)	BYTES (2000)
CHAR	WSTRING
TEXT	WSTRING
LONGTEXT	NCLOB
MEDIUMTEXT	NCLOB
TINYTEXT	WSTRING(255)
GEOMETRY	BLOB
POINT	BLOB
LINESTRING	BLOB
POLYGON	BLOB
MULTIPOINT	BLOB
MULTILINESTRING	BLOB
MULTIPOLYGON	BLOB
GEOMETRYCOLLECTION	BLOB
ENUM	WSTRING (<i>Länge</i>) <i>Länge</i> bezieht sich hier auf die Länge des längsten Werts in ENUM.

MySQL-Datentypen	AWS DMS Datentypen
SET	WSTRING (<i>Länge</i>) <i>Länge</i> bezieht sich hier auf die Gesamtlänge aller Werte in SET, einschließlich Kommas.
JSON	CLOB

Note

In einigen Fällen können Sie die Datentypen DATETIME und TIMESTAMP mit einem „Null“-Wert (d. h. 0000-00-00) angeben. Stellen Sie in diesem Fall sicher, dass die Zieldatenbank in der Replikationsaufgabe „Null“-Werte für die Datentypen DATETIME und TIMESTAMP unterstützt. Andernfalls werden diese Werte als "null" im Ziel erfasst.

Verwenden einer SAP ASE-Datenbank als Quelle für AWS DMS

Sie können Daten aus einer Datenbank in SAP Adaptive Server Enterprise (ASE) – ehemals Sybase – mithilfe von AWS DMS migrieren. Mit einer SAP-ASE-Datenbank als Quelle können Sie Daten zu allen anderen unterstützten AWS DMS-Zieldatenbanken migrieren.

Informationen zu den Versionen von SAP ASE, die AWS DMS als Quelle unterstützt, finden Sie unter [Quellen für AWS DMS](#).

Weitere Informationen zum Arbeiten mit SAP ASE-Datenbanken und AWS DMS finden Sie in den folgenden Abschnitten.

Themen

- [Voraussetzungen für die Verwendung einer SAP ASE-Datenbank als Quelle für AWS DMS](#)
- [Einschränkungen bei der Verwendung von SAP ASE als Quelle für AWS DMS](#)
- [Erforderliche Berechtigungen für die Verwendung von SAP ASE als Quelle für AWS DMS](#)
- [Entfernen des Kürzungspunkts](#)
- [Endpunkteinstellungen bei Verwendung von SAP ASE als Quelle für AWS DMS](#)
- [Quelldatentypen für SAP ASE](#)

Voraussetzungen für die Verwendung einer SAP ASE-Datenbank als Quelle für AWS DMS

Um eine SAP ASE-Datenbank als Quelle für AWS DMS zu verwenden, gehen Sie wie folgt vor:

- Aktivieren Sie die SAP ASE-Replikation für Tabellen mit dem Befehl `sp_setreptable`. Weitere Informationen finden Sie unter [Sybase-Infocenter-Archiv](#).
- Deaktivieren Sie RepAgent auf der SAP ASE-Datenbank. Weitere Informationen finden Sie unter [Beenden und Deaktivieren des RepAgent-Threads in der Primärdatenbank](#).
- Um auf einer für nicht-lateinische Zeichen (z. B. Chinesisch) konfigurierten Windows-EC2-Instance auf SAP ASE Version 15.7 zu replizieren, installieren Sie SAP ASE 15.7 SP121 auf dem Zielrechner.

Note

Für die laufende CDC-Replikation zur Erfassung von Datenänderungen (Change Data Capture, CDC) führt DMS `dbcc logtransfer` und `dbcc log aus`, um Daten aus dem Transaktionsprotokoll zu lesen.

Einschränkungen bei der Verwendung von SAP ASE als Quelle für AWS DMS

Die folgenden Einschränkungen gelten, wenn Sie eine SAP ASE-Datenbank als Quelle für AWS DMS verwenden:

- Sie können für jede SAP-ASE-Datenbank nur eine AWS DMS-Aufgabe mit laufender Replikation oder CDC ausführen. Sie können mehrere reine Volllastaufgaben parallel ausführen.
- Es ist nicht möglich, eine Tabelle umzubenennen. Beispielsweise schlägt der folgende Befehl fehl:

```
sp_rename 'Sales.SalesRegion', 'SalesReg;
```

- Es ist nicht möglich, eine Spalte umzubenennen. Beispielsweise schlägt der folgende Befehl fehl:

```
sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';
```

- Nullwerte am Ende von Zeichenfolgen binären Datentyps werden gekürzt, wenn sie in der Zieldatenbank repliziert werden. Beispielsweise wird

0x0010000001000000000 in der Quelltable zu
0x0010000001 in der Zieltabelle.

- Wenn die Datenbank standardmäßig keine NULL-Werte zulässt, erstellt AWS DMS die Zieltabelle mit Spalten, die keine NULL-Werte zulassen. Wenn also eine Volllast- oder CDC-Replikationsaufgabe leere Werte enthält, löst AWS DMS einen Fehler aus. Sie können diese Fehler verhindern, indem Sie mithilfe der folgenden Befehle NULL-Werte in der Quelldatenbank zulassen.

```
sp_dboption database_name, 'allow nulls by default', 'true'
go
use database_name
CHECKPOINT
go
```

- Der Indexbefehl `reorg rebuild` wird nicht unterstützt.
- AWS DMS unterstützt weder Cluster noch die Verwendung von MSA (Multi-Site Availability)/Warm Standby als Quelle.
- Wenn der Transformations-Header-Ausdruck `AR_H_TIMESTAMP` in Zuordnungsregeln verwendet wird, werden die Millisekunden für eine hinzugefügte Spalte nicht erfasst.
- Das Ausführen von Zusammenführungsoperationen während CDC resultiert in einem Fehler, der nicht behoben werden kann. Führen Sie einen Volllastvorgang aus, um das Ziel wieder zu synchronisieren.
- Rollback-Auslöser-Ereignisse werden für Tabellen, die ein Sperrschema für Datenzeilen verwenden, nicht unterstützt.
- AWS DMS kann eine Replikationsaufgabe nicht fortsetzen, nachdem eine Tabelle innerhalb des Aufgabenbereichs aus einer SAP-Quelldatenbank gelöscht wurde. Wenn die DMS-Replikationsaufgabe angehalten und eine DML-Operation (INSERT, UPDATE, DELETE) ausgeführt und anschließend die Tabelle gelöscht wurde, müssen Sie die Replikationsaufgabe neu starten.

Erforderliche Berechtigungen für die Verwendung von SAP ASE als Quelle für AWS DMS

Um eine SAP-ASE-Datenbank als Quelle in einer AWS DMS-Aufgabe verwenden zu können, müssen Sie Berechtigungen erteilen. Erteilen Sie dem in den AWS DMS-Datenbankdefinitionen angegebenen Benutzerkonto die folgenden Berechtigungen in der SAP-ASE-Datenbank:

- `sa_role`

- replication_role
- sybase_ts_role
- Wenn Sie über die Berechtigung zum Ausführen der gespeicherten Prozedur `sp_setreptable` verfügen müssen, aktiviert AWS DMS standardmäßig die SAP-ASE-Replikationsoption. Wenn Sie `sp_setreptable` direkt vom Datenbankendpunkt aus und nicht über AWS DMS selbst für eine Tabelle ausführen möchten, können Sie das zusätzliche Verbindungsattribut `enableReplication` verwenden. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von SAP ASE als Quelle für AWS DMS](#).

Entfernen des Kürzungspunkts

Wenn eine Aufgabe gestartet wird, setzt AWS DMS einen `$replication_truncation_point`-Eintrag in die `syslogshold`-Systemansicht. Damit wird angezeigt, dass ein Replikationsprozess läuft. Während AWS DMS arbeitet, wird der Replikationskürzungspunkt in regelmäßigen Abständen weiter gesetzt, entsprechend der Menge der Daten, die bereits in das Ziel kopiert wurde.

Nachdem der Eintrag `$replication_truncation_point` eingerichtet wurde, muss die AWS DMS-Aufgabe ununterbrochen ausgeführt werden, um zu verhindern, dass das Datenbankprotokoll übermäßig groß wird. Wenn Sie die AWS DMS-Aufgabe dauerhaft stoppen möchten, entfernen Sie den Replikationskürzungspunkt durch Ausgabe des folgenden Befehls:

```
dbcc settrunc('ltm','ignore')
```

Nachdem der Kürzungspunkt entfernt wurde, kann die AWS DMS-Aufgabe nicht wieder aufgenommen werden. Das Protokoll wird weiterhin automatisch an den Checkpoints gekürzt (falls automatische Kürzung eingestellt ist).

Endpunkteinstellungen bei Verwendung von SAP ASE als Quelle für AWS DMS

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer SAP-ASE-Quelldatenbank verwenden. Sie legen die Einstellungen fest, wenn Sie den Quellendpunkt mithilfe der AWS DMS-Konsole erstellen oder indem Sie den Befehl `create-endpoint` in der [AWS CLI](#) aufrufen. Verwenden Sie die JSON-Syntax `--sybase-settings '{"EndpointSetting": "value", ...}'`.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit SAP ASE als Quelle verwenden können.

Name	Beschreibung
Charset	<p data-bbox="686 254 1487 331">Legen Sie dieses Attribut auf den SAP-ASE-Namen fest, der dem internationalen Zeichensatz entspricht.</p> <p data-bbox="686 380 992 415">Standardwert: <code>iso_1</code></p> <p data-bbox="686 464 1386 541">Beispiel: <code>--sybase-settings '{"Charset": "utf8"}'</code></p> <p data-bbox="686 590 927 625">Zulässige Werte:</p> <ul data-bbox="686 674 854 1848" 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>• <code>cp1250</code>• <code>cp1251</code>• <code>cp1252</code>• <code>cp1253</code>

Name	Beschreibung
	<ul style="list-style-type: none">• 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• tis620• turkish8• utf8

Name	Beschreibung
	Schlagen Sie bei weiteren Fragen zu den unterstützten Zeichensätzen in einer SAP-ASE-Datenbank unter Adaptive server supported character sets nach.
EnableReplication	<p>Legen Sie dieses Attribut fest, wenn Sie <code>sp_setrep table</code> vom Datenbankendpunkt aus und nicht über AWS DMS aktivieren möchten.</p> <p>Standardwert: <code>true</code></p> <p>Gültige Werte: <code>true</code> oder <code>false</code>.</p> <p>Beispiel: <code>--sybase-settings '{"Enable Replication": false}'</code></p>
EncryptPassword	<p>Legen Sie dieses Attribut fest, wenn Sie <code>"net password encryption reqd"</code> in der Quelldatenbank aktiviert haben.</p> <p>Standardwert: <code>0</code></p> <p>Gültige Werte: <code>0</code>, <code>1</code> oder <code>2</code></p> <p>Beispiel: <code>--sybase-settings '{"EncryptPassword": 1}'</code></p> <p>Weitere Informationen zu diesen Parameterwerten finden Sie unter Adaptive Server Enterprise: Using the EncryptPassword Connection string property.</p>

Name	Beschreibung
Provider	<p>Legen Sie dieses Attribut fest, wenn Sie Transport Layer Security (TLS) 1.2 für Versionen von ASE 15.7 und höher verwenden möchten. Beachten Sie, dass AWS TLS Version 1.2 oder höher erfordert und Version 1.3 empfohlen wird.</p> <p>Standardwert: Adaptive Server Enterprise</p> <p>Zulässige Werte: Adaptive Server Enterprise 16.03.06</p> <p>Beispiel: <code>--sybase-settings '{"Provider": "Adaptive Server Enterprise 16.03.06"}'</code></p>

Quelldatentypen für SAP ASE

Eine Liste der SAP ASE-Quelldatentypen, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung von AWS DMS-Datentypen finden Sie in der folgenden Tabelle. AWS DMS unterstützt keine SAP ASE-Quellentabellen mit Spalten des benutzerdefinierten Typs (UDT). Replizierte Spalten mit diesem Datentyp werden als NULL erstellt.

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt [Ziele für die Datenmigration](#) für Ihren Zielendpunkt.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

SAP ASE-Datentypen	AWS DMS-Datentypen
BIGINT	INT8
UNSIGNED BIGINT	UINT8
INT	INT4
UNSIGNED INT	UINT4

SAP ASE-Datentypen	AWS DMS-Datentypen
SMALLINT	INT2
UNSIGNED SMALLINT	UINT2
TINYINT	UINT1
DECIMAL	NUMERIC
NUMERIC	NUMERIC
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
MONEY	NUMERIC
SMALLMONEY	NUMERIC
DATETIME	DATETIME
BIGDATETIME	DATETIME(6)
SMALLDATETIME	DATETIME
DATUM	DATUM
TIME	TIME
BIGTIME	TIME
CHAR	STRING
UNICHAR	WSTRING
NCHAR	WSTRING
VARCHAR	STRING

SAP ASE-Datentypen	AWS DMS-Datentypen
UNIVARCHAR	WSTRING
NVARCHAR	WSTRING
BINARY	BYTES
VARBINARY	BYTES
BIT	BOOLEAN
TEXT	CLOB
UNITEXT	NCLOB
IMAGE	BLOB

Verwenden von MongoDB als Quelle für AWS DMS

Informationen zu den Versionen von MongoDB, die AWS DMS als Quelle unterstützt, finden Sie unter [Quellen für AWS DMS](#).

Beachten Sie Folgendes im Zusammenhang mit der Unterstützung der MongoDB-Versionen:

- Die AWS DMS-Versionen 3.4.5 und höher unterstützen die MongoDB-Versionen 4.2 und 4.4.
- Die AWS DMS-Versionen 3.4.5 und höher sowie die MongoDB-Versionen 4.2 und höher unterstützen verteilte Transaktionen. Weitere Informationen zu verteilten MongoDB-Transaktionen finden Sie unter [Transactions](#) in der [Dokumentation zu MongoDB](#).
- Die AWS DMS-Versionen 3.5.0 und höher unterstützen keine MongoDB-Versionen vor 3.6.
- Die AWS DMS-Versionen 3.5.1 und höher unterstützen die MongoDB-Version 5.0.
- Die AWS DMS-Versionen 3.5.2 und höher unterstützen die MongoDB-Version 6.0.

Falls Sie noch keine Erfahrung mit MongoDB haben, beachten Sie die folgenden wichtigen MongoDB-Datenbankkonzepte:

- Ein Datensatz in MongoDB ist ein Dokument, bei dem es sich um eine Datenstruktur aus Feld-Wert-Paaren handelt. Der Wert eines Felds kann andere Dokumente, Arrays und Dokument-Arrays enthalten. Ein Dokument entspricht etwa einer Zeile in einer relationalen Datenbanktabelle.
- Eine Sammlung ist in MongoDB eine Gruppe von Dokumente und entspricht in etwa einer Tabelle in einer relationalen Datenbank.
- Eine Datenbank in MongoDB ist ein Satz von Sammlungen und entspricht in etwa einem Schema in einer relationalen Datenbank.
- Intern wird ein MongoDB-Dokument als binäre JSON-Datei (BSON) in einem komprimierten Format gespeichert, das einen Typ für jedes Feld im Dokument enthält. Jedes Dokument hat eine eindeutige ID.

AWS DMS unterstützt zwei Migrationsmodi, wenn MongoDB als Quelle verwendet wird: den Dokumentmodus und den Tabellenmodus. Sie geben den zu verwendenden Migrationsmodus während der Erstellung des MongoDB-Endpunkts oder durch Festlegen des Parameters Metadatenmodus in der AWS DMS-Konsole an. Optional können Sie eine zweite Spalte mit der Bezeichnung `_id` erstellen, die als Primärschlüssel fungiert, indem Sie im Endpunktkonfigurationsbereich das Häkchen für `_id` als separate Spalte setzen.

Der ausgewählte Migrationsmodus wirkt sich wie nachstehend erläutert auf das resultierende Format der Zieldaten aus.

Dokumentmodus

Im Dokumentmodus wird das MongoDB-Dokument unverändert migriert. Dies bedeutet, dass die Dokumentdaten in einer einzelnen Spalte mit dem Namen `_doc` in einer Zieltabelle zusammengefasst werden. Der Dokumentmodus ist die Standardeinstellung, wenn Sie MongoDB als Quellendpunkt verwenden.

Nehmen wir als Beispiel die folgenden Dokumente in einer MongoDB-Sammlung mit dem Namen "myCollection".

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Nach der Migration der Daten in eine relationale Datenbanktabelle mit Dokumentmodus sind die Daten folgendermaßen strukturiert. Die Datenfelder im MongoDB-Dokument sind in der Spalte `_doc` zusammengefasst.

oid_id	_doc
5a94815f40bd44d1b02bdfe0	{ "a" : 1, "b" : 2, "c" : 3 }
5a94815f40bd44d1b02bdfe1	{ "a" : 4, "b" : 5, "c" : 6 }

Sie können optional den `extractDocID` des zusätzlichen Verbindungsattributs auf `true` einstellen, um eine zweite Spalte mit dem Namen `"_id"` zu erstellen, die als Primärschlüssel fungiert. Sie müssen diesen Parameter auf `true` setzen, wenn Sie CDC verwenden möchten.

Nachfolgend ist dargestellt, wie AWS DMS die Erstellung und Umbenennung von Sammlungen im Dokumentmodus verwaltet:

- Wenn Sie eine neue Sammlung zur Quelldatenbank hinzufügen, erstellt AWS DMS eine neue Zieltabelle für die Sammlung und repliziert alle Dokumente.
- Wenn Sie eine vorhandene Sammlung in der Quelldatenbank umbenennen, benennt AWS DMS die Zieltabelle nicht um.

Wenn der Zielendpunkt Amazon DocumentDB ist, führen Sie die Migration im Dokumentmodus aus.

Tabellenmodus

Im Tabellenmodus wandelt AWS DMS jedes Top-Level-Feld in einem MongoDB-Dokument in eine Spalte in der Zieltabelle um. Wenn ein Feld verschachtelt ist, reduziert AWS DMS die verschachtelten Werte auf eine einzelne Spalte. AWS DMS fügt der Spaltengruppe in der Zieltabelle dann ein Schlüsselfeld und Datentypen hinzu.

Für jedes MongoDB-Dokument fügt AWS DMS jeden Schlüssel und Typ der Spaltengruppe in der Tabelle der Zieldatenbank hinzu. Im Tabellenmodus migriert AWS DMS beispielsweise das vorherige Beispiel in die folgende Tabelle.

oid_id	a	b	c
5a94815f40bd44d1b02bdfe0	1	2	3
5a94815f40bd44d1b02bdfe1	4	5	6

Verschachtelte Werte werden in einer Spalte mit durch Punkte getrennte Schlüsselnamen auf eine Ebene gebracht. Die Spalte erhält als Name die Namensverkettung der reduzierten Felder, die durch Punkte voneinander getrennt sind. Beispiel: AWS DMS migriert ein JSON-Dokument mit einem Feld verschachtelter Werte wie beispielsweise {"a" : {"b" : {"c": 1}}}} in eine Spalte mit dem Namen a.b.c..

Zum Erstellen der Zielspalten scannt AWS DMS eine angegebene Anzahl von MongoDB-Dokumenten und erstellt eine Gruppe aller Felder und deren Typen. AWS DMS verwendet diese Gruppe dann, um die Spalten der Zieltabelle zu erstellen. Wenn Sie Ihren MongoDB-Quellendpunkt mithilfe der Konsole erstellen oder ändern, können Sie die Anzahl der zu scannenden Dokumente angeben. Der Standardwert ist 1000 Dokumente. Wenn Sie die AWS CLI verwenden, können Sie das zusätzliche Verbindungsattribut docsToInvestigate verwenden.

Hier sehen Sie, wie AWS DMS Dokumente und Sammlungen im Tabellenmodus verwaltet:

- Wenn Sie ein Dokument zu einer vorhandenen Sammlung hinzufügen, wird das Dokument repliziert. Wenn es Felder gibt, die am Ziel nicht vorhanden sind, werden diese Felder nicht repliziert.
- Wenn Sie ein Dokument aktualisieren, wird das aktualisierte Dokument repliziert. Wenn es Felder gibt, die am Ziel nicht vorhanden sind, werden diese Felder nicht repliziert.
- Das Löschen eines Dokuments wird vollständig unterstützt.
- Das Hinzufügen einer neuen Sammlung führt nicht zu einer neuen Tabelle in der Zieldatenbank, wenn der Vorgang zeitgleich mit einer CDC-Aufgabe erfolgt.
- In der Phase der Erfassung von Datenänderungen (Change Data Capture, CDC) unterstützt AWS DMS das Umbenennen einer Sammlung nicht.

Themen

- [Erforderliche Berechtigungen für die Verwendung von MongoDB als Quelle für AWS DMS](#)
- [Konfigurieren eines MongoDB-Replikatsatzes für CDC](#)
- [Sicherheitsanforderungen für die Verwendung von MongoDB als Quelle für AWS DMS](#)
- [Segmentieren von MongoDB-Sammlungen und parallele Migration](#)
- [Migration mehrerer Datenbanken bei Verwendung von MongoDB als Quelle für AWS DMS](#)
- [Einschränkungen bei der Verwendung von MongoDB als Quelle für AWS DMS](#)
- [Einstellungen für die Endpunktconfiguration bei Verwendung von MongoDB als Quelle für AWS DMS](#)

- [Quelldatentypen für MongoDB](#)

Erforderliche Berechtigungen für die Verwendung von MongoDB als Quelle für AWS DMS

Bei einer AWS DMS-Migration mit einer MongoDB-Quelle können Sie entweder ein Benutzerkonto mit Stammverzeichnisberechtigungen oder einen Benutzer nur mit Berechtigungen für die zu migrierende Datenbank erstellen.

Der folgende Code erstellt einen Benutzer als Stammkonto.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  }
)
```

Für eine MongoDB-3.x-Quelle wird mit dem folgenden Code ein Benutzer mit minimalen Berechtigungen für die zu migrierende Datenbank erstellt.

```
use database_to_migrate
db.createUser(
  {
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "local" }, "read" ]
  })
```

Für eine MongoDB-4.x-Quelle wird mit dem folgenden Code ein Benutzer mit minimalen Berechtigungen erstellt.

```
{ resource: { db: "", collection: "" }, actions: [ "find", "changeStream" ] }
```

Erstellen Sie beispielsweise die folgende Rolle in der Datenbank „admin“.

```
use admin
```

```
db.createRole(  
{  
  role: "changestreamrole",  
  privileges: [  
    { resource: { db: "", collection: "" }, actions: [ "find","changeStream" ] }  
  ],  
  roles: []  
}  
)
```

Nachdem die Rolle erstellt wurde, erstellen Sie einen Benutzer in der zu migrierenden Datenbank.

```
> use test  
> db.createUser(  
{  
  user: "dms-user12345",  
  pwd: "password",  
  roles: [ { role: "changestreamrole", db: "admin" }, "read"]  
})
```

Konfigurieren eines MongoDB-Replikatsatzes für CDC

Um die laufende Replikation oder CDC mit MongoDB verwenden zu können, benötigt AWS DMS Zugriff auf das Protokoll der MongoDB-Operationen (Oplog). Zum Erstellen des Protokolls "oplog" müssen Sie einen Replikatsatz bereitstellen, sofern keiner vorhanden ist. Weitere Informationen finden Sie in der [MongoDB-Dokumentation](#).

Sie können CDC mit dem primären oder sekundären Knoten eines MongoDB-Replikatsatzes als Quellendpunkt verwenden.

So wandeln Sie eine eigenständige Instance in einen Replikatsatz um

1. Stellen Sie über die Befehlszeile eine Verbindung mit mongo her.

```
mongo localhost
```

2. Stoppen Sie den Service mongod.

```
service mongod stop
```

3. Starten Sie mongod mit dem folgenden Befehl neu:

```
mongod --replSet "rs0" --auth -port port_number
```

4. Testen Sie die Verbindung zum Replikatsatz mithilfe der folgenden Befehle:

```
mongo -u root -p password --host rs0/localhost:port_number  
--authenticationDatabase "admin"
```

Wenn Sie eine Migration im Dokumentmodus durchführen möchten, wählen Sie beim Erstellen des MongoDB-Endpunkts die Option `_id as a separate column` aus. Bei Auswahl dieser Option wird eine zweite Spalte namens `_id` erstellt, die als Primärschlüssel fungiert. Diese zweite Spalte wird von AWS DMS zur Unterstützung von DML-Operationen (Data Manipulation Language) benötigt.

Note

AWS DMS verwendet das Protokoll der Operationen (Oplog), um Änderungen während der laufenden Replikation zu erfassen. Wenn MongoDB die Datensätze aus dem Oplog löscht, bevor sie von AWS DMS gelesen werden, schlagen Ihre Aufgaben fehl. Wir empfehlen, die Oplog-Größe so zu ändern, dass die Änderungen mindestens 24 Stunden lang beibehalten werden.

Sicherheitsanforderungen für die Verwendung von MongoDB als Quelle für AWS DMS

AWS unterstützt zwei Authentifizierungsmethoden für MongoDB. Die beiden Authentifizierungsmethoden werden verwendet, um das Passwort zu verschlüsseln, daher werden sie nur verwendet, wenn der Parameter `authType` auf `PASSWORD` festgelegt ist.

Die MongoDB-Authentifizierungsmethoden sind:

- MONGODB-CR – für die Abwärtskompatibilität
- SCRAM-SHA-1 – die Standardeinstellung bei Verwendung von MongoDB-Version 3.x und 4.0

Wenn keine Authentifizierungsmethode angegeben ist, verwendet AWS DMS die Standardmethode für die Version der MongoDB-Quelle.

Segmentieren von MongoDB-Sammlungen und parallele Migration

Um die Leistung einer Migrationsaufgabe zu verbessern, unterstützen MongoDB-Quellendpunkte zwei Optionen für paralleles vollständiges Laden bei der Tabellenzuordnung.

Anders ausgedrückt: Sie können eine Sammlung parallel migrieren, indem Sie entweder die automatische Segmentierung oder die Bereichssegmentierung mit der Tabellenzuordnung für paralleles vollständiges Laden in den JSON-Einstellungen verwenden. Mit der automatischen Segmentierung können Sie die Kriterien angeben, nach denen AWS DMS Ihre Migrationsquelle in jedem Thread automatisch segmentieren soll. Mit der Bereichssegmentierung können Sie AWS DMS den spezifischen Bereich jedes Segments mitteilen, den DMS in jedem Thread migrieren soll. Weitere Informationen zu diesen Einstellungen finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Parallele Migration einer MongoDB-Datenbank mithilfe von Bereichen für die automatische Segmentierung

Sie können Ihre Dokumente parallel migrieren, indem Sie die Kriterien angeben, nach denen AWS DMS Ihre Daten für jeden Thread automatisch partitionieren (segmentieren) soll. Insbesondere geben Sie die Anzahl an Dokumenten an, die pro Thread migriert werden sollen. Bei diesem Verfahren versucht AWS DMS, die Segmentgrenzen zu optimieren, um eine maximale Leistung pro Thread zu erzielen.

Sie können die Segmentierungskriterien mithilfe der folgenden Tabelleneinstellungsoptionen in der Tabellenzuordnung angeben.

Tabelleneinstellungsoption	Beschreibung
"type"	(Erforderlich) Setzen Sie dies für MongoDB als Quelle auf "partitions-auto" .
"number-of-partitions"	(Optional) Gesamtzahl der für die Migration verwendeten Partitionen (Segmente). Der Standardwert ist 16.
"collection-count-from-meta-data"	(Optional) Wenn diese Option auf true gesetzt ist, verwendet AWS DMS eine geschätzte Sammlungsanzahl, um die Anzahl von Partitionen zu bestimmen. Wenn die Option auf false

Tabelleneinstellungsoption	Beschreibung
	gesetzt ist, verwendet AWS DMS die tatsächliche Sammlungsanzahl. Der Standardwert ist <code>true</code> .
"max-records-skip-per-page"	<p>(Optional) Die Anzahl der Datensätze, die bei der Ermittlung der Grenzen für jede Partition gleichzeitig übersprungen werden sollen. AWS DMS verwendet einen Ansatz zum seitenweisen Überspringen, um die Mindestgrenze für eine Partition zu bestimmen. Der Standardwert ist 10 000.</p> <p>Das Festlegen eines relativ großen Werts kann zu Cursor-Timeouts und Aufgabenfehlern führen. Das Festlegen eines relativ niedrigen Werts führt zu mehr Operationen pro Seite und zu einem langsameren vollständigen Laden.</p>
"batch-size"	(Optional) Begrenzt die Anzahl der in einem Stapel zurückgegebenen Dokumente. Für jeden Stapel ist ein Round-Trip zum Server erforderlich. Wenn die Stapelgröße Null (0) ist, verwendet der Cursor die vom Server definierte maximale Stapelgröße. Der Standardwert ist 0.

Das folgende Beispiel zeigt eine Tabellenzuordnung für die automatische Segmentierung.

```
{
  "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
        }
    }
]
}
```

Für die automatische Segmentierung gelten die folgenden Einschränkungen. Bei der Migration für die einzelnen Segmente werden die Sammlungsanzahl und der `_id`-Mindestwert für die Sammlung separat abgerufen. Anschließend wird die Mindestgrenze für dieses Segment anhand eines seitenweisen Überspringens berechnet.

Stellen Sie daher sicher, dass der `_id`-Mindestwert für jede Sammlung konstant bleibt, bis alle Segmentgrenzen in der Sammlung berechnet wurden. Wenn Sie den `_id`-Mindestwert für eine Sammlung während der Berechnung der Segmentgrenze ändern, kann dies zu Datenverlust oder zu Fehlern aufgrund von doppelten Zeilen führen.

Parallele Migration einer MongoDB-Datenbank mithilfe der Bereichssegmentierung

Sie können Ihre Dokumente parallel migrieren, indem Sie die Bereiche für jedes Segment in einem Thread angeben. Bei diesem Verfahren teilen Sie AWS DMS in jedem Thread mit, welche spezifischen Dokumente migriert werden sollen, je nachdem, welche Dokumentbereiche Sie pro Thread auswählen.

Die folgende Abbildung zeigt eine MongoDB-Sammlung mit sieben Elementen und `_id` als Primärschlüssel.

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

Um die Sammlung in drei spezifische Segmente aufzuteilen, die AWS DMS parallel migrieren kann, können Sie Ihrer Migrationsaufgabe Regeln für die Tabellenzuordnung hinzufügen. Dieses Verfahren wird im folgenden JSON-Beispiel veranschaulicht.

```
{ // 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

```

Mit dieser Definition für die Tabellenzuordnung wird die Quellsammlung in drei Segmente aufgeteilt, die parallel migriert werden. Im Folgenden sind die Segmentierungsgrenzen aufgeführt.

```
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)
```

Nach Abschluss der Migrationsaufgabe können Sie anhand der Aufgabenprotokolle überprüfen, ob die Tabellen parallel geladen wurden, wie im folgenden Beispiel gezeigt. Sie können auch die zum Entladen der einzelnen Segmente aus der Quelltable verwendete MongoDB-Klausel `find` überprüfen.

```
[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.
```

AWS DMS unterstützt derzeit die folgenden MongoDB-Datentypen als Segmentschlüsselspalte:

- Double
- String
- ObjectId
- 32-Bit-Ganzzahl
- 64-Bit-Ganzzahl

Migration mehrerer Datenbanken bei Verwendung von MongoDB als Quelle für AWS DMS

Die AWS DMS-Versionen 3.4.5 und höher unterstützen die Migration von mehreren Datenbanken in einer einzigen Aufgabe für alle unterstützten MongoDB-Versionen. Gehen Sie wie folgt vor, um mehrere Datenbanken zu migrieren:

1. Wenn Sie den MongoDB-Quellendpunkt erstellen, führen Sie eine der folgenden Aktionen aus:
 - Stellen Sie sicher, dass das Feld Datenbankname unter Endpunktkonfiguration auf der Seite Endpunkt erstellen der DMS-Konsole leer ist.
 - Weisen Sie dem Parameter DatabaseName in MongoDBSettings mithilfe des AWS CLI-Befehls `CreateEndpoint` einen leeren Zeichenfolgenwert zu.
2. Geben Sie für jede Datenbank, die Sie aus einer MongoDB-Quelle migrieren möchten, den Datenbanknamen als Schemanamen in der Tabellenzuordnung für die Aufgabe an. Sie können dies entweder mithilfe der geführten Eingabe in der Konsole oder direkt in JSON vornehmen. Weitere Informationen zur geführten Eingabe finden Sie unter [Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole](#). Weitere Informationen zu JSON finden Sie unter [Auswahlregeln und Aktionen](#).

Sie können beispielsweise die folgende JSON-Struktur angeben, um drei MongoDB-Datenbanken zu migrieren.

Example Migrieren aller Tabellen in einem Schema

Die folgende JSON-Struktur migriert alle Tabellen aus den Datenbanken Customers, Orders und Suppliers im Quellendpunkt zum Zielendpunkt.

```
{
  "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": []
    }
  ]
}
```

```
]
}
```

Einschränkungen bei der Verwendung von MongoDB als Quelle für AWS DMS

Die folgenden Einschränkungen gelten bei der Verwendung von MongoDB als Quelle für AWS DMS:

- Im Tabellenmodus müssen die Dokumente in einer Sammlung hinsichtlich des Datentyps, den sie für den Wert im gleichen Feld verwenden, konsistent sein. Wenn ein Dokument in einer Sammlung beispielsweise `{ a: { b: value ... } }` enthält, müssen alle Dokumente in der Sammlung, die auf den `value` des Felds `a.b` verweisen, denselben Datentyp für `value` verwenden, wo immer er in der Sammlung vorkommt.
- Wenn die Option `_id` als separate Spalte gesetzt ist, darf der ID-String 200 Zeichen nicht überschreiten.
- Objekt-ID und Array-Schlüssel werden in Spalten konvertiert, die mit dem Präfix `oid` und `array` im Tabellenmodus versehen sind.

Intern wird auf diese Spalten mit den mit Präfix versehenen Namen verwiesen. Wenn Sie in AWS DMS Transformationsregeln verwenden, die auf diese Spalten verweisen, geben Sie unbedingt den mit Präfix versehenen Spaltennamen an. Sie geben z. B. `$_oid__id` statt `$_id` oder `$_array__addresses` und nicht `$_addresses` an.

- Namen von Sammlungen und Schlüsseln dürfen das Dollarzeichen (\$) nicht enthalten.
- Im Tabellenmodus mit einem RDBMS-Ziel unterstützt AWS DMS keine Sammlungen, die das gleiche Feld mit unterschiedlicher Schreibweise (Groß-/Kleinschreibung) enthalten. Beispielsweise unterstützt AWS DMS nicht zwei Sammlungen mit den Namen `Field1` und `field1`.
- Der Tabellenmodus und der Dokumentmodus haben die oben beschriebenen Einschränkungen.
- Für die parallele Migration mithilfe der automatischen Segmentierung gelten die oben beschriebenen Einschränkungen.
- Quellfilter werden für MongoDB nicht unterstützt.
- AWS DMS unterstützt keine Dokumente mit einer größeren Verschachtelungsebene als 97.
- AWS DMS unterstützt die folgenden Features der MongoDB-Version 5.0 nicht:
 - Live-Resharding
 - Clientseitige Verschlüsselung auf Feldebene (Client-Side Field Level Encryption, CSFLE)
 - Migration von Zeitreihen-Sammlungen

Note

Eine in der Phase des vollständigen Ladens migrierte Zeitreihen-Sammlung wird in Amazon DocumentDB in eine normale Sammlung konvertiert, da DocumentDB keine Zeitreihen-Sammlungen unterstützt.

Einstellungen für die Endpunktconfiguration bei Verwendung von MongoDB als Quelle für AWS DMS

Wenn Sie Ihren MongoDB-Quellendpunkt einrichten, können Sie mithilfe der AWS DMS-Konsole mehrere Einstellungen für die Endpunktconfiguration festlegen.

In der folgenden Tabelle werden die verfügbaren Konfigurationseinstellungen bei Verwendung von MongoDB-Datenbanken als AWS DMS-Quelle beschrieben.

Einstellung (Attribut)	Zulässige Werte	Standardwert und Beschreibung
Authentifizierungsmodus	"none" "password"	Der Wert "password" fordert zur Eingabe eines Benutzernamens und eines Passworts auf. Wenn "none" angegeben ist, werden die Parameter Benutzername und Passwort nicht verwendet.
Authentifizierungsquelle	Ein gültiger MongoDB-Datenbankname.	Der Name der MongoDB-Datenbank, die Sie verwenden möchten, um Ihre Anmeldeinformationen für die Authentifizierung zu überprüfen. Der Standardwert ist "admin".
Authentifizierungsmechanismus	"default" "mongodb_cr" "scram_sha_1"	Der Authentifizierungsmechanismus. Der Wert von "default" ist "scram_sha_1". Diese Einstellung wird nicht verwendet, wenn authType auf "no" festgelegt ist.
Metadatenmodus	Dokument und Tabelle	Zur Auswahl von Dokument- oder Tabellenmodus

Einstellung (Attribut)	Zulässige Werte	Standardwert und Beschreibung
Anzahl der zu scannenden Dokumente (docsToInvestigate)	Eine positive Ganzzahl größer als 0.	Verwenden Sie diese Option nur im Tabellenmodus, um die Zieltabellendefinition zu definieren.
_id als separate Spalte	Häkchen im Kästchen	Optionales Auswahlfeld, mit dem eine zweite Spalte mit der Bezeichnung _id erstellt wird, die als Primärschlüssel fungiert.
socketTimeoutMS	NUMBER Nur zusätzliches Verbindungsattribut (Extra Connection Attribute, ECA)	Diese Einstellung wird in Millisekunden angegeben und konfiguriert das Verbindungs-Timeout für MongoDB-Clients. Wenn der Wert kleiner oder gleich Null ist, wird der Standardwert des MongoDB-Clients verwendet.
UseUpdateLookUp	boolesch true false	Wenn Sie true festlegen, kopiert AWS DMS während CDC-Aktualisierungsereignissen das gesamte aktualisierte Dokument in das Ziel. Wenn Sie false festlegen, verwendet AWS DMS den MongoDB-Aktualisierungsbefehl, um nur geänderte Felder im Dokument im Ziel zu aktualisieren.

Einstellung (Attribut)	Zulässige Werte	Standardwert und Beschreibung
Replicate ShardCollections	boolesch true false	<p>Wenn Sie true festlegen, repliziert AWS DMS Daten in Shard-Sammlungen. AWS DMS verwendet diese Einstellung nur, wenn es sich beim Zielendpunkt um einen elastischen DocumentDB-Cluster handelt.</p> <p>Wenn Sie für diese Einstellung true festlegen, beachten Sie Folgendes:</p> <ul style="list-style-type: none"> • Sie müssen TargetTablePrepMode auf nothing setzen. • AWS DMS setzt useUpdateLookup automatisch auf false.

Wenn Sie Dokument als Metadatenmodus auswählen, sind verschiedene Optionen verfügbar.

Wenn der Zielendpunkt DocumentDB ist, führen Sie die Migration unbedingt im Dokumentmodus aus. Ändern Sie außerdem Ihren Quellendpunkt und wählen Sie die Option `_id` als separate Spalte aus. Dies ist eine zwingende Voraussetzung, wenn Ihr MongoDB-Quell-Workload Transaktionen beinhaltet.

Quelldatentypen für MongoDB

Eine Datenmigration, bei der MongoDB als Quelle für AWS DMS verwendet wird, unterstützt die meisten MongoDB-Datentypen. In der folgenden Tabelle sind die MongoDB-Quelldatentypen, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung von AWS DMS-Datentypen aufgeführt. Weitere Informationen zu MongoDB-Datentypen finden Sie unter [BSON-Typen](#) in der MongoDB-Dokumentation.

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

MongoDB-Datentypen	AWS DMS-Datentypen
Boolesch	Bool
Binär	BLOB
Datum	Datum
Zeitstempel	Datum
Int	INT4
Long	INT8
Double	REAL8
String (UTF-8)	CLOB
Array	CLOB
OID	String
REGEX	CLOB
CODE	CLOB

Verwenden von Amazon DocumentDB (mit MongoDB-Kompatibilität) als Quelle für AWS DMS

Informationen zu den Versionen von Amazon DocumentDB (mit MongoDB-Kompatibilität), die AWS DMS als Quelle unterstützt, finden Sie unter [Quellen für AWS DMS](#).

Wenn Sie Amazon DocumentDB als Quelle verwenden, können Sie Daten von einem Amazon-DocumentDB-Cluster zu einem anderen migrieren. Sie können auch Daten von einem Amazon-DocumentDB-Cluster zu einem der anderen von AWS DMS unterstützten Zielendpunkte migrieren.

Wenn Sie noch keine Erfahrung mit Amazon DocumentDB haben, sollten Sie die folgenden wichtigen Konzepte für Amazon-DocumentDB-Datenbanken beachten:

- Ein Datensatz in Amazon DocumentDB ist ein Dokument, bei dem es sich um eine Datenstruktur aus Feld-Wert-Paaren handelt. Der Wert eines Felds kann andere Dokumente, Arrays und Dokument-Arrays enthalten. Ein Dokument entspricht etwa einer Zeile in einer relationalen Datenbanktabelle.
- Eine Sammlung in Amazon DocumentDB ist eine Gruppe von Dokumenten und entspricht in etwa einer Tabelle in einer relationalen Datenbank.
- Eine Datenbank in Amazon DocumentDB ist ein Satz von Sammlungen und entspricht in etwa einem Schema in einer relationalen Datenbank.

AWS DMS unterstützt zwei Migrationsmodi, wenn Amazon DocumentDB als Quelle verwendet wird: den Dokumentmodus und den Tabellenmodus. Sie geben den Migrationsmodus während der Erstellung des Amazon-DocumentDB-Quellendpunkts in der AWS DMS-Konsole an. Verwenden Sie dazu entweder die Option `Metadatenmodus` oder das zusätzliche Verbindungsattribut `nestingLevel`. Im Folgenden wird erläutert, wie sich der ausgewählte Migrationsmodus auf das resultierende Format der Zieldaten auswirkt.

Dokumentmodus

Im Dokumentmodus wird das JSON-Dokument unverändert migriert. Das bedeutet, dass die Dokumentdaten in einem von zwei Elementen zusammengefasst werden. Wenn Sie eine relationale Datenbank als Ziel verwenden, handelt es sich bei den Daten um eine einzelne Spalte mit der Bezeichnung `_doc` in einer Zieltabelle. Wenn Sie eine nicht relationale Datenbank als Ziel verwenden, handelt es sich bei den Daten um ein einzelnes JSON-Dokument. Der Dokumentmodus ist der Standardmodus, den wir für die Migration zu einem Amazon-DocumentDB-Ziel empfehlen.

Nehmen wir als Beispiel die folgenden Dokumente in einer Amazon-DocumentDB-Sammlung mit dem Namen `myCollection`.

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Nach der Migration der Daten in eine relationale Datenbanktabelle mit Dokumentmodus sind die Daten folgendermaßen strukturiert. Die Datenfelder im Dokument sind in der Spalte `_doc` zusammengefasst.

oid_id	_doc
5a94815f40bd44d1b02bdfe0	{ "a" : 1, "b" : 2, "c" : 3 }
5a94815f40bd44d1b02bdfe1	{ "a" : 4, "b" : 5, "c" : 6 }

Sie können optional das zusätzliche Verbindungsattribut `extractDocID` auf `true` setzen, um eine zweite Spalte mit der Bezeichnung `"_id"` zu erstellen, die als Primärschlüssel fungiert. Wenn Sie die Erfassung von Datenänderungen (Change Data Capture, CDC) verwenden möchten, setzen Sie diesen Parameter auf `true`, es sei denn, Sie verwenden Amazon DocumentDB als Ziel.

Note

Wenn Sie eine neue Sammlung zur Quelldatenbank hinzufügen, erstellt AWS DMS eine neue Zieltabelle für die Sammlung und repliziert alle Dokumente.

Tabellenmodus

Im Tabellenmodus wandelt AWS DMS jedes Feld der obersten Ebene in einem Amazon-DocumentDB-Dokument in eine Spalte in der Zieltabelle um. Wenn ein Feld verschachtelt ist, reduziert AWS DMS die verschachtelten Werte auf eine einzelne Spalte. AWS DMS fügt der Spaltengruppe in der Zieltabelle dann ein Schlüsselfeld und Datentypen hinzu.

Für jedes Amazon-DocumentDB-Dokument fügt AWS DMS jeden Schlüssel und Typ der Spaltengruppe der Zieltabelle hinzu. Im Tabellenmodus migriert AWS DMS beispielsweise das vorherige Beispiel in die folgende Tabelle.

oid_id	a	b	c
5a94815f4 0bd44d1b02bdfe0	1	2	3
5a94815f4 0bd44d1b02bdfe1	4	5	6

Verschachtelte Werte werden in einer Spalte mit durch Punkte getrennte Schlüsselnamen auf eine Ebene gebracht. Die Spalte erhält als Namen die Aneinanderreihung der reduzierten Feldnamen, die durch Punkte voneinander getrennt werden. Beispiel: AWS DMS migriert ein JSON-Dokument mit einem Feld verschachtelter Werte wie beispielsweise {"a" : {"b" : {"c": 1}}}} in eine Spalte mit dem Namen a.b.c..

Zum Erstellen der Zielspalten scannt AWS DMS eine angegebene Anzahl von Amazon-DocumentDB-Dokumenten und erstellt eine Gruppe von allen Feldern und ihren Typen. AWS DMS verwendet diese Gruppe dann, um die Spalten der Zieltabelle zu erstellen. Wenn Sie Ihren Amazon-DocumentDB-Quellendpunkt mithilfe der Konsole erstellen oder ändern, können Sie die Anzahl der zu scannenden Dokumente angeben. Der Standardwert beträgt 1 000 Dokumente. Wenn Sie die AWS CLI verwenden, können Sie das zusätzliche Verbindungsattribut docsToInvestigate verwenden.

Hier sehen Sie, wie AWS DMS Dokumente und Sammlungen im Tabellenmodus verwaltet:

- Wenn Sie ein Dokument zu einer vorhandenen Sammlung hinzufügen, wird das Dokument repliziert. Wenn es Felder gibt, die am Ziel nicht vorhanden sind, werden diese Felder nicht repliziert.
- Wenn Sie ein Dokument aktualisieren, wird das aktualisierte Dokument repliziert. Wenn es Felder gibt, die am Ziel nicht vorhanden sind, werden diese Felder nicht repliziert.
- Das Löschen eines Dokuments wird vollständig unterstützt.
- Das Hinzufügen einer neuen Sammlung führt nicht zu einer neuen Tabelle in der Zieldatenbank, wenn der Vorgang zeitgleich mit einer CDC-Aufgabe erfolgt.
- In der Phase der Erfassung von Datenänderungen (Change Data Capture, CDC) unterstützt AWS DMS das Umbenennen einer Sammlung nicht.

Themen

- [Festlegen von Berechtigungen zur Verwendung von Amazon DocumentDB als Quelle](#)
- [Konfigurieren von CDC für einen Amazon-DocumentDB-Cluster](#)
- [Herstellen einer Verbindung mit Amazon DocumentDB über TLS](#)
- [Erstellen eines Amazon-DocumentDB-Quellendpunkts](#)
- [Segmentieren von Amazon-DocumentDB-Sammlungen und parallele Migration](#)
- [Migration mehrerer Datenbanken bei Verwendung von Amazon DocumentDB als Quelle für AWS DMS](#)

- [Einschränkungen bei Verwendung von Amazon DocumentDB als Quelle für AWS DMS](#)
- [Verwenden von Endpunkteinstellungen mit Amazon DocumentDB als Quelle](#)
- [Quelldatentypen für Amazon DocumentDB](#)

Festlegen von Berechtigungen zur Verwendung von Amazon DocumentDB als Quelle

Wenn Sie die Amazon-DocumentDB-Quelle für eine AWS DMS-Migration verwenden, können Sie ein Benutzerkonto mit Root-Berechtigungen erstellen. Alternativ können Sie einen Benutzer erstellen, der ausschließlich über Berechtigungen für die zu migrierende Datenbank verfügt.

Mit dem folgenden Code wird ein Benutzer als Root-Konto erstellt.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  })
```

Für Amazon DocumentDB 3.6 wird mit dem folgenden Code ein Benutzer mit minimalen Berechtigungen für die zu migrierende Datenbank erstellt.

```
use database_to_migrate
db.createUser(
  {
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "db_name" }, "read" ]
  })
```

Für Amazon DocumentDB 4.0 und höher verwendet AWS DMS einen bereitstellungsweiten Änderungsstream. Hier wird mit dem folgenden Code ein Benutzer mit minimalen Berechtigungen erstellt.

```
db.createUser(
  {
```

```
user: "dms-user",
pwd: "password",
roles: [ { role: "readAnyDatabase", db: "admin" } ]
})
```

Konfigurieren von CDC für einen Amazon-DocumentDB-Cluster

Um die laufende Replikation oder CDC mit Amazon DocumentDB verwenden zu können, benötigt AWS DMS Zugriff auf die Änderungsstreams des Amazon-DocumentDB-Clusters. Eine Beschreibung der zeitlich geordneten Reihenfolge von Aktualisierungsereignissen in den Sammlungen und Datenbanken Ihres Clusters finden Sie unter [Using Change Streams](#) im Entwicklerhandbuch für Amazon DocumentDB.

Authentifizieren Sie sich mit der MongoDB-Shell bei Ihrem Amazon-DocumentDB-Cluster. Führen Sie anschließend den folgenden Befehl aus, um Änderungsstreams zu aktivieren.

```
db.adminCommand({modifyChangeStreams: 1,
  database: "DB_NAME",
  collection: "",
  enable: true});
```

Damit wird der Änderungsstream für alle Sammlungen in Ihrer Datenbank aktiviert. Nach Aktivierung der Änderungsstreams können Sie eine Migrationsaufgabe erstellen, um vorhandene Daten zu migrieren und gleichzeitig laufende Änderungen zu replizieren. Auch nach dem Laden der Massendaten erfasst AWS DMS weiterhin Änderungen und wendet sie an. Schließlich werden die Quell- und Zieldatenbank synchronisiert, sodass die Ausfallzeit für eine Migration minimiert wird.

Note

AWS DMS verwendet das Protokoll der Operationen (Oplog), um Änderungen während der laufenden Replikation zu erfassen. Wenn Amazon DocumentDB die Datensätze aus dem Oplog löscht, bevor sie von AWS DMS gelesen werden, schlagen Ihre Aufgaben fehl. Wir empfehlen, die Oplog-Größe so zu ändern, dass die Änderungen mindestens 24 Stunden lang beibehalten werden.

Herstellen einer Verbindung mit Amazon DocumentDB über TLS

Standardmäßig akzeptiert ein neu erstellter Amazon-DocumentDB-Cluster sichere Verbindungen nur mit Transport Layer Security (TLS). Wenn TLS aktiviert ist, erfordert jede Verbindung mit Amazon DocumentDB einen öffentlichen Schlüssel.

Sie können den öffentlichen Schlüssel für Amazon DocumentDB abrufen, indem Sie die Datei `rds-combined-ca-bundle.pem` aus einem AWS-gehosteten Amazon-S3-Bucket herunterladen.

Weitere Informationen zum Herunterladen dieser Datei finden Sie unter [Encrypting connections using TLS](#) im Entwicklerhandbuch für Amazon DocumentDB.

Nachdem Sie die Datei `rds-combined-ca-bundle.pem` heruntergeladen haben, können Sie den darin enthaltenen öffentlichen Schlüssel in AWS DMS importieren. Das entsprechende Verfahren wird im Folgenden beschrieben.

So importieren Sie den öffentlichen Schlüssel mithilfe der AWS DMS-Konsole

1. Melden Sie sich bei der AWS Management Console an und wählen Sie AWS DMS aus.
2. Wählen Sie im Navigationsbereich **Certificates** aus.
3. Wählen Sie **Import certificate** (Zertifikat importieren). Die Seite **Importieren eines CA-Zertifikats** wird angezeigt.
4. Führen Sie im Abschnitt **Zertifikatkonfiguration** eine der folgenden Aktionen aus:
 - Geben Sie in **Zertifikat-ID** einen eindeutigen Namen für das Zertifikat ein, z. B. `docdb-cert`.
 - Wählen Sie **Datei auswählen**, navigieren Sie zum Speicherort der Datei `rds-combined-ca-bundle.pem` und wählen Sie diese aus.
5. Klicken Sie auf **Add new CA certificate** (Neues CA-Zertifikat hinzufügen).

Im folgenden Beispiel für die AWS CLI wird der AWS DMS-Befehl `import-certificate` verwendet, um die Datei `rds-combined-ca-bundle.pem` mit dem öffentlichen Schlüssel zu importieren.

```
aws dms import-certificate \  
  --certificate-identifier docdb-cert \  
  --certificate-pem file:///./rds-combined-ca-bundle.pem
```

Erstellen eines Amazon-DocumentDB-Quellendpunkts

Sie können einen Amazon-DocumentDB-Quellendpunkt mithilfe der Konsole oder der AWS CLI erstellen. Gehen Sie bei Verwendung der Konsole folgendermaßen vor.

So konfigurieren Sie einen Amazon-DocumentDB-Quellendpunkt mithilfe der AWS DMS-Konsole

1. Melden Sie sich bei der AWS Management Console an und wählen Sie AWS DMS aus.
2. Wählen Sie im Navigationsbereich Endpunkte und anschließend Endpunkt erstellen aus.
3. Geben Sie für Endpunkt-Kennung einen Namen ein, anhand dessen Sie ihn leicht identifizieren können, z. B. `docdb-source`.
4. Wählen Sie für Quellen-Engine die Option Amazon DocumentDB (mit MongoDB-Kompatibilität) aus.
5. Geben Sie für Servername den Namen des Servers ein, auf dem sich der Amazon-DocumentDB-Datenbankendpunkt befindet. Sie könnten beispielsweise den öffentlichen DNS-Namen Ihrer Amazon-EC2-Instance eingeben, z. B. `democluster.cluster-cjf6q8nxfefi.us-east-2.docdb.amazonaws.com`.
6. Geben Sie für Port 27017 ein.
7. Für SSL mode (SSL-Modus) wählen Sie `verify-full` aus. Wenn Sie SSL in Ihrem Amazon-DocumentDB-Cluster deaktiviert haben, können Sie diesen Schritt überspringen.
8. Wählen Sie für CA-Zertifikat das Amazon-DocumentDB-Zertifikat `rd5-combined-ca-bundle.pem` aus. Anweisungen zum Hinzufügen dieses Zertifikats finden Sie unter [Herstellen einer Verbindung mit Amazon DocumentDB über TLS](#).
9. Geben Sie für Datenbankname den Namen Ihrer Datenbank ein.

Gehen Sie bei Verwendung der CLI folgendermaßen vor.

So konfigurieren Sie einen Amazon-DocumentDB-Quellendpunkt mithilfe der AWS CLI

- Führen Sie den folgenden AWS DMS-Befehl `create-endpoint` aus, um einen Amazon-DocumentDB-Quellendpunkt zu konfigurieren. Ersetzen Sie die Platzhalter durch Ihre eigenen Werte.

```
aws dms create-endpoint \  
    --endpoint-identifier a_memorable_name \  
    --endpoint-type source \  
    --engine-name docdb \  
    --server-name server_name \  
    --port 27017 \  
    --ssl-mode verify-full \  
    --ca-certificate rd5-combined-ca-bundle.pem \  
    --database-name database_name
```



```
--username value \  
--password value \  
--server-name servername_where_database_endpoint_resides \  
--port 27017 \  
--database-name name_of_endpoint_database
```

Segmentieren von Amazon-DocumentDB-Sammlungen und parallele Migration

Um die Leistung einer Migrationsaufgabe zu verbessern, unterstützen Amazon-DocumentDB-Quellendpunkte zwei Optionen des Features für paralleles vollständiges Laden bei der Tabellenzuordnung. Anders ausgedrückt: Sie können eine Sammlung parallel migrieren, indem Sie entweder die Optionen für die automatische Segmentierung oder für die Bereichssegmentierung der Tabellenzuordnung für paralleles vollständiges Laden in den JSON-Einstellungen verwenden. Mit den Optionen für die automatische Segmentierung können Sie die Kriterien angeben, nach denen AWS DMS Ihre Migrationsquelle in jedem Thread automatisch segmentieren soll. Mit den Optionen für die Bereichssegmentierung können Sie AWS DMS den spezifischen Bereich jedes Segments mitteilen, den DMS in jedem Thread migrieren soll. Weitere Informationen zu diesen Einstellungen finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Parallele Migration einer Amazon-DocumentDB-Datenbank mithilfe von Bereichen für die automatische Segmentierung

Sie können Ihre Dokumente parallel migrieren, indem Sie die Kriterien angeben, nach denen AWS DMS Ihre Daten für jeden Thread automatisch partitionieren (segmentieren) soll – insbesondere die Anzahl an Dokumenten, die pro Thread migriert werden sollen. Bei diesem Verfahren versucht AWS DMS, die Segmentgrenzen zu optimieren, um eine maximale Leistung pro Thread zu erzielen.

Sie können die Segmentierungskriterien mithilfe der folgenden Tabelleneinstellungsoptionen in der Tabellenzuordnung angeben:

Tabelleneinstellungsoption	Beschreibung
"type"	(Erforderlich) Setzen Sie dies für Amazon DocumentDB als Quelle auf "partitions-auto".

Tabelleneinstelloption	Beschreibung
"number-of-partitions"	(Optional) Gesamtzahl der für die Migration verwendeten Partitionen (Segmente). Der Standardwert ist 16.
"collection-count-from-metadata"	(Optional) Wenn dies auf <code>true</code> gesetzt ist, verwendet AWS DMS eine geschätzte Sammlungsanzahl, um die Anzahl von Partitionen zu bestimmen. Wenn es auf <code>false</code> gesetzt ist, verwendet AWS DMS die tatsächliche Sammlungsanzahl. Der Standardwert ist <code>true</code> .
"max-records-skip-per-page"	(Optional) Die Anzahl der Datensätze, die bei der Ermittlung der Grenzen für jede Partition gleichzeitig übersprungen werden sollen. AWS DMS verwendet einen Ansatz zum seitenweisen Überspringen, um die Mindestgrenze für eine Partition zu bestimmen. Der Standardwert ist 10 000. Das Festlegen eines relativ großen Werts kann zu Cursor-Timeouts und Aufgabenfehlern führen. Das Festlegen eines relativ niedrigen Werts führt zu mehr Operationen pro Seite und zu einem langsameren vollständigen Laden.
"batch-size"	(Optional) Begrenzt die Anzahl der in einem Stapel zurückgegebenen Dokumente. Für jeden Stapel ist ein Round-Trip zum Server erforderlich. Wenn die Stapelgröße Null (0) ist, verwendet der Cursor die vom Server definierte maximale Stapelgröße. Der Standardwert ist 0.

Das folgende Beispiel zeigt eine Tabellenzuordnung für die automatische Segmentierung.

```
{
  "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
  }
}
]
```

Für die automatische Segmentierung gelten die folgenden Einschränkungen. Bei der Migration für die einzelnen Segmente werden die Sammlungsanzahl und der `_id`-Mindestwert für die Sammlung separat abgerufen. Anschließend wird die Mindestgrenze für dieses Segment anhand eines seitenweisen Überspringens berechnet. Stellen Sie daher sicher, dass der `_id`-Mindestwert für jede Sammlung konstant bleibt, bis alle Segmentgrenzen in der Sammlung berechnet wurden. Wenn Sie den `_id`-Mindestwert für eine Sammlung während der Berechnung der Segmentgrenze ändern, kann dies zu Datenverlust oder zu Fehlern aufgrund von doppelten Zeilen führen.

Parallele Migration einer Amazon-DocumentDB-Datenbank mithilfe von spezifischen Segmentbereichen

Das folgende Beispiel zeigt eine Amazon-DocumentDB-Sammlung mit sieben Elementen und `_id` als Primärschlüssel.

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

Um die Sammlung in drei Segmente aufzuteilen und parallel zu migrieren, können Sie Ihrer Migrationsaufgabe Regeln für die Tabellenzuordnung hinzufügen, wie im folgenden JSON-Beispiel gezeigt.

```
{ // 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

```

Mit dieser Definition für die Tabellenzuordnung wird die Quellsammlung in drei Segmente aufgeteilt, die parallel migriert werden. Im Folgenden sind die Segmentierungsgrenzen aufgeführt.

```
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)
```

Nach Abschluss der Migrationsaufgabe können Sie anhand der Aufgabenprotokolle überprüfen, ob die Tabellen parallel geladen wurden, wie im folgenden Beispiel gezeigt. Sie können auch die zum Entladen der einzelnen Segmente aus der Quelltable verwendete Amazon-DocumentDB-Klausel `find` überprüfen.

```
[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.
```

AWS DMS unterstützt derzeit die folgenden Amazon-DocumentDB-Datentypen als Segmentschlüsselspalte:

- Double
- String
- ObjectId
- 32-Bit-Ganzzahl
- 64-Bit-Ganzzahl

Migration mehrerer Datenbanken bei Verwendung von Amazon DocumentDB als Quelle für AWS DMS

Die AWS DMS-Versionen 3.4.5 und höher unterstützen die Migration von mehreren Datenbanken in einer einzigen Aufgabe nur für die Amazon-DocumentDB-Versionen 4.0 und höher. Gehen Sie folgendermaßen vor, um mehrere Datenbanken zu migrieren:

1. Wenn Sie den Amazon-DocumentDB-Quellendpunkt erstellen:

- Lassen Sie in der AWS Management Console für AWS DMS Datenbankname unter Endpunktkonfiguration auf der Seite Endpunkt erstellen leer.
- Weisen Sie in der AWS Command Line Interface (AWS CLI) dem DatabaseName Parameter in DocumentDBSettings, den Sie für die CreateEndpoint Aktion angeben, einen leeren Zeichenfolgenwert zu.

2. Geben Sie für jede Datenbank, die Sie von diesem Amazon-DocumentDB-Quellendpunkt migrieren möchten, den Namen jeder Datenbank als Namen eines Schemas in der Tabellenzuordnung für die Aufgabe an, entweder mithilfe der geführten Eingabe in der Konsole oder direkt in JSON. Weitere Informationen zur geführten Eingabe finden Sie in der Beschreibung zu [Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole](#). Weitere Informationen zu JSON finden Sie unter [Auswahlregeln und Aktionen](#).

Sie können beispielsweise die folgende JSON-Struktur angeben, um drei Amazon-DocumentDB-Datenbanken zu migrieren.

Example Migrieren aller Tabellen in einem Schema

Die folgende JSON-Struktur migriert alle Tabellen aus den Datenbanken Customers, Orders und Suppliers im Quellendpunkt zum Zielendpunkt.

```
{
  "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"
    }
  ]
}
```


Einschränkungen bei Verwendung von Amazon DocumentDB als Quelle für AWS DMS

Die folgenden Einschränkungen gelten bei Verwendung von Amazon DocumentDB als Quelle für AWS DMS:

- Wenn die Option `_id` als separate Spalte gesetzt ist, darf der ID-String 200 Zeichen nicht überschreiten.
- Objekt-ID und Array-Schlüssel werden in Spalten konvertiert, die mit dem Präfix `oid` und `array` im Tabellenmodus versehen sind.

Intern wird auf diese Spalten mit den mit Präfix versehenen Namen verwiesen. Wenn Sie in AWS DMS Transformationsregeln verwenden, die auf diese Spalten verweisen, geben Sie unbedingt den mit Präfix versehenen Spaltennamen an. Geben Sie beispielsweise `$_oid__id` anstatt `$_id` oder `$_array__addresses` anstatt `$_addresses` an.

- Namen von Sammlungen und Schlüsseln dürfen das Dollarzeichen (\$) nicht enthalten.
- Der Tabellenmodus und der Dokumentmodus haben die oben genannten Einschränkungen.
- Für die parallele Migration mithilfe der automatischen Segmentierung gelten die oben beschriebenen Einschränkungen.
- Eine Amazon-DocumentDB-Quelle (MongoDB-kompatibel) unterstützt die Verwendung eines spezifischen Zeitstempels als Startposition für CDC nicht. Eine laufende Replikationsaufgabe beginnt unabhängig vom Zeitstempel mit der Erfassung von Änderungen.
- Bei Verwendung von DocumentDB (MongoDB-kompatibel) als Quelle kann DMS maximal 250 Datensätze pro Sekunde verarbeiten.
- AWS DMS unterstützt keine Dokumente mit einer größeren Verschachtelungsebene als 97.
- Quellfilter werden für DocumentDB nicht unterstützt.
- AWS DMS unterstützt keine CDC-Replikation (Change Data Capture) für DocumentDB als Quelle im elastischen Cluster-Modus.

Verwenden von Endpunkteinstellungen mit Amazon DocumentDB als Quelle

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer Amazon-DocumentDB-Quelldatenbank verwenden. Sie legen die Einstellungen fest, wenn Sie den Quellendpunkt mithilfe der AWS DMS-Konsole erstellen oder indem Sie den Befehl `create-endpoint` in der [AWS CLI](#) aufrufen. Verwenden Sie die JSON-Syntax `--doc-db-settings '{"EndpointSetting": "value", ...}'`.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Amazon DocumentDB als Quelle verwenden können.

Attributname	Zulässige Werte	Standardwert und Beschreibung
NestingLevel	"none" "one"	"none" – Geben Sie "none" an, um den Dokumentmodus zu verwenden. Geben Sie "one" an, um den Tabellenmodus zu verwenden.
ExtractDocID	boolesch true false	false – Verwenden Sie dieses Attribut, wenn NestingLevel auf "none" gesetzt ist. Wenn Ihre Zieldatenbank Amazon DocumentDB ist, legen Sie '{"ExtractDocID": true}' fest.
DocsToInvestigate	Eine positive Ganzzahl größer als 0.	1000 – Verwenden Sie dieses Attribut, wenn NestingLevel auf "one" gesetzt ist.
ReplicateShardCollections	boolesch true false	Wenn Sie true festlegen, repliziert AWS DMS Daten in Shard-Sammlungen. AWS DMS verwendet diese Einstellung nur, wenn es sich beim Zielendpunkt um einen elastischen DocumentDB-Cluster handelt. Wenn Sie für diese Einstellung true festlegen, beachten Sie Folgendes: <ul style="list-style-type: none"> • Sie müssen TargetTablePrepMode auf nothing setzen. • AWS DMS setzt useUpdateLookup automatisch auf false.

Quelldatentypen für Amazon DocumentDB

Die folgende Tabelle enthält die Amazon-DocumentDB-Quelldatentypen, die bei Verwendung von AWS DMS unterstützt werden. Diese Tabelle enthält außerdem die Standardzuordnung von AWS DMS-Datentypen. Weitere Informationen zu den Datentypen finden Sie unter [BSON types](#) in der Dokumentation zu MongoDB.

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

Amazon-DocumentDB-Datentypen	AWS DMS-Datentypen
Boolesch	Bool
Binär	BLOB
Datum	Datum
Zeitstempel	Datum
Int	INT4
Long	INT8
Double	REAL8
String (UTF-8)	CLOB
Array	CLOB
OID	String

Verwenden von Amazon S3 als Quelle für AWS DMS

Sie können Daten aus einem Amazon-S3-Bucket mithilfe von AWS DMS migrieren. Erteilen Sie dazu Zugriff auf einen Amazon-S3-Bucket, der eine oder mehrere Datendateien enthält. Fügen Sie diesem S3-Bucket eine JSON-Datei hinzu, die die Zuweisung zwischen den Daten und den Datenbanktabellen der Daten in diesen Dateien beschreibt.

Die Quelldatendateien müssen im Amazon-S3-Bucket enthalten sein, bevor die Volllastaufgabe beginnt. Geben Sie den Bucket-Namen mit dem `bucketName`-Parameter an.

Die Quelldatendateien müssen im CSV-Format vorliegen. Verwenden Sie die folgende Namenskonvention, wenn Sie sie benennen. In dieser Konvention ist *schemaName* das Quellschema und *tableName* ist der Name einer Tabelle in diesem Schema.

```
/schemaName/tableName/LOAD001.csv  
/schemaName/tableName/LOAD002.csv  
/schemaName/tableName/LOAD003.csv  
...
```

Angenommen, Ihre Datendateien befinden sich in mybucket im folgenden Amazon-S3-Pfad.

```
s3://mybucket/hr/employee
```

Zur Ladezeit nimmt AWS DMS an, dass der Name des Quellschemas hr und der Name der Quelltable employee ist.

Zusätzlich zu bucketName (erforderlich) können Sie optional einen bucketFolder-Parameter bereitstellen, um anzugeben, wo AWS DMS im Amazon-S3-Bucket nach Datendateien suchen soll. Wenn Sie im vorherigen Beispiel bucketFolder auf sourcedata setzen, liest AWS DMS die Datendateien im folgenden Pfad.

```
s3://mybucket/sourcedata/hr/employee
```

Sie können mithilfe der zusätzlichen Verbindungsattribute die Trennzeichen für die Spalte und die Zeile sowie den Nullwertindikator und andere Parameter festlegen. Weitere Informationen finden Sie unter [Endpunkteinstellungen für Amazon S3 als Quelle für AWS DMS](#).

Sie können einen Bucket-Eigentümer angeben und Sniping verhindern, wenn Sie die Amazon-S3-Endpunkteinstellung ExpectedBucketOwner verwenden, wie im Folgenden gezeigt. Wenn Sie dann eine Anforderung stellen, um eine Verbindung zu testen oder eine Migration durchzuführen, gleicht S3 die Konto-ID des Bucket-Eigentümers mit dem angegebenen Parameter ab.

```
--s3-settings='{ "ExpectedBucketOwner": "AWS_Account_ID" }'
```

Themen

- [Definieren von externen Tabellen für Amazon S3 als Quelle für AWS DMS](#)
- [Verwenden von CDC mit Amazon S3 als Quelle für AWS DMS](#)

- [Voraussetzungen für die Verwendung von Amazon S3 als Quelle für AWS DMS](#)
- [Einschränkungen bei Verwendung von Amazon S3 als Quelle für AWS DMS](#)
- [Endpunkteinstellungen für Amazon S3 als Quelle für AWS DMS](#)
- [Quelldatentypen für Amazon S3](#)

Definieren von externen Tabellen für Amazon S3 als Quelle für AWS DMS

Zusätzlich zu den Datendateien müssen Sie auch eine externe Tabellendefinition bereitstellen. Eine externe Tabellendefinition ist ein JSON-Dokument, in dem beschrieben wird, wie AWS DMS die Daten aus Amazon S3 interpretieren soll. Die maximale Größe dieses Dokuments ist 2 MB. Wenn Sie einen Quellendpunkt mithilfe der AWS DMS-Managementkonsole erstellen, können Sie JSON direkt in das Tabellenzuweisungsfeld eingeben. Wenn Sie die AWS Command Line Interface (AWS CLI) oder AWS DMS-API zum Durchführen von Migrationen verwenden, können Sie eine JSON-Datei zur Angabe der externen Tabellendefinition erstellen.

Angenommen, Sie haben eine Datei mit den folgenden Daten.

```
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
```

Im Folgenden sehen Sie ein Beispiel für eine externe Tabellendefinition für diese Daten.

```
{
  "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"
}
]
}

```

In diesem JSON-Dokument sind folgende Elemente enthalten:

TableCount – die Anzahl der Quelltabellen. In diesem Beispiel es nur eine Tabelle.

Tables – ein Array bestehend aus einer JSON-Zuweisung je Quelltable. In diesem Beispiel es nur eine Zuweisung. Jede Zuweisung umfasst die folgenden Schlüsselemente:

- **TableName** – der Name der Quelltable.
- **TablePath** – der Pfad in Ihrem Amazon-S3-Bucket, in dem AWS DMS die Datei mit den Daten für die Volllastaufgabe finden kann. Wenn ein `bucketFolder`-Wert angegeben ist, wird dieser Wert dem Pfad vorangestellt.
- **TableOwner** – der Name des Schemas für diese Tabelle.
- **TableColumns** – ein Array einer oder mehrerer Zuweisungen, die jeweils eine Spalte in der Quelltable beschreiben:
 - **ColumnName** – der Name einer Spalte in der Quelltable.
 - **ColumnType** – der Datentyp für die Spalte. Die zulässigen Datentypen finden Sie unter [Quelldatentypen für Amazon S3](#).

- `ColumnLength` – die Anzahl an Bytes in dieser Spalte. Die maximale Spaltenlänge ist auf 2 147 483 647 Bytes (2 047 MegaBytes) begrenzt, da eine S3-Quelle den Modus FULL LOB nicht unterstützt. `ColumnLength` gilt für die folgenden Datentypen:
 - `BYTE`
 - `STRING`
- `ColumnNullable` – ein boolescher Wert, der `true` lautet, wenn diese Spalte NULL-Werte enthalten darf (Standard = `false`).
- `ColumnIsPk` – ein boolescher Wert, der `true` lautet, wenn diese Spalte Teil des Primärschlüssels ist (Standard = `false`).
- `ColumnDateFormat` – das Eingabedatumsformat für eine Spalte mit den Typen `DATE`, `TIME` und `DATETIME`, das verwendet wird, um eine Datenzeichenfolge in ein Datumsobjekt zu analysieren. Mögliche Werte sind:

```
- 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` – die Gesamtanzahl der Spalten. Diese Anzahl muss der Anzahl der Elemente im `TableColumns`-Array entsprechen.

Wenn Sie nichts anderes angeben, geht AWS DMS davon aus, dass `ColumnLength` Null ist.

Note

In unterstützten Versionen von AWS DMS können die S3-Quelldaten auch eine optionale Operationsspalte als erste Spalte vor dem Spaltenwert `TableName` enthalten. Diese Operationsspalte identifiziert die Operation (`INSERT`), die verwendet wird, um die Daten während eines vollständigen Ladevorgangs zu einem S3-Zielendpunkt zu migrieren. Falls vorhanden, ist der Wert dieser Spalte das erste Zeichen des `INSERT`-Schlüsselworts der Operation (`I`). Sofern vorhanden, gibt diese Spalte in der Regel an, dass die S3-Quelle während einer früheren Migration durch DMS als S3-Ziel erstellt wurde.

In früheren DMS-Versionen als 3.4.2 war diese Spalte in S3-Quelldaten, die aus einer früheren DMS-Volllastaufgabe erstellt wurden, nicht vorhanden. Durch das Hinzufügen dieser Spalte zu den S3-Zieldaten werden alle Zeilen in einem konsistenten Format zum S3-Ziel geschrieben. Dies geschieht unabhängig davon, ob die Daten während eines vollständigen oder während eines CDC-Ladevorgangs geschrieben werden. Weitere Informationen zu den Optionen zum Formatieren von S3-Zieldaten finden Sie unter [Angabe von Quelldatenbankoperationen in migrierten S3-Daten](#).

Für eine Spalte des Typs NUMERIC müssen Sie die Genauigkeit und Skalierung festlegen. Genauigkeit ist die Gesamtanzahl der Stellen in einer Zahl und Skalierung die Anzahl der Dezimalstellen. Dafür verwenden Sie die Elemente `ColumnPrecision` und `ColumnScale`, wie im Folgenden dargestellt.

```
...
  {
    "ColumnName": "HourlyRate",
    "ColumnType": "NUMERIC",
    "ColumnPrecision": "5"
    "ColumnScale": "2"
  }
...
```

Geben Sie für eine Spalte vom Typ DATETIME mit Daten, die Sekundenbruchteile enthalten, die Skalierung an. Skalierung ist die Anzahl der Dezimalstellen für die Sekundenbruchteile. Diese kann zwischen 0 und 9 liegen. Verwenden Sie dafür das Element `ColumnScale`, wie im Folgenden dargestellt.

```
...
{
  "ColumnName": "HireDate",
  "ColumnType": "DATETIME",
  "ColumnScale": "3"
}
...
```

Wenn Sie nichts anderes angeben, geht AWS DMS davon aus, dass `ColumnScale` Null ist, und die Sekundenbruchteile werden gekürzt.

Verwenden von CDC mit Amazon S3 als Quelle für AWS DMS

Nachdem AWS DMS einen vollständigen Datenladevorgang ausgeführt hat, können optional Datenänderungen für den Zielendpunkt repliziert werden. Zu diesem Zweck laden Sie Dateien zur Erfassung von Datenänderungen (Change Data Capture (CDC)-Dateien) in Ihren Amazon-S3-Bucket hoch. AWS DMS liest diese CDC-Dateien, wenn Sie sie hochladen. Anschließend werden die Änderungen für den Zielendpunkt übernommen.

Die CDC-Dateien werden wie folgt benannt:

```
CDC00001.csv  
CDC00002.csv  
CDC00003.csv  
...
```

Note

Um CDC-Dateien im Änderungsdatenordner zu replizieren, laden Sie sie erfolgreich in lexikalischer (sequenzieller) Reihenfolge hoch. Laden Sie z. B. die Datei „CDC00002.csv“ vor der Datei „CDC00003.csv“ hoch. Andernfalls wird die Datei „CDC00002.csv“ übersprungen und nicht repliziert, wenn Sie sie zeitlich nach „CDC00003.csv“ laden. Die Datei „CDC00004.csv“ wird jedoch erfolgreich repliziert, wenn sie nach der Datei „CDC00003.csv“ geladen wird.

Um anzugeben, wo AWS DMS die Dateien finden kann, legen Sie den Parameter `cdcPath` fest. Wenn Sie im vorherigen Beispiel `cdcPath` auf *changedata* setzen, liest AWS DMS die CDC-Dateien im folgenden Pfad.

```
s3://mybucket/changedata
```

Wenn Sie `cdcPath` auf *changedata* und `bucketFolder` auf *myFolder* setzen, liest AWS DMS die CDC-Dateien im folgenden Pfad.

```
s3://mybucket/myFolder/changedata
```

Die Datensätze in einer CDC-Datei werden wie folgt formatiert:

- Operation – die Änderungsoperation, die ausgeführt werden soll: INSERT oder I, UPDATE oder U oder DELETE oder D. Diese Schlüsselwort- und Zeichenwerte unterscheiden zwischen Groß- und Kleinschreibung.

Note

In unterstützten AWS DMS-Versionen kann AWS DMS die Operation, die für jeden Ladedatensatz ausgeführt werden soll, auf zwei Arten identifizieren. AWS DMS kann diese anhand des Schlüsselwortwerts des Datensatzes (z. B. INSERT) oder anhand des Anfangszeichens des Schlüsselworts (z. B. I) ermitteln. In früheren Versionen konnte AWS DMS die Ladeoperation nur am vollständigen Wert des Schlüsselworts erkennen. In früheren Versionen von AWS DMS wurde der vollständige Wert des Schlüsselworts geschrieben, um die CDC-Daten zu protokollieren. Vorherige Versionen schrieben den Operationswert darüber hinaus nur mit dem Anfangszeichen des Schlüsselworts in S3-Ziele.

Dadurch, dass nun beide Formate erkannt werden, kann AWS DMS die Operation verarbeiten, unabhängig davon, wie die Operationsspalte zum Erstellen der S3-Quelldaten geschrieben wurde. Dieser Ansatz unterstützt die Verwendung von S3-Zieldaten als Quelle für eine spätere Migration. Dank dieses Ansatzes müssen Sie nicht länger das Format des Anfangszeichens eines Schlüsselworts ändern, das in der Operationsspalte der späteren S3-Quelle erscheint

- Tabellenname – der Name der Quelltable.
- Schemaname – der Name des Quellschemas.
- Daten – eine oder mehrere Spalten, die die zu ändernden Daten darstellen.

Im Folgenden sehen Sie ein Beispiel einer CDC-Datei für eine Tabelle mit dem Namen 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
```

Voraussetzungen für die Verwendung von Amazon S3 als Quelle für AWS DMS

Wenn Sie Amazon S3 als Quelle für AWS DMS verwenden möchten, muss sich der S3-Quell-Bucket in derselben AWS-Region wie die DMS-Replikations-Instance befinden, mit der Sie Ihre

Daten migrieren. Darüber hinaus muss das AWS-Konto, das Sie für die Migration verwenden, über Lesezugriff auf den Quell-Bucket verfügen.

Die dem Benutzerkonto zugewiesene AWS Identity and Access Management (IAM)-Rolle, mit der die Migrationsaufgabe erstellt wird, muss den folgenden Satz von Berechtigungen aufweisen.

```
{
  "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*"
      ]
    }
  ]
}
```

Die dem Benutzerkonto zugewiesene AWS Identity and Access Management (IAM)-Rolle, mit der die Migrationsaufgabe erstellt wird, muss den folgenden Satz von Berechtigungen aufweisen, wenn die Versionsverwaltung für den Amazon-S3-Bucket aktiviert ist.

```
{
  "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*"
    ]
  }
]
}

```

Einschränkungen bei Verwendung von Amazon S3 als Quelle für AWS DMS

Die folgenden Einschränkungen gelten bei Verwendung von Amazon S3 als Quelle:

- Aktivieren Sie die Versionsverwaltung für S3 nicht. Wenn Sie die Versionsverwaltung für S3 benötigen, verwenden Sie Lebenszyklusrichtlinien, um alte Versionen aktiv zu löschen. Andernfalls kann es aufgrund eines Timeouts eines `list-object`-Aufrufs in S3 zu Verbindungsfehlern beim Endpunkttest kommen. Informationen zum Erstellen einer Lebenszyklusrichtlinie für einen S3-Bucket finden Sie unter [Verwalten Ihres Speicher-Lebenszyklus](#). Informationen zum Löschen einer Version eines S3-Objekts finden Sie unter [Löschen von Objekten aus einem versionsverwaltungsfähigen Bucket](#).
- Ein VPC-fähiger S3-Bucket (Gateway-VPC) wird in den Versionen 3.4.7 und höher unterstützt.

Endpunkteinstellungen für Amazon S3 als Quelle für AWS DMS

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer Amazon-S3-Quelldatenbank verwenden. Sie legen die Einstellungen fest, wenn Sie den Quellendpunkt mithilfe der AWS DMS-Konsole erstellen oder indem Sie den Befehl `create-endpoint` in der [AWS CLI](#) aufrufen. Verwenden Sie die JSON-Syntax `--s3-settings '{"EndpointSetting": "value", ...}'`.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Amazon S3 als Quelle verwenden können.

Option	Beschreibung
BucketFolder	<p>(Optional) Ein Ordnername im S3-Bucket. Wenn dieses Attribut angegeben ist, werden die Quelldatendateien und CDC-Dateien im Pfad <code>s3://myBucket/bucketFolder /schemaName /tableName /</code> bzw. <code>s3://myBucket/bucketFolder /</code> gelesen. Wenn dieses Attribut nicht angegeben ist, wird der Pfad <code>schemaName /tableName /</code> verwendet.</p> <pre>'{"BucketFolder": " <i>sourceData</i> "'}</pre>
BucketName	<p>Der Name des S3-Buckets.</p> <pre>'{"BucketName": " <i>myBucket</i> "'}</pre>
CdcPath	<p>Der Speicherort von CDC-Dateien. Dieses Attribut ist erforderlich, wenn eine Aufgabe Änderungsdaten erfasst. Andernfalls ist es optional. Wenn CdcPath vorhanden ist, liest AWS DMS CDC-Dateien von diesem Pfad und repliziert die Datenänderungen auf dem Zielpunkt. Weitere Informationen finden Sie unter Verwenden von CDC mit Amazon S3 als Quelle für AWS DMS.</p> <pre>'{"CdcPath": " <i>changeData</i> "'}</pre>
CsvDelimiter	<p>Das Trennzeichen, das zum Trennen von Spalten in den Quelldateien dient. Standardmäßig wird ein Komma verwendet. Ein Beispiel folgt.</p> <pre>'{"CsvDelimiter": ", "'}</pre>
CsvNullValue	<p>Eine benutzerdefinierte Zeichenfolge, die AWS DMS beim Lesen aus der Quelle als Null behandelt. Der Standardwert ist eine leere Zeichenfolge. Wenn Sie diesen Parameter nicht festlegen, behandelt AWS DMS eine leere Zeichenfolge als Nullwert. Wenn Sie diesen Parameter auf eine Zeichenfolge wie „\N“ setzen, behandelt AWS DMS diese Zeichenfolge als Nullwert und leere Zeichenfolgen als leere Zeichenfolgewerte.</p>
CsvRowDelimiter	<p>Das Trennzeichen, das zum Trennen von Zeilen in den Quelldateien dient. Standardmäßig wird ein Zeilenumbruch (\n) verwendet.</p>

Option	Beschreibung
	<code>'{"CsvRowDelimiter": "\n"}'</code>
IgnoreHeaderRows	<p>Wenn dieser Wert auf 1 eingestellt ist, ignoriert AWS DMS die erste Zeile in einer CSV-Datei. Mit dem Wert 1 wird die Funktion aktiviert, mit 0 deaktiviert.</p> <p>Der Standardwert ist 0.</p> <p><code>'{"IgnoreHeaderRows": 1}'</code></p>
Rfc4180	<p>Wenn dieser Wert auf <code>true</code> oder <code>y</code> gesetzt ist, muss auf jedes führende doppelte Anführungszeichen ein endendes doppeltes Anführungszeichen folgen. Diese Formatierung entspricht RFC 4180. Wenn dieser Parameter auf <code>false</code> oder <code>n</code> gesetzt ist, werden Zeichenfolgen-Literale wie vorhanden in das Ziel kopiert. In diesem Fall signalisiert ein Trennzeichen (Zeile oder Spalte) das Ende des Feldes. Daher können Sie keine Trennzeichen als Teil der Zeichenfolge verwenden, da diese das Ende des Werts bezeichnen.</p> <p>Der Standardwert ist <code>true</code>.</p> <p>Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p><code>'{"Rfc4180": false}'</code></p>

Quelldatentypen für Amazon S3

Bei einer Datenmigration, die Amazon S3 als Quelle für AWS DMS verwendet, müssen die Daten aus Amazon S3 den AWS DMS-Datentypen zugeordnet werden. Weitere Informationen finden Sie unter [Definieren von externen Tabellen für Amazon S3 als Quelle für AWS DMS](#).

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

Die folgenden AWS DMS-Datentypen werden mit Amazon S3 als Quelle verwendet:

- BYTE – erfordert ColumnLength. Weitere Informationen finden Sie unter [Definieren von externen Tabellen für Amazon S3 als Quelle für AWS DMS](#).
- DATUM
- TIME
- DATETIME – weitere Informationen finden Sie im Beispiel für den Typ DATETIME unter [Definieren von externen Tabellen für Amazon S3 als Quelle für AWS DMS](#).
- INT1
- INT2
- INT4
- INT8
- NUMERIC – erfordert ColumnPrecision und ColumnScale. AWS DMS unterstützt die folgenden Höchstwerte:
 - ColumnPrecision: 38
 - ColumnScale: 31

Weitere Informationen finden Sie im Beispiel für den Typ NUMERIC unter [Definieren von externen Tabellen für Amazon S3 als Quelle für AWS DMS](#).

- REAL4
- REAL8
- STRING – erfordert ColumnLength. Weitere Informationen finden Sie unter [Definieren von externen Tabellen für Amazon S3 als Quelle für AWS DMS](#).
- UINT1
- UINT2
- UINT4
- UINT8
- BLOB
- CLOB
- BOOLEAN

Verwendung der IBM Db2 für Linux-, Unix-, Windows- und Amazon RDS-Datenbank (Db2 LUW) als Quelle für AWS DMS

Mit AWS Database Migration Service (DMS) können Sie Daten von einer IBM Db2 für Linux-, Unix-, Windows- und Amazon RDS-Datenbank (Db2 LUW) in jede unterstützte Zieldatenbank migrieren. AWS DMS

Informationen zu Versionen von Db2 unter Linux, Unix, Windows und RDS, die als Quelle AWS DMS unterstützt werden, finden Sie unter [Quellen für AWS DMS](#)

Sie können Secure Sockets Layer (SSL) verwenden, um Verbindungen zwischen Ihrem Db2 LUW-Endpunkt und der Replikations-Instance zu verschlüsseln. Weitere Informationen zur Verwendung von SSL mit einem Db2 LUW-Endpunkt finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).

Voraussetzungen für die Verwendung von Db2 LUW als Quelle für AWS DMS

Die folgenden Voraussetzungen müssen erfüllt sein, damit Sie eine Db2 LUW-Datenbank als Quelle verwenden können.

Zum Aktivieren einer fortlaufenden Replikation, auch als Change Data Capture (CDC) bezeichnet, gehen Sie wie folgt vor:

- Stellen Sie die Datenbank so ein, dass sie wiederherstellbar ist. AWS DMS Dazu müssen Änderungen erfasst werden. Eine Datenbank ist wiederherstellbar, wenn einer oder beide der Datenbank-Konfigurationsparameter LOGARCHMETH1 und LOGARCHMETH2 auf ON gesetzt sind.

Wenn Ihre Datenbank wiederherstellbar ist, AWS DMS können Sie bei Bedarf auf Db2 zugreifen. ARCHIVE LOG

- Stellen Sie sicher, dass die DB2-Transaktionsprotokolle verfügbar sind und über einen ausreichenden Aufbewahrungszeitraum verfügen, bis zu dem sie verarbeitet werden können. AWS DMS
- DB2 erfordert eine SYSADM- oder DBADM-Autorisierung zum Extrahieren von Transaktionsprotokoll-Datensätzen. Erteilen Sie dem Benutzerkonto die folgenden Berechtigungen:
 - SYSADM oder DBADM
 - DATAACCESS

Note

Für reine Volllast-Aufgaben benötigt das DMS-Benutzerkonto die Berechtigung DATAACCESS.

- Wenn Sie IBM DB2 für LUW Version 9.7 als Quelle verwenden, legen Sie das zusätzliche Verbindungsattribut (Extra Connection Attribute, ECA) `CurrentLSN` wie folgt fest:

`CurrentLSN=LSN`, wobei *LSN* eine Protokoll-Sequenznummer (Log Sequence Number, LSN) angibt, bei der die Replikation beginnen soll. Oder `CurrentLSN=scan`.

Einschränkungen bei der Verwendung von Db2 LUW als Quelle für AWS DMS

AWS DMS unterstützt keine geclusterten Datenbanken. Sie können jedoch für jeden der Endpunkte eines Clusters eine separate Db2 LUW definieren. Sie können beispielsweise eine Volllast-Migrationsaufgabe mit einem beliebigen Knoten im Cluster erstellen und dann für jeden Knoten separate Aufgaben erstellen.

AWS DMS unterstützt den `B00LEAN` Datentyp in Ihrer Db2-LUW-Quelldatenbank nicht.

Bei der Verwendung der fortlaufenden Replikation (CDC) gelten die folgenden Einschränkungen:

- Wenn eine Tabelle mit mehreren Partitionen gekürzt wird, entspricht die Anzahl der in der AWS DMS Konsole angezeigten DDL-Ereignisse der Anzahl der Partitionen. Der Grund hierfür ist, dass Db2 LUW eine separate DDL für jede Partition aufzeichnet.
- Die folgenden DDL-Aktionen werden für partitionierte Tabellen nicht unterstützt:
 - ALTER TABLE ADD PARTITION
 - ALTER TABLE DETACH PARTITION
 - ALTER TABLE ATTACH PARTITION
- AWS DMS unterstützt keine laufende Replikationsmigration von einer DB2 High Availability Disaster Recovery (HADR) -Standby-Instanz. Auf die Standby-Instance kann nicht zugegriffen werden.
- Der Datentyp `DECFLOAT` wird nicht unterstützt. Folglich werden Änderungen an `DECFLOAT`-Spalten während der fortlaufenden Replikation ignoriert.
- Die Anweisung `RENAME COLUMN` wird nicht unterstützt.

- Bei Aktualisierungen von MDC-Tabellen (Multi-Dimensional Clustering) wird jedes Update in der AWS DMS Konsole als INSERT + DELETE angezeigt.
- Wenn die Aufgabeneinstellung Include LOB columns in replication (LOB-Spalten in Replikation einschließen) nicht aktiviert ist, werden alle Tabellen mit LOB-Spalten während der fortlaufenden Replikation gesperrt.
- Bei Db2 LUW-Versionen 10.5 und höher werden Zeichenkettenspalten variabler Länge mit gespeicherten Daten ignoriert. out-of-row Diese Einschränkung gilt nur für Tabellen, die mit erweiterter Zeilengröße für Spalten mit Datentypen wie VARCHAR und VARGRAPHIC erstellt wurden. Verschieben Sie die Tabelle in einen Tabellenbereich mit einer höheren Seitengröße, um diese Einschränkung zu umgehen. Weitere Informationen finden Sie unter [What can I do if I want to change the pagesize of DB2 tablespaces?](#).
- Bei der laufenden Replikation unterstützt DMS nicht die Migration von Daten, die vom DB2-LOAD-Hilfsprogramm auf Seitenebene geladen wurden. Verwenden Sie stattdessen das IMPORT-Hilfsprogramm, das SQL-Inserts nutzt. Weitere Informationen finden Sie unter [Differences between the import and load utilities](#).
- Während der Ausführung einer Replikationsaufgabe erfasst DMS CREATE TABLE DDLs nur, wenn die Tabellen mit dem Attribut DATA CAPTURE CHANGE erstellt wurden.
- Bei der Verwendung des Db2-Datenbankpartitionsfeatures (DPF) gelten für DMS die folgenden Einschränkungen:
 - DMS kann in einer DPF-Umgebung keine Transaktionen zwischen Db2-Knoten koordinieren. Dies ist auf Einschränkungen innerhalb der IBM DB2READLOG API-Schnittstelle zurückzuführen. In DPF können sich Transaktionen über mehrere Db2-Knoten erstrecken, je nachdem, wie DB2 die Daten partitioniert. Daher muss Ihre DMS-Lösung Transaktionen von jedem Db2-Knoten unabhängig voneinander erfassen.
 - DMS kann lokale Transaktionen von jedem Db2-Knoten im DPF-Cluster erfassen, indem die Einstellung 1 auf mehrere DMS-Quellendpunkte connectNode gesetzt wird. Diese Konfiguration entspricht den logischen Knotennummern, die in der DB2-Serverkonfigurationsdatei definiert sind. db2nodes . cfg
 - Lokale Transaktionen auf einzelnen Db2-Knoten können Teil einer größeren, globalen Transaktion sein. DMS wendet jede lokale Transaktion unabhängig auf dem Ziel an, ohne dass dies mit Transaktionen auf anderen Db2-Knoten koordiniert wird. Diese unabhängige Verarbeitung kann zu Komplikationen führen, insbesondere wenn Zeilen zwischen Partitionen verschoben werden.
 - Wenn DMS von mehreren Db2-Knoten aus repliziert, gibt es keine Garantie für die korrekte Reihenfolge der Operationen auf dem Ziel, da DMS Operationen unabhängig für jeden Db2-

Knoten anwendet. Sie müssen sicherstellen, dass die Erfassung lokaler Transaktionen unabhängig von jedem Db2-Knoten für Ihren speziellen Anwendungsfall funktioniert.

- Bei der Migration aus einer DPF-Umgebung empfehlen wir, zuerst eine Volllastaufgabe ohne zwischengespeicherte Ereignisse auszuführen und dann ausschließlich CDC-Aufgaben auszuführen. Es wird empfohlen, eine Aufgabe pro DB2-Knoten auszuführen, beginnend mit dem Full Load-Startzeitstempel oder LRI (Log Record Identifier), den Sie mit der Endpunkteinstellung festgelegt haben. `StartFromContext` Informationen zur Bestimmung Ihres Replikationsstartpunkts [finden Sie in der IBM Support-Dokumentation unter Den LSN- oder LRI-Wert für den Replikationsstart](#) ermitteln.
- Wenn Sie bei einer laufenden Replikation (CDC) die Replikation ab einem bestimmten Zeitstempel starten möchten, müssen Sie das Verbindungsattribut `StartFromContext` auf den erforderlichen Zeitstempel setzen.
- Derzeit unterstützt DMS das Feature Db2 pureScale nicht. Dabei handelt es sich um eine Erweiterung von DB2 LUW, mit der Sie Ihre Datenbanklösung skalieren können.
- AWS DMS unterstützt CDC nicht, wenn Db2 für Amazon RDS als Quelle verwendet wird.

Endpunkteinstellungen bei Verwendung von Db2 LUW als Quelle für AWS DMS

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer Db2-LUW-Quelldatenbank verwenden. Sie geben die Einstellungen an, wenn Sie den Quellendpunkt mithilfe der AWS DMS Konsole oder mithilfe des `create-endpoint` Befehls in [AWS CLI](#), mit der `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax erstellen.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Db2 LUW als Quelle verwenden können.

Name	Beschreibung
<code>CurrentLSN</code>	Verwenden Sie für die laufende Replikation (CDC) <code>CurrentLSN</code> , um eine Protokoll-Sequenznummer (Log Sequence Number, LSN) anzugeben, ab der die Replikation beginnen soll.
<code>MaxKBytesPerRead</code>	Maximale Anzahl von Bytes pro Lesezugriff als NUMBER-Wert. Der Standardwert ist 64 KB.

Name	Beschreibung
SetDataCaptureChanges	Aktiviert die fortlaufende Replikation (CDC) als BOOLEAN-Wert. Der Standardwert ist „true“.

Name	Beschreibung
StartFromContext	<p>Verwenden Sie <code>StartFromContext</code> für die laufende Replikation (CDC), um die Untergrenze eines Protokolls anzugeben, bei der die Replikation beginnen soll. <code>StartFromContext</code> akzeptiert verschiedene Formen von Werten. Gültige Werte sind:</p> <ul style="list-style-type: none">• <code>timestamp</code> (UTC). Beispielsweise: <pre>{ "StartFromContext": "timestamp:2021-09-21T13:00:00" }</pre>• <code>NOW</code> Für IBM DB2 LUW Version 10.5 und höher startet <code>NOW</code> in Kombination mit <code>CurrentLSN: scan</code> die Aufgabe vom neuesten LSO aus. Beispielsweise: <pre>{ "CurrentLSN": "scan", "StartFromContext": "NOW" }</pre>• Ein spezifischer LRI. Beispielsweise: <pre>{ "StartFromContext": "0100000000000022C C000000000004FB13" }</pre> <p>Um den LRI/LSN-Bereich einer Protokolldatei zu ermitteln, führen Sie den Befehl <code>db2f1sn</code> wie im folgenden Beispiel gezeigt aus.</p> <pre>db2f1sn -db <i>SAMPLE</i> -lri range 2</pre> <p>Die Ausgabe sieht bei diesem Beispiel in etwa wie folgt aus.</p>

Name	Beschreibung
	<div data-bbox="690 210 1507 445" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>S0000002.LOG: has LRI range 0000000000000000100 00000000002254000000000004F9A6 to 00000000000000010000000000022CC00000000004 FB13</p> </div> <p>In dieser Ausgabe lautet die Protokolldatei S0000002. LOG und der StartFromContextLRI-Wert entspricht den 34 Byte am Ende des Bereichs.</p> <div data-bbox="690 651 1507 730" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>01000000000000022CC00000000004FB13</p> </div>

Quelldatentypen für IBM Db2 LUW

Datenmigration, die Db2 LUW als Quelle verwendet, AWS DMS unterstützt die meisten Db2 LUW-Datentypen. Die folgende Tabelle zeigt die Db2-LUW-Quelldatentypen, die bei der Verwendung unterstützt werden, AWS DMS sowie die Standardzuweisung von Datentypen. AWS DMS Weitere Informationen über die Db2 LUW-Datentypen finden Sie in der [Db2 LUW-Dokumentation](#).

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#)

Db2 LUW-Datentypen	AWS DMS Datentypen
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s)
FLOAT	REAL8

Db2 LUW-Datentypen	AWS DMS Datentypen
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Bei einer Genauigkeit von 16 wird REAL8 verwendet, bei einer Genauigkeit von 34 STRING.
GRAPHIC (n)	WSTRING für grafische Zeichenfolgen mit fester Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 127
VARGRAPHIC (n)	WSTRING für grafische Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 16.352
LONG VARGRAPHIC (n)	CLOB für grafische Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 16.352
CHARACTER (n)	STRING für Zeichenfolgen mit fester Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 255
VARCHAR (n)	STRING für Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 32.704
LONG VARCHAR (n)	CLOB für Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 32.704
CHAR (n) FOR BIT DATA	BYTES

Db2 LUW-Datentypen	AWS DMS Datentypen
VARCHAR (n) FOR BIT DATA	BYTES
LONG VARCHAR FOR BIT DATA	BYTES
DATUM	DATUM
TIME	TIME
TIMESTAMP (ZEITSTEMPEL)	DATETIME
BLOB (n)	BLOB Die maximale Länge beträgt 2.147.483.647 Byte.
CLOB (n)	CLOB Die maximale Länge beträgt 2.147.483.647 Byte.
DBCLOB (n)	CLOB Die maximale Länge beträgt 1.073.741.824 Doppelbyte-Zeichen.
XML	CLOB

Verwendung von Datenbanken in IBM Db2 für z/OS als Quelle für AWS DMS

Mit AWS Database Migration Service (AWS DMS) können Sie Daten von einer Datenbank in IBM Db2 für z/OS in jede unterstützte Zieldatenbank migrieren.

Informationen zu den Versionen von Db2 für z/OS, die AWS DMS als Quelle unterstützt, finden Sie unter [Quellen für AWS DMS](#).

Voraussetzungen für die Verwendung von Db2 für z/OS als Quelle für AWS DMS

Wenn Sie eine Datenbank in IBM Db2 für z/OS als Quelle für AWS DMS verwenden möchten, erteilen Sie dem Db2-für-z/OS-Benutzer, der in den Einstellungen zur Quellendpunktverbindung angegeben ist, die folgenden Berechtigungen.

```
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
```

Gewähren Sie außerdem `SELECT ON user defined`-Quellentabellen.

Ein AWS DMS-Quellendpunkt von IBM Db2 für z/OS verwendet für den Datenzugriff den IBM Data Server Driver für ODBC. Der Datenbankserver muss über eine gültige IBM-ODBC-Connect-Lizenz für DMS verfügen, um eine Verbindung zu diesem Endpunkt herstellen zu können.

Einschränkungen bei Verwendung von Db2 für z/OS als Quelle für AWS DMS

Die folgenden Einschränkungen gelten, wenn Sie eine Datenbank in IBM Db2 für z/OS als Quelle für AWS DMS verwenden:

- Es werden nur Replikationsaufgaben mit Volllast unterstützt. Change Data Capture (CDC) wird nicht unterstützt.
- Paralleles Laden wird nicht unterstützt.
- Die Datenvalidierung von Ansichten wird nicht unterstützt.
- Schema-, Tabellen- und Spaltennamen müssen in Tabellenzuordnungen für Transformationen auf Spalten-/Tabellenebene und Auswahlfilter auf Zeilenebene in GROSSBUCHSTABEN angegeben werden.

Quelldatentypen für IBM Db2 für z/OS

Datenmigrationen, die Db2 für z/OS als Quelle für AWS DMS verwenden, unterstützen die meisten Datentypen von Db2 für z/OS. In der folgenden Tabelle sind die Quelldatentypen von Db2 für z/OS, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung von AWS DMS-Datentypen aufgeführt.

Weitere Informationen zu den Datentypen von Db2 für z/OS finden Sie in der [Dokumentation zu IBM Db2 für z/OS](#).

Weitere Informationen zum Anzeigen des Datentyps, der im Ziel zugewiesen ist, finden Sie im Abschnitt für den Zielendpunkt, den Sie verwenden.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

Datentypen von Db2 für z/OS	AWS DMS-Datentypen
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s) Wenn in der DB2-Konfiguration ein Dezimaltrennzeichen auf ein Komma (,) gesetzt ist, konfigurieren Sie Replicate so, dass die DB2-Einstellung unterstützt wird.
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Bei einer Genauigkeit von 16 wird REAL8 verwendet, bei einer Genauigkeit von 34 STRING.
GRAPHIC (n)	Wenn $n \geq 127$, dann WSTRING für grafische Zeichenfolgen mit fester Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 127
VARGRAPHIC (n)	WSTRING für grafische Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-

Datentypen von Db2 für z/OS	AWS DMS-Datentypen
	Zeichen mit einer Länge größer 0 und kleiner oder gleich 16.352
LONG VARGRAPHIC (n)	CLOB für grafische Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 16.352
CHARACTER (n)	STRING für Zeichenfolgen mit fester Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 255
VARCHAR (n)	STRING für Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 32.704
LONG VARCHAR (n)	CLOB für Zeichenfolgen mit variabler Länge und dem Typ Doppelbyte-Zeichen mit einer Länge größer 0 und kleiner oder gleich 32.704
CHAR (n) FOR BIT DATA	BYTES
VARCHAR (n) FOR BIT DATA	BYTES
LONG VARCHAR FOR BIT DATA	BYTES
DATUM	DATUM
TIME	TIME
TIMESTAMP	DATETIME
BLOB (n)	BLOB Die maximale Länge beträgt 2.147.483.647 Byte.

Datentypen von Db2 für z/OS	AWS DMS-Datentypen
CLOB (n)	CLOB Die maximale Länge beträgt 2.147.483.647 Byte.
DBCLOB (n)	CLOB Die maximale Länge beträgt 1.073.741.824 Doppelbyte-Zeichen.
XML	CLOB
BINARY	BYTES
VARBINARY	BYTES
ROWID	BYTES. Weitere Informationen zum Arbeiten mit ROWID finden Sie in den folgenden Abschnitten:
TIMESTAMP WITH TIME ZONE	Nicht unterstützt

ROWID-Spalten werden standardmäßig migriert, wenn der Zieltabellen-Vorbereitungsmodus für die Aufgabe auf `DROP_AND_CREATE` (Standard) gesetzt ist. Bei der Datenüberprüfung werden diese Spalten ignoriert, da die Zeilen außerhalb der spezifischen Datenbank und Tabelle bedeutungslos sind. Sie können einen der folgenden Schritte ausführen, um die Migration dieser Spalten zu deaktivieren:

- Erstellen Sie die Zieltabelle ohne diese Spalten vorab. Setzen Sie dann den Zieltabellen-Vorbereitungsmodus der Aufgabe entweder auf `DO_NOTHING` oder `TRUNCATE_BEFORE_LOAD`. Sie können AWS Schema Conversion Tool (AWS SCT) verwenden, um die Zieltabelle ohne die Spalten vorab zu erstellen.
- Fügen Sie einer Aufgabe eine Tabellenzuordnungsregel hinzu, die diese Spalten herausfiltert, sodass sie ignoriert werden. Weitere Informationen finden Sie unter [Transformationsregeln und Aktionen](#).

EBCDIC-Sortierungen im Service PostgreSQL für AWS Mainframe Modernization

Das Programm AWS Mainframe Modernization unterstützt Sie bei der Modernisierung Ihrer Mainframe-Anwendungen auf von AWS verwaltete Laufzeitumgebungen. Es stellt Tools und Ressourcen bereit, die Sie bei der Planung und Implementierung von Migrations- und Modernisierungsprojekten unterstützen. Weitere Informationen zur Mainframe-Modernisierung und -Migration finden Sie unter [Mainframe Modernization with AWS](#).

Einige Datensätze von IBM Db2 für z/OS sind im Zeichensatz Extended Binary Coded Decimal Interchange (EBCDIC) codiert. Dieser Zeichensatz wurde entwickelt, bevor ASCII (American Standard Code for Information Interchange) allgemein verwendet wurde. Eine Codepage ordnet jedes Textzeichen den Zeichen in einem Zeichensatz zu. Eine herkömmliche Codepage enthält die Zuordnungsinformationen zwischen einem Codepoint und einer Zeichen-ID. Eine Zeichen-ID ist eine 8-Byte-Datenzeichenfolge. Ein Codepoint ist eine 8-Bit-Binärzahl, die ein Zeichen darstellt. Codepoints werden in der Regel als hexadezimale Abbildungen ihrer Binärwerte angezeigt.

Wenn Sie derzeit entweder die Komponente Micro Focus oder BluAge des Service Mainframe Modernization verwenden, müssen Sie AWS DMS mitteilen, dass bestimmte Codepoints geändert (übersetzt) werden sollen. Sie können die AWS DMS-Aufgabeneinstellungen verwenden, um die Änderungen vorzunehmen. Das folgende Beispiel zeigt, wie Sie die AWS DMS-Operation `CharacterSetSettings` verwenden, um die Änderungen in einer DMS-Aufgabeneinstellung zuzuordnen.

```
"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"}
  ]
}
```

Es gibt bereits einige EBCDIC-Sortierungen für PostgreSQL, die die erforderlichen Änderungen erfassen. Es werden verschiedene Codepages unterstützt. Die folgenden Abschnitte enthalten JSON-Beispiele dafür, was Sie für alle unterstützten Codepages ändern müssen. Sie können einfach die erforderliche JSON-Struktur, die Sie für Ihre DMS-Aufgabe benötigen, kopieren und einfügen.

Spezifische EBCDIC-Sortierungen für Micro Focus

Ändern Sie bei Micro Focus eine Teilmenge von Zeichen wie erforderlich für die folgenden Sortierungen.

```
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 Datenänderungen pro Sortierung bei Micro Focus:

en_us_cp1140m

Codeänderung:

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{"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

Codeänderung:

0000	0180
00B8	0160
00BC	0161
00BD	017D
00BE	017E
00A8	0152
00B4	0153
00A6	0178

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{"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

Codeänderung:

0000	0180
00A6	0160
00B8	0161
00BC	017D
00BD	017E
00BE	0152

```
00A8    0153
00B4    0178
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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

Codeänderung:

```
0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "SourceCharacterCodePoint": "0000", "TargetCharacterCodePoint": "0180" }
, { "SourceCharacterCodePoint": "00B8", "TargetCharacterCodePoint": "0160" }
, { "SourceCharacterCodePoint": "00BC", "TargetCharacterCodePoint": "0161" }
, { "SourceCharacterCodePoint": "00BD", "TargetCharacterCodePoint": "017D" }
, { "SourceCharacterCodePoint": "00BE", "TargetCharacterCodePoint": "017E" }
, { "SourceCharacterCodePoint": "00A8", "TargetCharacterCodePoint": "0152" }
, { "SourceCharacterCodePoint": "00B4", "TargetCharacterCodePoint": "0153" }
, { "SourceCharacterCodePoint": "00A6", "TargetCharacterCodePoint": "0178" }
```


en_us_cp1144m

Codeänderung:

```
0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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

Codeänderung:

```
0000    0180
00A6    0160
00B8    0161
00A8    017D
00BC    017E
00BD    0152
00BE    0153
00B4    0178
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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

Codeänderung:

0000	0180
00A6	0160
00B8	0161
00BC	017D
00BD	017E
00BE	0152
00A8	0153
00B4	0178

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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

Codeänderung:

0000	0180
------	------

```

00B8    0160
00A8    0161
00BC    017D
00BD    017E
00BE    0152
00B4    0153
00A6    0178

```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```

{"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

Codeänderung:

```

0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178

```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```

{"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"}
```

Spezifische EBCDIC-Sortierungen für BluAge

Ändern Sie bei BluAge alle folgenden niedrigen Werte und hohen Werte wie erforderlich. Diese Sortierungen sollten nur zur Unterstützung des Service Mainframe Migration BluAge verwendet werden.

```
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 Datenänderungen bei BluAge:

da-DK-cp277b und da-DK-cp1142b

Codeänderung:

```
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
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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 und de-DE-1141b

Codeänderung:

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
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "SourceCharacterCodePoint": "0180", "TargetCharacterCodePoint": "0180" }
, { "SourceCharacterCodePoint": "0001", "TargetCharacterCodePoint": "0181" }
, { "SourceCharacterCodePoint": "0002", "TargetCharacterCodePoint": "0182" }
, { "SourceCharacterCodePoint": "0003", "TargetCharacterCodePoint": "0183" }
, { "SourceCharacterCodePoint": "009C", "TargetCharacterCodePoint": "0184" }
, { "SourceCharacterCodePoint": "0009", "TargetCharacterCodePoint": "0185" }
, { "SourceCharacterCodePoint": "0086", "TargetCharacterCodePoint": "0186" }
, { "SourceCharacterCodePoint": "007F", "TargetCharacterCodePoint": "0187" }
, { "SourceCharacterCodePoint": "0097", "TargetCharacterCodePoint": "0188" }
, { "SourceCharacterCodePoint": "008D", "TargetCharacterCodePoint": "0189" }
, { "SourceCharacterCodePoint": "008E", "TargetCharacterCodePoint": "018A" }
, { "SourceCharacterCodePoint": "000B", "TargetCharacterCodePoint": "018B" }
, { "SourceCharacterCodePoint": "000C", "TargetCharacterCodePoint": "018C" }
, { "SourceCharacterCodePoint": "000D", "TargetCharacterCodePoint": "018D" }
```

```
, {"SourceCharacterCodePoint": "000E", "TargetCharacterCodePoint": "018E"}
, {"SourceCharacterCodePoint": "000F", "TargetCharacterCodePoint": "018F"}
, {"SourceCharacterCodePoint": "0010", "TargetCharacterCodePoint": "0190"}
, {"SourceCharacterCodePoint": "0011", "TargetCharacterCodePoint": "0191"}
, {"SourceCharacterCodePoint": "0012", "TargetCharacterCodePoint": "0192"}
, {"SourceCharacterCodePoint": "0013", "TargetCharacterCodePoint": "0193"}
, {"SourceCharacterCodePoint": "009D", "TargetCharacterCodePoint": "0194"}
, {"SourceCharacterCodePoint": "0085", "TargetCharacterCodePoint": "0195"}
, {"SourceCharacterCodePoint": "0008", "TargetCharacterCodePoint": "0196"}
, {"SourceCharacterCodePoint": "0087", "TargetCharacterCodePoint": "0197"}
, {"SourceCharacterCodePoint": "0018", "TargetCharacterCodePoint": "0198"}
, {"SourceCharacterCodePoint": "0019", "TargetCharacterCodePoint": "0199"}
, {"SourceCharacterCodePoint": "0092", "TargetCharacterCodePoint": "019A"}
, {"SourceCharacterCodePoint": "008F", "TargetCharacterCodePoint": "019B"}
, {"SourceCharacterCodePoint": "001C", "TargetCharacterCodePoint": "019C"}
, {"SourceCharacterCodePoint": "001D", "TargetCharacterCodePoint": "019D"}
, {"SourceCharacterCodePoint": "001E", "TargetCharacterCodePoint": "019E"}
, {"SourceCharacterCodePoint": "001F", "TargetCharacterCodePoint": "019F"}
, {"SourceCharacterCodePoint": "0080", "TargetCharacterCodePoint": "01A0"}
, {"SourceCharacterCodePoint": "0081", "TargetCharacterCodePoint": "01A1"}
, {"SourceCharacterCodePoint": "0082", "TargetCharacterCodePoint": "01A2"}
, {"SourceCharacterCodePoint": "0083", "TargetCharacterCodePoint": "01A3"}
, {"SourceCharacterCodePoint": "0084", "TargetCharacterCodePoint": "01A4"}
, {"SourceCharacterCodePoint": "000A", "TargetCharacterCodePoint": "01A5"}
, {"SourceCharacterCodePoint": "0017", "TargetCharacterCodePoint": "01A6"}
, {"SourceCharacterCodePoint": "001B", "TargetCharacterCodePoint": "01A7"}
, {"SourceCharacterCodePoint": "0088", "TargetCharacterCodePoint": "01A8"}
, {"SourceCharacterCodePoint": "0089", "TargetCharacterCodePoint": "01A9"}
, {"SourceCharacterCodePoint": "008A", "TargetCharacterCodePoint": "01AA"}
, {"SourceCharacterCodePoint": "008B", "TargetCharacterCodePoint": "01AB"}
, {"SourceCharacterCodePoint": "008C", "TargetCharacterCodePoint": "01AC"}
, {"SourceCharacterCodePoint": "0005", "TargetCharacterCodePoint": "01AD"}
, {"SourceCharacterCodePoint": "0006", "TargetCharacterCodePoint": "01AE"}
, {"SourceCharacterCodePoint": "0007", "TargetCharacterCodePoint": "01AF"}
, {"SourceCharacterCodePoint": "0090", "TargetCharacterCodePoint": "01B0"}
, {"SourceCharacterCodePoint": "0091", "TargetCharacterCodePoint": "01B1"}
, {"SourceCharacterCodePoint": "0016", "TargetCharacterCodePoint": "01B2"}
, {"SourceCharacterCodePoint": "0093", "TargetCharacterCodePoint": "01B3"}
, {"SourceCharacterCodePoint": "0094", "TargetCharacterCodePoint": "01B4"}
, {"SourceCharacterCodePoint": "0095", "TargetCharacterCodePoint": "01B5"}
, {"SourceCharacterCodePoint": "0096", "TargetCharacterCodePoint": "01B6"}
, {"SourceCharacterCodePoint": "0004", "TargetCharacterCodePoint": "01B7"}
, {"SourceCharacterCodePoint": "0098", "TargetCharacterCodePoint": "01B8"}
, {"SourceCharacterCodePoint": "0099", "TargetCharacterCodePoint": "01B9"}
```

```
,{"SourceCharacterCodePoint": "009A", "TargetCharacterCodePoint": "01BA"}  
,{"SourceCharacterCodePoint": "009B", "TargetCharacterCodePoint": "01BB"}  
,{"SourceCharacterCodePoint": "0014", "TargetCharacterCodePoint": "01BC"}  
,{"SourceCharacterCodePoint": "0015", "TargetCharacterCodePoint": "01BD"}  
,{"SourceCharacterCodePoint": "009E", "TargetCharacterCodePoint": "01BE"}  
,{"SourceCharacterCodePoint": "001A", "TargetCharacterCodePoint": "01BF"}  
,{"SourceCharacterCodePoint": "009F", "TargetCharacterCodePoint": "027F"}
```

en-GB-285b und en-GB-1146b

Codeänderung:

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

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{"SourceCharacterCodePoint": "0180", "TargetCharacterCodePoint": "0180"}  
, {"SourceCharacterCodePoint": "0001", "TargetCharacterCodePoint": "0181"}  
, {"SourceCharacterCodePoint": "0002", "TargetCharacterCodePoint": "0182"}  
, {"SourceCharacterCodePoint": "0003", "TargetCharacterCodePoint": "0183"}
```

```
, {"SourceCharacterCodePoint": "009C", "TargetCharacterCodePoint": "0184"}
, {"SourceCharacterCodePoint": "0009", "TargetCharacterCodePoint": "0185"}
, {"SourceCharacterCodePoint": "0086", "TargetCharacterCodePoint": "0186"}
, {"SourceCharacterCodePoint": "007F", "TargetCharacterCodePoint": "0187"}
, {"SourceCharacterCodePoint": "0097", "TargetCharacterCodePoint": "0188"}
, {"SourceCharacterCodePoint": "008D", "TargetCharacterCodePoint": "0189"}
, {"SourceCharacterCodePoint": "008E", "TargetCharacterCodePoint": "018A"}
, {"SourceCharacterCodePoint": "000B", "TargetCharacterCodePoint": "018B"}
, {"SourceCharacterCodePoint": "000C", "TargetCharacterCodePoint": "018C"}
, {"SourceCharacterCodePoint": "000D", "TargetCharacterCodePoint": "018D"}
, {"SourceCharacterCodePoint": "000E", "TargetCharacterCodePoint": "018E"}
, {"SourceCharacterCodePoint": "000F", "TargetCharacterCodePoint": "018F"}
, {"SourceCharacterCodePoint": "0010", "TargetCharacterCodePoint": "0190"}
, {"SourceCharacterCodePoint": "0011", "TargetCharacterCodePoint": "0191"}
, {"SourceCharacterCodePoint": "0012", "TargetCharacterCodePoint": "0192"}
, {"SourceCharacterCodePoint": "0013", "TargetCharacterCodePoint": "0193"}
, {"SourceCharacterCodePoint": "009D", "TargetCharacterCodePoint": "0194"}
, {"SourceCharacterCodePoint": "0085", "TargetCharacterCodePoint": "0195"}
, {"SourceCharacterCodePoint": "0008", "TargetCharacterCodePoint": "0196"}
, {"SourceCharacterCodePoint": "0087", "TargetCharacterCodePoint": "0197"}
, {"SourceCharacterCodePoint": "0018", "TargetCharacterCodePoint": "0198"}
, {"SourceCharacterCodePoint": "0019", "TargetCharacterCodePoint": "0199"}
, {"SourceCharacterCodePoint": "0092", "TargetCharacterCodePoint": "019A"}
, {"SourceCharacterCodePoint": "008F", "TargetCharacterCodePoint": "019B"}
, {"SourceCharacterCodePoint": "001C", "TargetCharacterCodePoint": "019C"}
, {"SourceCharacterCodePoint": "001D", "TargetCharacterCodePoint": "019D"}
, {"SourceCharacterCodePoint": "001E", "TargetCharacterCodePoint": "019E"}
, {"SourceCharacterCodePoint": "001F", "TargetCharacterCodePoint": "019F"}
, {"SourceCharacterCodePoint": "0080", "TargetCharacterCodePoint": "01A0"}
, {"SourceCharacterCodePoint": "0081", "TargetCharacterCodePoint": "01A1"}
, {"SourceCharacterCodePoint": "0082", "TargetCharacterCodePoint": "01A2"}
, {"SourceCharacterCodePoint": "0083", "TargetCharacterCodePoint": "01A3"}
, {"SourceCharacterCodePoint": "0084", "TargetCharacterCodePoint": "01A4"}
, {"SourceCharacterCodePoint": "000A", "TargetCharacterCodePoint": "01A5"}
, {"SourceCharacterCodePoint": "0017", "TargetCharacterCodePoint": "01A6"}
, {"SourceCharacterCodePoint": "001B", "TargetCharacterCodePoint": "01A7"}
, {"SourceCharacterCodePoint": "0088", "TargetCharacterCodePoint": "01A8"}
, {"SourceCharacterCodePoint": "0089", "TargetCharacterCodePoint": "01A9"}
, {"SourceCharacterCodePoint": "008A", "TargetCharacterCodePoint": "01AA"}
, {"SourceCharacterCodePoint": "008B", "TargetCharacterCodePoint": "01AB"}
, {"SourceCharacterCodePoint": "008C", "TargetCharacterCodePoint": "01AC"}
, {"SourceCharacterCodePoint": "0005", "TargetCharacterCodePoint": "01AD"}
, {"SourceCharacterCodePoint": "0006", "TargetCharacterCodePoint": "01AE"}
, {"SourceCharacterCodePoint": "0007", "TargetCharacterCodePoint": "01AF"}
```

```
,{"SourceCharacterCodePoint": "0090", "TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091", "TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016", "TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093", "TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094", "TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095", "TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096", "TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004", "TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098", "TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099", "TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A", "TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B", "TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014", "TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015", "TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E", "TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A", "TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F", "TargetCharacterCodePoint": "027F"}
```

en-us-037b und en-us-1140b

Codeänderung:

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

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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 und es-ES-1145b

Codeänderung:

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

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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 und fi-FI-1143b

Codeänderung:

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

```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```

{"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 und fr-FR-1147b

Codeänderung:

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
```

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{"SourceCharacterCodePoint": "0180", "TargetCharacterCodePoint": "0180"}
, {"SourceCharacterCodePoint": "0001", "TargetCharacterCodePoint": "0181"}
, {"SourceCharacterCodePoint": "0002", "TargetCharacterCodePoint": "0182"}
, {"SourceCharacterCodePoint": "0003", "TargetCharacterCodePoint": "0183"}
, {"SourceCharacterCodePoint": "009C", "TargetCharacterCodePoint": "0184"}
, {"SourceCharacterCodePoint": "0009", "TargetCharacterCodePoint": "0185"}
, {"SourceCharacterCodePoint": "0086", "TargetCharacterCodePoint": "0186"}
, {"SourceCharacterCodePoint": "007F", "TargetCharacterCodePoint": "0187"}
```

```
, {"SourceCharacterCodePoint": "0097", "TargetCharacterCodePoint": "0188"}
, {"SourceCharacterCodePoint": "008D", "TargetCharacterCodePoint": "0189"}
, {"SourceCharacterCodePoint": "008E", "TargetCharacterCodePoint": "018A"}
, {"SourceCharacterCodePoint": "000B", "TargetCharacterCodePoint": "018B"}
, {"SourceCharacterCodePoint": "000C", "TargetCharacterCodePoint": "018C"}
, {"SourceCharacterCodePoint": "000D", "TargetCharacterCodePoint": "018D"}
, {"SourceCharacterCodePoint": "000E", "TargetCharacterCodePoint": "018E"}
, {"SourceCharacterCodePoint": "000F", "TargetCharacterCodePoint": "018F"}
, {"SourceCharacterCodePoint": "0010", "TargetCharacterCodePoint": "0190"}
, {"SourceCharacterCodePoint": "0011", "TargetCharacterCodePoint": "0191"}
, {"SourceCharacterCodePoint": "0012", "TargetCharacterCodePoint": "0192"}
, {"SourceCharacterCodePoint": "0013", "TargetCharacterCodePoint": "0193"}
, {"SourceCharacterCodePoint": "009D", "TargetCharacterCodePoint": "0194"}
, {"SourceCharacterCodePoint": "0085", "TargetCharacterCodePoint": "0195"}
, {"SourceCharacterCodePoint": "0008", "TargetCharacterCodePoint": "0196"}
, {"SourceCharacterCodePoint": "0087", "TargetCharacterCodePoint": "0197"}
, {"SourceCharacterCodePoint": "0018", "TargetCharacterCodePoint": "0198"}
, {"SourceCharacterCodePoint": "0019", "TargetCharacterCodePoint": "0199"}
, {"SourceCharacterCodePoint": "0092", "TargetCharacterCodePoint": "019A"}
, {"SourceCharacterCodePoint": "008F", "TargetCharacterCodePoint": "019B"}
, {"SourceCharacterCodePoint": "001C", "TargetCharacterCodePoint": "019C"}
, {"SourceCharacterCodePoint": "001D", "TargetCharacterCodePoint": "019D"}
, {"SourceCharacterCodePoint": "001E", "TargetCharacterCodePoint": "019E"}
, {"SourceCharacterCodePoint": "001F", "TargetCharacterCodePoint": "019F"}
, {"SourceCharacterCodePoint": "0080", "TargetCharacterCodePoint": "01A0"}
, {"SourceCharacterCodePoint": "0081", "TargetCharacterCodePoint": "01A1"}
, {"SourceCharacterCodePoint": "0082", "TargetCharacterCodePoint": "01A2"}
, {"SourceCharacterCodePoint": "0083", "TargetCharacterCodePoint": "01A3"}
, {"SourceCharacterCodePoint": "0084", "TargetCharacterCodePoint": "01A4"}
, {"SourceCharacterCodePoint": "000A", "TargetCharacterCodePoint": "01A5"}
, {"SourceCharacterCodePoint": "0017", "TargetCharacterCodePoint": "01A6"}
, {"SourceCharacterCodePoint": "001B", "TargetCharacterCodePoint": "01A7"}
, {"SourceCharacterCodePoint": "0088", "TargetCharacterCodePoint": "01A8"}
, {"SourceCharacterCodePoint": "0089", "TargetCharacterCodePoint": "01A9"}
, {"SourceCharacterCodePoint": "008A", "TargetCharacterCodePoint": "01AA"}
, {"SourceCharacterCodePoint": "008B", "TargetCharacterCodePoint": "01AB"}
, {"SourceCharacterCodePoint": "008C", "TargetCharacterCodePoint": "01AC"}
, {"SourceCharacterCodePoint": "0005", "TargetCharacterCodePoint": "01AD"}
, {"SourceCharacterCodePoint": "0006", "TargetCharacterCodePoint": "01AE"}
, {"SourceCharacterCodePoint": "0007", "TargetCharacterCodePoint": "01AF"}
, {"SourceCharacterCodePoint": "0090", "TargetCharacterCodePoint": "01B0"}
, {"SourceCharacterCodePoint": "0091", "TargetCharacterCodePoint": "01B1"}
, {"SourceCharacterCodePoint": "0016", "TargetCharacterCodePoint": "01B2"}
, {"SourceCharacterCodePoint": "0093", "TargetCharacterCodePoint": "01B3"}
```

```
, {"SourceCharacterCodePoint": "0094", "TargetCharacterCodePoint": "01B4"}  
, {"SourceCharacterCodePoint": "0095", "TargetCharacterCodePoint": "01B5"}  
, {"SourceCharacterCodePoint": "0096", "TargetCharacterCodePoint": "01B6"}  
, {"SourceCharacterCodePoint": "0004", "TargetCharacterCodePoint": "01B7"}  
, {"SourceCharacterCodePoint": "0098", "TargetCharacterCodePoint": "01B8"}  
, {"SourceCharacterCodePoint": "0099", "TargetCharacterCodePoint": "01B9"}  
, {"SourceCharacterCodePoint": "009A", "TargetCharacterCodePoint": "01BA"}  
, {"SourceCharacterCodePoint": "009B", "TargetCharacterCodePoint": "01BB"}  
, {"SourceCharacterCodePoint": "0014", "TargetCharacterCodePoint": "01BC"}  
, {"SourceCharacterCodePoint": "0015", "TargetCharacterCodePoint": "01BD"}  
, {"SourceCharacterCodePoint": "009E", "TargetCharacterCodePoint": "01BE"}  
, {"SourceCharacterCodePoint": "001A", "TargetCharacterCodePoint": "01BF"}  
, {"SourceCharacterCodePoint": "009F", "TargetCharacterCodePoint": "027F"}
```

it-IT-280b und it-IT-1144b

Codeänderung:

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

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "SourceCharacterCodePoint": "0180", "TargetCharacterCodePoint": "0180" }
, { "SourceCharacterCodePoint": "0001", "TargetCharacterCodePoint": "0181" }
, { "SourceCharacterCodePoint": "0002", "TargetCharacterCodePoint": "0182" }
, { "SourceCharacterCodePoint": "0003", "TargetCharacterCodePoint": "0183" }
, { "SourceCharacterCodePoint": "009C", "TargetCharacterCodePoint": "0184" }
, { "SourceCharacterCodePoint": "0009", "TargetCharacterCodePoint": "0185" }
, { "SourceCharacterCodePoint": "0086", "TargetCharacterCodePoint": "0186" }
, { "SourceCharacterCodePoint": "007F", "TargetCharacterCodePoint": "0187" }
, { "SourceCharacterCodePoint": "0097", "TargetCharacterCodePoint": "0188" }
, { "SourceCharacterCodePoint": "008D", "TargetCharacterCodePoint": "0189" }
, { "SourceCharacterCodePoint": "008E", "TargetCharacterCodePoint": "018A" }
, { "SourceCharacterCodePoint": "000B", "TargetCharacterCodePoint": "018B" }
, { "SourceCharacterCodePoint": "000C", "TargetCharacterCodePoint": "018C" }
, { "SourceCharacterCodePoint": "000D", "TargetCharacterCodePoint": "018D" }
, { "SourceCharacterCodePoint": "000E", "TargetCharacterCodePoint": "018E" }
, { "SourceCharacterCodePoint": "000F", "TargetCharacterCodePoint": "018F" }
, { "SourceCharacterCodePoint": "0010", "TargetCharacterCodePoint": "0190" }
, { "SourceCharacterCodePoint": "0011", "TargetCharacterCodePoint": "0191" }
, { "SourceCharacterCodePoint": "0012", "TargetCharacterCodePoint": "0192" }
, { "SourceCharacterCodePoint": "0013", "TargetCharacterCodePoint": "0193" }
, { "SourceCharacterCodePoint": "009D", "TargetCharacterCodePoint": "0194" }
, { "SourceCharacterCodePoint": "0085", "TargetCharacterCodePoint": "0195" }
, { "SourceCharacterCodePoint": "0008", "TargetCharacterCodePoint": "0196" }
, { "SourceCharacterCodePoint": "0087", "TargetCharacterCodePoint": "0197" }
, { "SourceCharacterCodePoint": "0018", "TargetCharacterCodePoint": "0198" }
, { "SourceCharacterCodePoint": "0019", "TargetCharacterCodePoint": "0199" }
, { "SourceCharacterCodePoint": "0092", "TargetCharacterCodePoint": "019A" }
, { "SourceCharacterCodePoint": "008F", "TargetCharacterCodePoint": "019B" }
, { "SourceCharacterCodePoint": "001C", "TargetCharacterCodePoint": "019C" }
, { "SourceCharacterCodePoint": "001D", "TargetCharacterCodePoint": "019D" }
, { "SourceCharacterCodePoint": "001E", "TargetCharacterCodePoint": "019E" }
, { "SourceCharacterCodePoint": "001F", "TargetCharacterCodePoint": "019F" }
, { "SourceCharacterCodePoint": "0080", "TargetCharacterCodePoint": "01A0" }
, { "SourceCharacterCodePoint": "0081", "TargetCharacterCodePoint": "01A1" }
, { "SourceCharacterCodePoint": "0082", "TargetCharacterCodePoint": "01A2" }
, { "SourceCharacterCodePoint": "0083", "TargetCharacterCodePoint": "01A3" }
, { "SourceCharacterCodePoint": "0084", "TargetCharacterCodePoint": "01A4" }
, { "SourceCharacterCodePoint": "000A", "TargetCharacterCodePoint": "01A5" }
, { "SourceCharacterCodePoint": "0017", "TargetCharacterCodePoint": "01A6" }
, { "SourceCharacterCodePoint": "001B", "TargetCharacterCodePoint": "01A7" }
, { "SourceCharacterCodePoint": "0088", "TargetCharacterCodePoint": "01A8" }
, { "SourceCharacterCodePoint": "0089", "TargetCharacterCodePoint": "01A9" }
```

```
,{"SourceCharacterCodePoint": "008A", "TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B", "TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C", "TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005", "TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006", "TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007", "TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090", "TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091", "TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016", "TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093", "TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094", "TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095", "TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096", "TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004", "TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098", "TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099", "TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A", "TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B", "TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014", "TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015", "TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E", "TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A", "TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F", "TargetCharacterCodePoint": "027F"}
```

nl-BE-500b und nl-BE-1148b

Codeänderung:

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

Entsprechende Eingabebezuordnung für eine AWS DMS-Aufgabe:

```
{ "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"}
```

Ziele für die Datenmigration

AWS Database Migration Service (AWS DMS) kann viele der beliebtesten Datenbanken als Ziel für die Datenreplikation verwenden. Das Ziel kann sich in einer Amazon Elastic Compute Cloud-Instance (Amazon EC2), einer Amazon Relational Database Service-Instance (Amazon RDS) oder einer lokalen Datenbank befinden.

Eine umfassende Liste gültiger Ziele finden Sie unter [Ziele für AWS DMS](#).

 Note

AWS DMS unterstützt für die folgenden Ziel-Endpunkttypen keine Migration über AWS-Regionen hinweg:

- Amazon DynamoDB
- Amazon OpenSearch Service
- Amazon Kinesis Data Streams

Themen

- [Verwenden einer Oracle-Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden einer Microsoft SQL Server-Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden einer PostgreSQL-Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden einer MySQL-kompatiblen Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden einer SAP ASE-Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden von Amazon S3 als Ziel für AWS Database Migration Service](#)
- [Verwenden einer Amazon-DynamoDB-Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden von Amazon Kinesis Data Streams als Ziel für AWS Database Migration Service](#)
- [Verwenden von Apache Kafka als Ziel für AWS Database Migration Service](#)
- [Verwenden eines Clusters in Amazon OpenSearch Service als Ziel für AWS Database Migration Service](#)
- [Verwenden von Amazon DocumentDB als Ziel für den AWS Database Migration Service](#)
- [Verwenden von Amazon Neptune als Ziel für AWS Database Migration Service](#)
- [Verwenden von Redis als Ziel für AWS Database Migration Service](#)
- [Verwenden von Babelfish als Ziel für AWS Database Migration Service](#)
- [Verwenden von Amazon Timestream als Ziel für AWS Database Migration Service](#)
- [Verwendung von Amazon RDS für Db2 und IBM Db2 LUW als Ziel für AWS DMS](#)

Verwenden einer Oracle-Datenbank als Ziel für AWS Database Migration Service

Sie können Daten entweder aus einer anderen Oracle-Datenbank oder aus einer der anderen unterstützten Datenbanken zu Oracle-Datenbankzielen migrieren. AWS DMS Sie können Secure Sockets Layer (SSL) verwenden, um Verbindungen zwischen Ihrem Oracle-Endpunkt und der Replikations-Instance zu verschlüsseln. Weitere Hinweise zur Verwendung von SSL mit einem Oracle-Endpunkt finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#). AWS DMS unterstützt auch die Verwendung von Oracle Transparent Data Encryption (TDE) zur Verschlüsselung ruhender Daten in der Zieldatenbank, da Oracle TDE zum Schreiben in die Datenbank weder einen Verschlüsselungsschlüssel noch ein Kennwort benötigt.

Hinweise zu Versionen von Oracle, die als Ziel AWS DMS unterstützen, finden Sie unter [Ziele für AWS DMS](#)

Wenn Sie Oracle als Ziel verwenden, gehen wir davon aus, dass die Daten in das Schema oder den Benutzer migriert werden sollen, das bzw. der für die Zielverbindung verwendet wird. Wenn Sie Daten in ein anderes Schema migrieren möchten, verwenden Sie dazu eine Schematransformation. Nehmen wir an, Ihr Zielendpunkt stellt eine Verbindung mit dem Benutzer RDSMASTER her und Sie möchten vom Benutzer PERFDATA1 zu PERFDATA2 migrieren. Erstellen Sie in diesem Fall eine Transformation, die dem folgenden Beispiel ähnelt.

```
{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "rename",
  "rule-target": "schema",
  "object-locator": {
    "schema-name": "PERFDATA1"
  },
  "value": "PERFDATA2"
}
```

Wenn Sie Oracle als Ziel verwenden, AWS DMS migriert alle Tabellen und Indizes in Standardtabellen- und Index-Tablespaces im Ziel. Wenn Sie Tabellen und Indizes in verschiedene Tabellen- und Index-Tablespaces migrieren möchten, verwenden Sie hierfür eine Tablespace-

Transformation. Angenommen, Sie verfügen über eine Reihe von Tabellen im INVENTORY-Schema, die einigen Tablespaces in der Oracle-Quelldatenbank zugeordnet sind. Für die Migration möchten Sie all diese Tabellen zu einem einzelnen INVENTORYSPACE-Tabellenraum in der Zieldatenbank zuweisen. Erstellen Sie in diesem Fall eine Transformation, die dem folgenden Beispiel ähnelt.

```
{
  "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"
}
```

Weitere Informationen zu Umwandlungen finden Sie unter [Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON](#).

Wenn Oracle sowohl Quelle als auch Ziel ist, können Sie vorhandene Tabellen- oder Index-Tablespace-Zuweisungen beibehalten, indem Sie das zusätzliche Verbindungsattribut für Oracle Source, `enableHomogenousTablespace=true`, festlegen. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).

Weitere Informationen zur Arbeit mit Oracle-Datenbanken als Ziel für finden Sie in den AWS DMS folgenden Abschnitten:

Themen

- [Einschränkungen von Oracle als Ziel für AWS Database Migration Service](#)
- [Benutzerkontoberechtigungen, die bei der Verwendung von Oracle als Ziel erforderlich sind](#)
- [Konfiguration einer Oracle-Datenbank als Ziel für AWS Database Migration Service](#)
- [Endpunkteinstellungen bei Verwendung von Oracle als Ziel für AWS DMS](#)
- [Zieldatentypen für Oracle](#)

Einschränkungen von Oracle als Ziel für AWS Database Migration Service

Bei der Verwendung von Oracle als Ziel für die Datenmigration gelten folgende Einschränkungen:

- AWS DMS erstellt kein Schema in der Oracle-Zieldatenbank. Sie müssen gewünschte Schemata selbst in der Oracle-Zieldatenbank erstellen. Der Schemaname muss für das Oracle-Ziel bereits vorhanden sein. Tabellen aus dem Quellschema werden in den Benutzer oder das Schema importiert, das für die Verbindung mit der Zielinstanz AWS DMS verwendet wird. Um mehrere Schemata zu migrieren, können Sie mehrere Replikationsaufgaben erstellen. Sie können Daten auch in verschiedene Schemata auf einem Ziel migrieren. Dazu müssen Sie Schematransformationsregeln für die AWS DMS Tabellenzuordnungen verwenden.
- AWS DMS unterstützt die `Use direct path full load` Option für Tabellen mit `INDEXTYPE CONTEXT` nicht. Um dieses Problem zu umgehen, können Sie den Array-Ladevorgang ausführen.
- Mit der Batch-optimierten Option „Apply (Anwenden)“ wird beim Laden in die Nettoänderungstabelle ein direkter Pfad verwendet, der den XML-Typ nicht unterstützt. Um dieses Problem zu umgehen, können Sie den transaktionalen Modus „Apply (Anwenden)“ verwenden.
- Leere Zeichenfolgen, die aus Quelldatenbanken migriert werden, können vom Oracle-Ziel unterschiedlich behandelt (z. B. in eine Leerzeichenfolge konvertiert) werden. Dies kann dazu führen, dass bei der AWS DMS Überprüfung eine Nichtübereinstimmung gemeldet wird.
- Sie können die Gesamtzahl der Spalten pro Tabelle, die im stapeloptimierten Anwendungsmodus unterstützt werden, mit der folgenden Formel ausdrücken:

$$2 * \text{columns_in_original_table} + \text{columns_in_primary_key} \leq 999$$

Wenn die Originaltabelle beispielsweise 25 Spalten hat und ihr Primärschlüssel aus 5 Spalten besteht, beträgt die Gesamtzahl der Spalten 55. Wenn eine Tabelle die unterstützte Anzahl von Spalten überschreitet, werden alle Änderungen im one-by-one Modus übernommen.

- AWS DMS unterstützt Autonomous DB auf Oracle Cloud Infrastructure (OCI) nicht.

Benutzerkontoberechtigungen, die bei der Verwendung von Oracle als Ziel erforderlich sind

Um ein Oracle-Ziel in einer AWS Database Migration Service Aufgabe zu verwenden, gewähren Sie die folgenden Rechte in der Oracle-Datenbank. Sie erteilen diese Berechtigungen dem Benutzerkonto, das in den Oracle-Datenbankdefinitionen für AWS DMS angegeben wurde.

- SELECT ANY TRANSACTION
- SELECT on V\$NLS_PARAMETERS
- SELECT on V\$TIMEZONE_NAMES
- SELECT on ALL_INDEXES
- SELECT on ALL_OBJECTS
- SELECT on DBA_OBJECTS
- SELECT on ALL_TABLES
- SELECT on ALL_USERS
- SELECT on ALL_CATALOG
- SELECT on ALL_CONSTRAINTS
- SELECT on ALL_CONS_COLUMNS
- SELECT on ALL_TAB_COLS
- SELECT on ALL_IND_COLUMNS
- DROP ANY TABLE
- SELECT ANY TABLE
- INSERT ANY TABLE
- UPDATE ANY TABLE
- CREATE ANY VIEW
- DROP ANY VIEW
- CREATE ANY PROCEDURE
- ALTER ANY PROCEDURE
- DROP ANY PROCEDURE
- CREATE ANY SEQUENCE
- ALTER ANY SEQUENCE
- DROP ANY SEQUENCE
- DELETE ANY TABLE

Erteilen Sie für die folgenden Anforderungen diese zusätzlichen Berechtigungen:

- Um eine bestimmte Tabellenliste zu verwenden, weisen Sie den replizierten Tabellen die Berechtigungen SELECT und auch ALTER zu.

- Damit ein Benutzer in einem Standardtabellenbereich eine Tabelle erstellen kann, erteilen Sie die Berechtigung `GRANT UNLIMITED TABLESPACE`.
- Für die Anmeldung gewähren Sie die Berechtigung `CREATE SESSION`.
- Wenn Sie einen direkten Pfad verwenden (Standard bei Volllast), `GRANT LOCK ANY TABLE` to *dms_user*;
- Wenn das Schema bei Verwendung des Tabellenvorbereitungsmodus „DROP and CREATE“ anders ist, `GRANT CREATE ANY INDEX` to *dms_user*;
- Bei einigen Szenarien mit Volllast können Sie die Option „DROP und CREATE Tabelle“ oder „TRUNCATE vor dem Laden“ wählen, wobei sich ein Zieltabellenschema vom DMS-Benutzer unterscheidet. Erteilen Sie in diesem Fall die Berechtigung `DROP ANY TABLE`.
- Um Änderungen in Änderungstabellen oder einer Audit-Tabelle zu speichern, in der sich das Zieltabellenschema vom DMS-Benutzer unterscheidet, gewähren Sie `CREATE ANY TABLE` und `CREATE ANY INDEX`.

Für AWS Database Migration Service die Zieldatenbank sind Leserechte erforderlich

Dem AWS DMS Benutzerkonto müssen Leseberechtigungen für die folgenden DBA-Tabellen gewährt werden:

- `SELECT` on `DBA_USERS`
- `SELECT` on `DBA_TAB_PRIVS`
- `SELECT` on `DBA_OBJECTS`
- `SELECT` on `DBA_SYNONYMS`
- `SELECT` on `DBA_SEQUENCES`
- `SELECT` on `DBA_TYPES`
- `SELECT` on `DBA_INDEXES`
- `SELECT` on `DBA_TABLES`
- `SELECT` on `DBA_TRIGGERS`
- `SELECT` on `SYS.DBA_REGISTRY`

Wenn eine der erforderlichen Berechtigungen `V$xxx` nicht erteilt werden kann, erteilen Sie sie `V_`
`$xxx`.

Bewertungen vor der Migration

Um die unter [Bewertungen von Oracle](#) Mit Oracle as a Target aufgeführten Bewertungen vor der Migration verwenden zu können, müssen Sie dem `dms_user` Datenbankbenutzer in der Zieldatenbank die folgenden Berechtigungen hinzufügen:

```
GRANT SELECT ON V_$INSTANCE TO dms_user;
```

Konfiguration einer Oracle-Datenbank als Ziel für AWS Database Migration Service

Bevor Sie eine Oracle-Datenbank als Datenmigrationsziel verwenden können, müssen Sie ein Oracle-Benutzerkonto für angeben AWS DMS. Das Benutzerkonto muss über Lese-/Schreibberechtigungen für die Oracle-Datenbank verfügen, wie angegeben unter [Benutzerkontoberechtigungen, die bei der Verwendung von Oracle als Ziel erforderlich sind](#).

Endpunkteinstellungen bei Verwendung von Oracle als Ziel für AWS DMS

Sie können Endpunkteinstellungen zum Konfigurieren Ihrer Oracle-Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen an, wenn Sie den Zielpunkt mithilfe der AWS DMS Konsole oder mithilfe des `create-endpoint` Befehls in [AWS CLI](#), mit der `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax erstellen.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Oracle als Ziel verwenden können.

Name	Beschreibung
EscapeCharacter	<p>Legen Sie für dieses Attribut ein Escape-Zeichen fest. Mit diesem Escape-Zeichen können Sie festlegen, dass sich ein einzelnes Platzhalterzeichen in Tabellenzuordnungsaustrücken wie ein normales Zeichen verhält. Weitere Informationen finden Sie unter Platzhalter bei der Tabellenzuordnung.</p> <p>Standardwert: Null</p> <p>Gültige Werte: alle Zeichen außer Platzhalterzeichen</p> <p>Beispiel: <code>--oracle-settings '{"Escape Character": "#"}'</code></p>

Name	Beschreibung
UseDirectPathFullLoad	<p>Wenn auf gesetztY, AWS DMS verwendet Volllast einen direkten Pfad. Geben Sie diesen Wert an, um das Direktpfad-Protokoll in der Oracle Call Interface (OCI) zu aktivieren. Dieses OCI-Protokoll ermöglicht das Massensladen von Oracle-Zieltabellen während eines vollständigen Ladevorgangs.</p> <p>Standardwert: true</p> <p>Zulässige Werte: true/false</p> <p>Beispiel: <code>--oracle-settings '{"UseDirectPathFullLoad": false}'</code></p>

Name	Beschreibung
DirectPathParallelLoad	<p>Wenn <code>true</code> festgelegt ist, gibt dieses Attribut einen parallelen Ladevorgang an, wenn <code>UseDirectPathFullLoad</code> auf <code>Y</code> eingestellt ist. Dieses Attribut gilt auch nur, wenn Sie die Funktion für AWS DMS paralleles Laden verwenden. Weitere Informationen finden Sie in der Beschreibung der Operation <code>parallel-load</code> unter Regeln und Operationen für Tabellen- und Sammlungs-einstellungen.</p> <p>Eine Einschränkung bei der Angabe dieser Einstellung für paralleles Laden besteht darin, dass die Zieltabelle keine Einschränkungen oder Indizes aufweisen kann. Weitere Informationen zu dieser Einschränkung finden Sie unter Enabling Constraints After a Parallel Direct Path Load. Wenn Einschränkungen oder Indizes aktiviert sind, hat das Einstellen dieses Attributs auf <code>true</code> keine Auswirkung.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--oracle-settings '{"DirectPathParallelLoad": true}'</code></p>

Name	Beschreibung
DirectPathNoLog	<p>Wenn <code>true</code> festgelegt ist, hilft dieses Attribut, die Commit-Rate für die Oracle-Zieldatenbank zu erhöhen, indem direkt in Tabellen geschrieben und kein Trail zu Datenbankprotokollen geschrieben wird. Weitere Informationen finden Sie unter Direct-Load INSERT. Auch dieses Attribut gilt nur, wenn Sie <code>UseDirectPathFullLoad</code> auf <code>Y</code> einstellen.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>Beispiel: <code>--oracle-settings '{"DirectPathNoLog": true}'</code></p>
CharLengthSemantics	<p>Gibt an, ob die Länge einer Zeichenspalte in Bytes oder in Zeichen ist. Um anzugeben, dass die Länge der Zeichenspalte in Zeichen ist, setzen Sie dieses Attribut auf <code>CHAR</code>. Andernfalls ist die Länge der Zeichenspalte in Bytes.</p> <p>Standardwert: Nicht auf <code>CHAR</code> eingestellt</p> <p>Zulässige Werte: <code>CHAR</code></p> <p>Beispiel: <code>--oracle-settings '{"CharLengthSemantics": "CHAR"}'</code></p>

Name	Beschreibung
AlwaysReplaceEmptyString	<p>AWS DMS fügt bei der Migration zu einem Oracle-Ziel ein zusätzliches Leerzeichen hinzu, um eine leere Zeichenfolge zu replizieren. Im Allgemeinen gibt es in Oracle keine Notation für eine leere Zeichenfolge. Wenn Sie eine leere Zeichenfolge in varchar2 einfügen, laden Sie leere Zeichenfolgen als NULL. Wenn Sie die Daten in Oracle als NULL einfügen möchten, setzen Sie dieses Attribut auf FALSE.</p> <p>Standardwert: true</p> <p>Zulässige Werte: true/false</p> <p>Beispiel: <code>--oracle-settings '{"Always ReplaceEmptyString": false}'</code></p>

Zieldatentypen für Oracle

Eine Oracle-Zieldatenbank, die mit verwendet wird, AWS DMS unterstützt die meisten Oracle-Datentypen. Die folgende Tabelle zeigt die Oracle-Zieldatentypen, die bei der Verwendung unterstützt werden, AWS DMS sowie die Standardzuweisung von AWS DMS Datentypen. Weitere Informationen zum Anzeigen der Datentypen, die von der Quelle zugeordnet werden, finden Sie im Abschnitt für die Quelle, die Sie verwenden.

AWS DMS Datentyp	Oracle-Datentyp
BOOLEAN	NUMBER (1)
BYTES	RAW (Länge)
DATUM	DATETIME
TIME	TIMESTAMP (0)
DATETIME	TIMESTAMP (Skalierung)

AWS DMS Datentyp	Oracle-Datentyp
INT1	NUMBER (3)
INT2	NUMBER (5)
INT4	NUMBER (10)
INT8	NUMBER (19)
NUMERIC	NUMBER (p,s)
REAL4	FLOAT
REAL8	FLOAT
STRING	<p>Mit Datumsangabe: DATE</p> <p>Mit Zeitangabe: TIMESTAMP</p> <p>Mit Zeitstempelangabe: TIMESTAMP</p> <p>Mit Angabe von "timestamp_with_timezone": TIMESTAMP WITH TIMEZONE</p> <p>Mit Angabe von "timestamp_with_local_timezone": TIMESTAMP WITH LOCAL TIMEZONE Mit Angabe von "interval_year_to_month": INTERVAL YEAR TO MONTH</p> <p>Mit Angabe von "interval_day_to_second": INTERVAL DAY TO SECOND</p> <p>Wenn Länge > 4000: CLOB</p> <p>In allen anderen Fällen: VARCHAR2 (Länge)</p>
UINT1	NUMBER (3)
UINT2	NUMBER (5)
UINT4	NUMBER (10)

AWS DMS Datentyp	Oracle-Datentyp
UINT8	NUMBER (19)
WSTRING	Wenn Länge > 2000: NCLOB In allen anderen Fällen: NVARCHAR2 (Länge)
BLOB	BLOB Um diesen Datentyp mit verwenden zu können AWS DMS, müssen Sie die Verwendung von BLOBs für eine bestimmte Aufgabe aktivieren. BLOB-Datentypen werden nur in Tabellen unterstützt, die einen Primärschlüssel enthalten.
CLOB	CLOB Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von CLOBs für eine bestimmte Aufgabe aktivieren. Bei Change Data Capture (CDC) unterstützt werden CLOB-Datentypen nur in Tabellen unterstützt, die einen Primärschlüssel enthalten. STRING Ein Oracle VARCHAR2-Datentyp in der Quelle mit einer deklarierten Größe von mehr als 4000 Byte wird über den AWS DMS CLOB einem STRING auf dem Oracle-Ziel zugeordnet.
NCLOB	NCLOB Um diesen Datentyp mit zu verwenden AWS DMS, müssen Sie die Verwendung von NCLOBs für eine bestimmte Aufgabe aktivieren. Bei CDC werden NCLOB-Datentypen nur in Tabellen unterstützt, die einen Primärschlüssel enthalten. WSTRING Ein Oracle VARCHAR2-Datentyp in der Quelle mit einer deklarierten Größe von mehr als 4000 Byte wird über den AWS DMS NCLOB einem WSTRING auf dem Oracle-Ziel zugeordnet.

AWS DMS Datentyp	Oracle-Datentyp
XMLTYPE	<p>Der Zieldatentyp XMLTYPE ist nur bei Oracle-zu-Oracle-Replikationsaufgaben relevant.</p> <p>Wenn die Quelldatenbank Oracle ist, werden die Quelldatentypen unverändert in der Oracle-Zielfdatenbank repliziert. Beispiel: Ein XMLTYPE-Datentyp in der Quelldatenbank wird als XMLTYPE-Datentyp in der Zielfdatenbank erstellt.</p>

Verwenden einer Microsoft SQL Server-Datenbank als Ziel für AWS Database Migration Service

Sie können Daten mithilfe von AWS DMS in Microsoft SQL Server-Datenbanken migrieren. Mit einer SQL Server-Datenbank als Ziel können Sie Daten aus einer anderen SQL Server-Datenbank oder einer der anderen unterstützten Datenbanken migrieren.

Informationen zu den Versionen von SQL Server, die AWS DMS als Ziel unterstützt, finden Sie unter [Ziele für AWS DMS](#).

AWS DMS unterstützt die On-Premises- und Amazon-RDS-Editionen von Enterprise, Standard, Workgroup und Developer.

Weitere Informationen zur Arbeit mit AWS DMS und SQL Server-Zielfdatenbanken finden Sie in den folgenden Themen.

Themen

- [Einschränkungen bei der Verwendung von SQL Server als Ziel für AWS Database Migration Service](#)
- [Sicherheitsanforderungen bei der Verwendung von SQL Server als Ziel für AWS Database Migration Service](#)
- [Endpunkteinstellungen bei Verwendung von SQL Server als Ziel für AWS DMS](#)
- [Zieldatentypen für Microsoft SQL Server](#)

Einschränkungen bei der Verwendung von SQL Server als Ziel für AWS Database Migration Service

Die folgenden Einschränkungen gelten, wenn Sie eine SQL Server-Datenbank als Ziel für AWS DMS verwenden:

- Wenn Sie eine SQL Server-Zieltabelle mit einer berechneten Spalte manuell erstellen, wird die Volllastreplikation bei Verwendung des BCP-Dienstprogramms zum Kopieren großer Datenmengen nicht unterstützt. Um eine Replikation mit vollständigem Ladevorgang zu verwenden, deaktivieren Sie das BCP-Laden, indem Sie das zusätzliche Verbindungsattribut (Extra Connection Attribute, ECA) 'useBCPFullLoad=false' auf dem Endpunkt festlegen. Informationen zum Einrichten von ECAs auf Endpunkten finden Sie unter [Erstellen der Quell- und Zielendpunkte](#). Weitere Informationen zum Arbeiten mit BCP finden Sie in der [Microsoft SQL Server-Dokumentation](#).
- Bei der Replikation von Tabellen mit den räumlichen Datentypen von SQL Server (GEOMETRY und GEOGRAPHY) ersetzt AWS DMS alle räumlichen Referenzkennungen (Spatial Reference Identifier, SRID), die Sie möglicherweise mit der Standard-SRID eingefügt haben. Die Standard-SRID für GEOMETRY ist 0 und für GEOGRAPHY 4326.
- Temporäre Tabellen werden nicht unterstützt. Die Migration von temporären Tabellen ist mit einer reinen Replikationsaufgabe im transaktionalen Anwendungsmodus möglich, wenn diese Tabellen manuell auf dem Ziel erstellt werden.
- Derzeit werden boolean-Datentypen in einer PostgreSQL-Quelle als bit-Datentyp mit inkonsistenten Werten zu einem SQLServer-Ziel migriert.

Führen Sie folgende Schritte aus, um dieses Problem zu umgehen:

- Erstellen Sie die Tabelle vorab mit einem Datentyp VARCHAR(1) für die Spalte (oder lassen Sie AWS DMS die Tabelle erstellen). Dann lassen Sie die nachgeschaltete Verarbeitung ein „F“ als „False“ und ein „T“ als „True“ behandeln.
- Um die nachgeschaltete Verarbeitung nicht ändern zu müssen, fügen Sie der Aufgabe eine Transformationsregel hinzu, um die „F“-Werte in „0“ und die „T“-Werte in „1“ zu ändern, und speichern Sie diese als SQL-Server-Bit-Datentyp.
- AWS DMS unterstützt keine Änderungsverarbeitung zum Festlegen von Nullfähigkeit für Spalten (unter Verwendung der Klausel ALTER COLUMN [SET|DROP] NOT NULL mit ALTER TABLE-Anweisungen).
- Windows-Authentifizierung wird nicht unterstützt.

Sicherheitsanforderungen bei der Verwendung von SQL Server als Ziel für AWS Database Migration Service

Im folgenden Abschnitt werden die Sicherheitsanforderungen bei der Verwendung von AWS DMS mit einem Microsoft SQL Server-Ziel beschrieben:

- Das AWS DMS-Benutzerkonto muss mindestens über die `db_owner`-Benutzerrolle für die Microsoft SQL Server-Datenbank verfügen, mit der Sie eine Verbindung herstellen.
- Ein SQL Server-Systemadministrator muss diese Berechtigung allen AWS DMS-Benutzerkonten erteilen.

Endpunkteinstellungen bei Verwendung von SQL Server als Ziel für AWS DMS

Sie können Endpunkteinstellungen zum Konfigurieren Ihrer SQL-Server-Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen beim Erstellen des Zielendpunkts über die AWS DMS-Konsole oder über den Befehl `create-endpoint` in der [AWS CLI](#) mit der JSON-Syntax `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` an.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit SQL Server als Ziel verwenden können.

Name	Beschreibung
<code>ControlTablesFileGroup</code>	<p>Geben Sie eine Dateigruppe für die internen AWS DMS-Tabellen an. Wenn die Replikationsaufgabe gestartet wird, werden alle internen AWS DMS-Steuerungstabellen (<code>awsdms_apply_exception</code>, <code>awsdms_apply</code>, <code>awsdms_changes</code>) in der angegebenen Dateigruppe erstellt.</p> <p>Standardwert: –</p> <p>Zulässige Werte: String</p> <p>Beispiel: <code>--microsoft-sql-server-settings '{"ControlTablesFileGroup": "filegroup1"}'</code></p>

Name	Beschreibung
	<p>Es folgt ein Beispiel für einen Befehl zum Erstellen einer Dateigruppe.</p> <pre data-bbox="688 327 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>Verwenden Sie dieses zusätzliche Verbindungsattribut (Extra Connection Attribute, ECA), um das Timeout für die Client-Anweisung für die SQL-Server-Instanz in Sekunden festzulegen. Der Standardwert liegt bei 60 Sekunden.</p> <p>Beispiel: <code>'{"ExecuteTimeout": 100}'</code></p>
UseBCPFullLoad	<p>Verwenden Sie dieses Attribut zum Übertragen von Daten für vollständige Ladevorgänge mithilfe von BCP. Wenn die Zieltabelle eine Identitätsspalte enthält, die in der Quelltable nicht vorhanden ist, müssen Sie die Option BCP zum Laden der Tabelle verwenden deaktivieren.</p> <p>Standardwert: aktiviert</p> <p>Zulässige Werte: true/false</p> <p>Beispiel: <code>--microsoft-sql-server-settings {"UseBCPFullLoad": false}'</code></p>

Zieldatentypen für Microsoft SQL Server

In der folgenden Tabelle sind die Microsoft SQL Server-Zieldatentypen aufgeführt, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung von AWS DMS-Datentypen. Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

AWS DMS-Datentyp	SQL Server-Datentyp
BOOLEAN	TINYINT
BYTES	VARBINARY (Länge)
DATUM	Verwenden Sie für SQL Server 2008 und höher DATE. Verwenden Sie für frühere Versionen, wenn die Skalierung 3 oder weniger ist, DATETIME. Verwenden Sie in allen anderen Fällen VARCHAR (37).
TIME	Verwenden Sie für SQL Server 2008 und höher DATETIME2 (%d). Verwenden Sie für frühere Versionen, wenn die Skalierung 3 oder weniger ist, DATETIME. Verwenden Sie in allen anderen Fällen VARCHAR (37).
DATETIME	Verwenden Sie für SQL Server 2008 und höher DATETIME2 (Skalierung). Verwenden Sie für frühere Versionen, wenn die Skalierung 3 oder weniger ist, DATETIME. Verwenden Sie in allen anderen Fällen VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT

AWS DMS-Datentyp	SQL Server-Datentyp
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	FLOAT
STRING	<p>Wenn die Spalte eine Datums- oder Zeitspalte ist, gehen Sie wie folgt vor:</p> <ul style="list-style-type: none"> • Verwenden Sie für SQL Server 2008 und höher DATETIME2. • Verwenden Sie für frühere Versionen, wenn die Skalierung 3 oder weniger ist, DATETIME. Verwenden Sie in allen anderen Fällen VARCHAR (37). <p>Wenn die Spalte keine Datums- oder Zeitspalte ist, verwenden Sie VARCHAR (Länge).</p>
UINT1	TINYINT
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR (Länge)
BLOB	<p>VARBINARY(max)</p> <p>IMAGE</p> <p>Um diesen Datentyp mit AWS DMS zu verwenden, müssen Sie die Verwendung von BLOBs für eine spezifische Aufgabe aktivieren. AWS DMS unterstützt BLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>

AWS DMS-Datentyp	SQL Server-Datentyp
CLOB	VARCHAR(max) Wenn Sie diesen Datentyp mit AWS DMS verwenden möchten, müssen Sie die Verwendung von CLOBs für eine spezifische Aufgabe aktivieren. Bei Change Data Capture (CDC) unterstützt AWS DMS CLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.
NCLOB	NVARCHAR(max) Wenn Sie diesen Datentyp mit AWS DMS verwenden möchten, müssen Sie die Verwendung von NCLOBs für eine spezifische Aufgabe aktivieren. Bei CDC unterstützt AWS DMS NCLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.

Verwenden einer PostgreSQL-Datenbank als Ziel für AWS Database Migration Service

Sie können Daten entweder aus einer anderen PostgreSQL-Datenbank oder aus einer der anderen unterstützten Datenbanken in PostgreSQL-Datenbanken migrieren. AWS DMS

Hinweise zu Versionen von PostgreSQL, die als Ziel AWS DMS unterstützen, finden Sie unter [Ziele für AWS DMS](#)

AWS DMS zu verwendende Version

Verwenden Sie eine beliebige verfügbare AWS DMS Version.

Verwenden Sie AWS DMS Version 3.4.3 und höher.

AWS DMS zu verwendende Version

Verwenden Sie AWS DMS Version 3.4.7 und höher.

Verwenden Sie AWS DMS Version 3.5.1 und höher.

Note

- Amazon Aurora Serverless ist als ZIEL für Amazon Aurora mit PostgreSQL-Kompatibilität verfügbar. Weitere Informationen zu Amazon Aurora Serverless finden Sie unter [Using Amazon Aurora Serverless v2](#) im Amazon Aurora Aurora-Benutzerhandbuch.
- Aurora-Serverless-DB-Cluster sind nur von einer Amazon VPC aus zugänglich und können keine [öffentliche IP-Adresse](#) verwenden. Wenn Sie also eine Replikations-Instance in einer anderen Region als Aurora PostgreSQL Serverless haben möchten, müssen Sie [VPC-Peering](#) konfigurieren. Prüfen Sie andernfalls die Verfügbarkeit von Aurora-PostgreSQL-Serverless-[Regionen](#) und entscheiden Sie sich, eine dieser Regionen sowohl für Aurora PostgreSQL Serverless als auch für Ihre Replikations-Instance zu verwenden.
- Die Babelfish-Funktion ist in Amazon Aurora integriert. Hierfür entstehen keine zusätzlichen Kosten. Weitere Informationen finden Sie unter [Verwendung von Babelfish für Aurora PostgreSQL als Ziel für AWS Database Migration Service](#).

AWS DMS verfolgt einen table-by-table Ansatz bei der Migration von Daten von der Quelle zum Ziel in der Vollastphase. Die Tabellenreihenfolge kann während der Vollastphase nicht garantiert werden. Die Tabellen sind während der Vollastphase und während des Anwendens der zwischengespeicherten Transaktionen für einzelne Tabellen nicht synchron. Dies hat zur Folge, dass aktive referentielle Integritätseinschränkungen während der Vollastphase zu Aufgabenfehlern führen können.

In PostgreSQL werden Fremdschlüssel (referentielle Integritätseinschränkungen) mithilfe von Auslösern implementiert. AWS DMS lädt während der Vollastphase jede Tabelle einzeln. Es wird ausdrücklich empfohlen, dass Sie Fremdschlüsseleinschränkungen während eines vollständigen Ladevorgangs mit einer der folgenden Methoden deaktivieren:

- Deaktivieren Sie vorübergehend alle Auslöser aus der Instance und beenden Sie den vollständigen Ladevorgang.
- Verwenden Sie den `session_replication_role`-Parameter in PostgreSQL.

Ein Auslöser kann sich zu jedem beliebigen Zeitpunkt in einem der folgenden Zustände befinden: `origin`, `replica`, `always` und `disabled`. Wenn der `session_replication_role`-Parameter auf `replica` gesetzt ist, sind nur Auslöser aktiv, die sich im Status `replica` befinden, und sie werden ausgelöst, wenn sie aufgerufen werden. Andernfalls bleiben die Auslöser inaktiv.

PostgreSQL verfügt über einen ausfallsicheren Mechanismus, um zu verhindern, dass eine Tabelle gekürzt wird, auch dann, wenn `session_replication_role` festgelegt ist. Sie können diesen als Alternative zur Deaktivierung von Auslösern verwenden, um den vollständigen Ladevorgang abzuschließen. Legen Sie hierfür den Vorbereitungsmodus der Zieltabelle auf `DO_NOTHING` fest. Andernfalls schlagen `DROP`- und `TRUNCATE`-Operationen fehl, wenn Fremdschlüsseleinschränkungen bestehen.

In Amazon RDS können Sie diesen Parameter mithilfe einer Parametergruppe steuern/festlegen. Für eine PostgreSQL-Instance, die in Amazon EC2 ausgeführt wird, können Sie den Parameter direkt festlegen.

Weitere Informationen zur Arbeit mit einer PostgreSQL-Datenbank als Ziel für AWS DMS finden Sie in den folgenden Abschnitten:

Themen

- [Einschränkungen bei der Verwendung von PostgreSQL als Ziel für AWS Database Migration Service](#)
- [Sicherheitsanforderungen bei der Verwendung einer PostgreSQL-Datenbank als Ziel für AWS Database Migration Service](#)
- [Endpunkteinstellungen und zusätzliche Verbindungsattribute \(ECAs\) bei Verwendung von PostgreSQL als Ziel für AWS DMS](#)
- [Zieldatentypen für PostgreSQL](#)
- [Verwenden von Babelfish für Aurora PostgreSQL als Ziel für AWS Database Migration Service](#)

Einschränkungen bei der Verwendung von PostgreSQL als Ziel für AWS Database Migration Service

Bei der Verwendung einer PostgreSQL-Datenbank als Ziel für AWS DMS gelten die folgenden Einschränkungen:

- Bei heterogenen Migrationen wird der JSON-Datentyp intern in den nativen CLOB-Datentyp konvertiert.
- Wenn bei einer Migration von Oracle nach PostgreSQL eine Spalte in Oracle ein NULL-Zeichen (Hexadezimalwert U+0000) enthält, wird das NULL-Zeichen in ein Leerzeichen AWS DMS umgewandelt (Hexadezimalwert U+0020). Der Grund hierfür ist eine PostgreSQL-Einschränkung.
- AWS DMS unterstützt keine Replikation in eine Tabelle mit einem eindeutigen Index, der mit der Coalesce-Funktion erstellt wurde.
- Wenn Ihre Tabellen Sequenzen verwenden, aktualisieren Sie den Wert von NEXTVAL für jede Sequenz in der Zieldatenbank, nachdem Sie die Replikation aus der Quelldatenbank beendet haben. AWS DMS kopiert Daten aus Ihrer Quelldatenbank, migriert aber während der laufenden Replikation keine Sequenzen zur Zieldatenbank.

Sicherheitsanforderungen bei der Verwendung einer PostgreSQL-Datenbank als Ziel für AWS Database Migration Service

Aus Sicherheitsgründen muss das für die Datenmigration verwendete Benutzerkonto ein registrierter Benutzer in einer PostgreSQL-Zieldatenbank sein.

Ihr PostgreSQL-Zielendpunkt erfordert Mindestbenutzerberechtigungen, um eine AWS DMS Migration durchzuführen. Weitere Informationen finden Sie in den folgenden Beispielen.

```
CREATE USER newuser WITH PASSWORD 'your-password';  
ALTER SCHEMA schema_name OWNER TO newuser;
```

Oder

```
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;
```

Endpunkteinstellungen und zusätzliche Verbindungsattribute (ECAs) bei Verwendung von PostgreSQL als Ziel für AWS DMS

Sie können Endpunkteinstellungen und Extra Connection Attributes (ECAs) verwenden, um Ihre PostgreSQL-Zieldatenbank zu konfigurieren.


Sie geben die Einstellungen an, wenn Sie den Zielendpunkt mithilfe der AWS DMS Konsole erstellen, oder indem Sie den `create-endpoint` Befehl in [AWS CLI](#), mit der `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax verwenden.

Sie geben ECAs mithilfe des `ExtraConnectionAttributes` Parameters für Ihren Endpunkt an.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit PostgreSQL als Ziel verwenden können.

Name	Beschreibung
<code>MaxFileSize</code>	<p>Gibt die maximale Größe (in KB) der CSV-Datei zum Übertragen von Daten nach PostgreSQL an.</p> <p>Standardwert: 32768 KB (32 MB)</p> <p>Gültige Werte: 1 bis 1 048 576 KB (bis zu 1,1 GB)</p> <p>Beispiel: <code>--postgre-sql-settings '{"MaxFileSize": 512}'</code></p>
<code>ExecuteTimeout</code>	<p>Stellt die Zeitbeschränkung der Client-Anweisung für die PostgreSQL-Instance in Sekunden ein. Der Standardwert liegt bei 60 Sekunden.</p> <p>Beispiel: <code>--postgre-sql-settings '{"ExecuteTimeout": 100}'</code></p>

Name	Beschreibung
AfterConnectScript= SET session_replication_role = replica	Dieses Attribut verfügt über AWS DMS Bypass-Fremdschlüssel und Benutzer-Trigger, um die Zeit zu reduzieren, die für das Laden von Daten in großen Mengen benötigt wird.

Name	Beschreibung
MapUnboundedNumericAsString	<p data-bbox="686 226 1495 548">Dieser Parameter behandelt Spalten mit unbegrenzten NUMERIC-Datentypen als STRING, um eine erfolgreiche Migration ohne Verlust der Präzision des numerischen Werts zu ermöglichen. Verwenden Sie diesen Parameter nur für die Replikation von einer PostgreSQL-Quelle zu einem PostgreSQL-Ziel oder für Datenbanken mit PostgreSQL-Kompatibilität.</p> <p data-bbox="686 594 963 625">Standardwert: false</p> <p data-bbox="686 672 1065 703">Zulässige Werte: false/true</p> <p data-bbox="686 749 1430 835">Beispiel: <code>--postgresql-settings '{"MapUnboundedNumericAsString": "true"}</code></p> <p data-bbox="686 882 1495 1150">Die Verwendung dieses Parameters kann zu einer gewissen Beeinträchtigung der Replikationsleistung führen, da eine Transformation von numerischen Werten in Zeichenfolgen und zurück in numerische Werte erfolgt. Dieser Parameter wird für DMS-Version 3.4.4 und höher unterstützt</p> <div data-bbox="686 1192 1507 1757" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p data-bbox="716 1234 834 1266"> Note</p><p data-bbox="764 1291 1425 1419">Verwenden Sie <code>MapUnboundedNumericAsString</code> nur in PostgreSQL-Quell- und -Zielendpunkten gemeinsam.</p><p data-bbox="764 1434 1446 1757">Bei Verwendung von <code>MapUnboundedNumericAsString</code> für PostgreSQL-Quellendpunkte wird die Präzision während CDC auf 28 beschränkt. Bei Verwendung von <code>MapUnboundedNumericAsString</code> für Zielendpunkte werden Daten mit einer Präzision von 28 und einer Skalierung von 6 migriert.</p></div>

Name	Beschreibung
	<p>Verwenden Sie <code>MapUnboundedNumericAsString</code> nicht mit Nicht-PostgreSQL-Zielen.</p>
<p><code>loadUsingCSV</code></p>	<p>Verwenden Sie dieses Extra Connection Attribute (ECA), um Daten für Vollladevorgänge mit dem Befehl <code>COPY</code> zu übertragen.</p> <p>Standardwert: <code>true</code></p> <p>Zulässige Werte: <code>true/false</code></p> <p>ECA-Beispiel: <code>loadUsingCSV=true;</code></p> <p>Hinweis: Wenn Sie diesen ECA auf „False“ setzen, kann dies zu einer gewissen Beeinträchtigung der Replikationsleistung führen, da INSERTs direkt ausgeführt werden.</p>
<p><code>DatabaseMode</code></p>	<p>Verwenden Sie dieses Attribut, um das Standardverhalten der Replikationsverarbeitung von Postgresql-kompatiblen Endpunkten zu ändern, die eine zusätzliche Konfiguration erfordern, wie z. B. Babelfish-Endpunkte.</p> <p>Standardwert: <code>DEFAULT</code></p> <p>Zulässige Werte: <code>DEFAULT, BABELFISH</code></p> <p>Beispiel: <code>DatabaseMode=default;</code></p>
<p><code>BabelfishDatabaseName</code></p>	<p>Verwenden Sie dieses Attribut, um den Namen der Babelfish-T-SQL-Zieldatenbank anzugeben, zu der migriert wird. Dieser ist erforderlich, wenn <code>DatabaseMode</code> auf <code>Babelfish</code> festgelegt ist. Es handelt sich hier nicht um die reservierte <code>babelfish_db</code> -Datenbank.</p> <p>Beispiel: <code>BabelfishDatabaseName=TargetDb;</code></p>

Zieldatentypen für PostgreSQL

Der PostgreSQL-Datenbankendpunkt für AWS DMS unterstützt die meisten PostgreSQL-Datenbankdatentypen. Die folgende Tabelle zeigt die Zieldatentypen der PostgreSQL-Datenbank, die bei der Verwendung unterstützt werden, AWS DMS und die Standardzuordnung von AWS DMS Datentypen.

Weitere Hinweise zu AWS DMS Datentypen finden Sie unter. [Datentypen für den AWS Database Migration Service](#)

AWS DMS Datentyp	PostgreSQL-Datentyp
BOOLEAN	BOOLEAN
BLOB	BYTEA
BYTES	BYTEA
DATUM	DATUM
TIME	TIME
DATETIME	Wenn die Skalierung 0 bis 6 ist, verwenden Sie TIMESTAMP. Wenn die Skalierung 7 bis 9 ist, verwenden Sie VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (P,S)
REAL4	FLOAT4
REAL8	FLOAT8
STRING	Ist die Länge 1 bis 21.845, verwenden Sie VARCHAR (Länge in Byte).

AWS DMS Datentyp	PostgreSQL-Datentyp
	Ist die Länge 21.846 bis 2.147.483.647, verwenden Sie VARCHAR (65535).
UINT1	SMALLINT
UINT2	INTEGER
UINT4	BIGINT
UINT8	BIGINT
WSTRING	Ist die Länge 1 bis 21.845, verwenden Sie VARCHAR (Länge in Byte). Ist die Länge 21.846 bis 2.147.483.647, verwenden Sie VARCHAR (65535).
NCLOB	TEXT
CLOB	TEXT

Note

AWS DMS Erstellt bei der Replikation aus einer PostgreSQL-Quelle die Zieltabelle mit den gleichen Datentypen für alle Spalten, mit Ausnahme von Spalten mit benutzerdefinierten Datentypen. In solchen Fällen wird der Datentyp als "character varying" im Ziel erstellt.

Verwenden von Babelfish für Aurora PostgreSQL als Ziel für AWS Database Migration Service

Sie können SQL-Server-Quellentabellen unter Verwendung von AWS Database Migration Service zu einem Babelfish-für-Amazon-Aurora-PostgreSQL-Ziel migrieren. Mithilfe von Babelfish versteht Aurora PostgreSQL T-SQL, den proprietären SQL-Dialekt von Microsoft SQL Server, und unterstützt dasselbe Kommunikationsprotokoll. Somit können für SQL Server geschriebene Anwendungen jetzt mit weniger Codeänderungen mit Aurora verwendet werden. Die Babelfish-Funktion ist in

Amazon Aurora integriert. Hierfür entstehen keine zusätzlichen Kosten. Sie können Babelfish über die Amazon-RDS-Konsole in Ihrem Amazon-Aurora-Cluster aktivieren.

Wenn Sie Ihren AWS DMS Zielendpunkt mithilfe der AWS DMS Konsolen-, API- oder CLI-Befehle erstellen, geben Sie als Ziel-Engine Amazon Aurora PostgreSQL an und nennen Sie die Datenbank `babelfish_db`. Fügen Sie im Abschnitt Endpunkteinstellungen Einstellungen hinzu, um für `DatabaseMode` `Babelfish` und für `BabelfishDatabaseName` den Namen der Babelfish-T-SQL-Zieldatenbank festzulegen.

Hinzufügen von Transformationsregeln zu Ihrer Migrationsaufgabe

Wenn Sie eine Migrationsaufgabe für ein Babelfish-Ziel definieren, müssen Sie Transformationsregeln einbeziehen, die sicherstellen, dass DMS die vorab erstellten T-SQL-Babelfish-Tabellen in der Zieldatenbank verwendet.

Fügen Sie Ihrer Migrationsaufgabe zunächst eine Transformationsregel hinzu, die alle Tabellennamen in Kleinbuchstaben umschreibt. Babelfish speichert die Namen von Tabellen, die Sie mit T-SQL erstellen, in Kleinbuchstaben im PostgreSQL-Katalog `pg_class`. Wenn Sie jedoch über SQL-Server-Tabellen mit Namen in Groß- und Kleinschreibung verfügen, erstellt DMS die Tabellen unter Verwendung von systemeigenen PostgreSQL-Datentypen anstelle der T-SQL-kompatiblen Datentypen. Aus diesem Grund sollten Sie unbedingt eine Transformationsregel hinzufügen, die alle Tabellennamen in Kleinbuchstaben umschreibt. Beachten Sie, dass Spaltennamen nicht in Kleinbuchstaben umgewandelt werden sollten.

Wenn Sie bei der Definition Ihres Clusters den Migrationsmodus für mehrere Datenbanken verwendet haben, fügen Sie als Nächstes eine Transformationsregel hinzu, die das ursprüngliche SQL-Server-Schema umbenennt. Achten Sie darauf, das SQL-Server-Schema so umzubenennen, dass der Name den Namen der T-SQL-Datenbank beinhaltet. Wenn der ursprüngliche SQL-Server-Schemaname beispielsweise `dbo` und Ihr T-SQL-Datenbankname `mydb` lautet, benennen Sie das Schema unter Verwendung einer Transformationsregel in `mydb_dbo` um.

Wenn Sie den Einzeldatenbankmodus verwenden, benötigen Sie keine Transformationsregel, um Schemanamen umzubenennen. Schemanamen haben eine one-to-one Zuordnung zur T-SQL-Zieldatenbank in Babelfish.

Die folgende Beispieltransformationsregel schreibt alle Tabellennamen in Kleinbuchstaben um und benennt den ursprünglichen SQL-Server-Schemanamen von `dbo` in `mydb_dbo` um.

```
{
  "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": []
}
]
```


Einschränkungen bei der Verwendung eines PostgreSQL-Zielendpunkts mit Babelfish-Tabellen

Bei der Verwendung eines PostgreSQL-Zielendpunkts mit Babelfish-Tabellen gelten folgende Einschränkungen:

- Verwenden Sie für den Zieltabellen-Vorbereitungsmodus nur die Modi Nichts unternehmen oder Kürzen. Verwenden Sie nicht den Modus Tabellen am Ziel ablegen. In diesem Modus erstellt DMS die Tabellen als PostgreSQL-Tabellen, die T-SQL möglicherweise nicht erkennt.
- AWS DMS unterstützt den Datentyp `sql_variant` nicht.
- Babelfish unterstützt die Datentypen `HEIRARCHYID`, `GEOMETRY` und `GEOGRAPHY` nicht. Für die Migration dieser Datentypen können Sie Transformationsregeln hinzufügen, um den Datentyp in `wstring(250)` zu konvertieren.
- Babelfish unterstützt nur die Migration der Datentypen `BINARY`, `VARBINARY` und `IMAGE` unter Verwendung des Datentyps `BYTEA`. Für frühere Versionen von Aurora PostgreSQL können Sie DMS verwenden, um diese Tabellen zu einem [Babelfish-Zielendpunkt](#) zu migrieren. Sie müssen keine Länge für den Datentyp `BYTEA` angeben, wie im folgenden Beispiel zu sehen.

```
[Picture] [VARBINARY](max) NULL
```

Ändern Sie den vorherigen T-SQL-Datentyp in den von T-SQL unterstützten Datentyp `BYTEA`.

```
[Picture] BYTEA NULL
```

- Wenn Sie in früheren Versionen von Aurora PostgreSQL Babelfish eine Migrationsaufgabe für die laufende Replikation von SQL Server zu Babelfish unter Verwendung des PostgreSQL-Zielendpunkts erstellen, müssen Sie den Datentyp `SERIAL` allen Tabellen zuweisen, die Spalten `IDENTITY` verwenden. Ab Aurora PostgreSQL (Version 15.3/14.8 und höher) und Babelfish (Version 3.2.0 und höher) wird die Identitätsspalte unterstützt und es ist nicht mehr erforderlich, den Datentyp `SERIAL` zuzuweisen. Weitere Informationen finden Sie unter [Verwendung von SERIAL](#) im Abschnitt zu Sequenzen und Identität des Playbooks zur Migration von SQL Server zu Aurora PostgreSQL. Wenn Sie dann die Tabelle in Babelfish erstellen, ändern Sie die Spaltendefinition der folgenden Angabe.

```
[IDCo1] [INT] IDENTITY(1,1) NOT NULL PRIMARY KEY
```

Ändern Sie die vorherige Angabe in Folgendes.

[IDCo1] SERIAL PRIMARY KEY

Das Babelfish-kompatible Aurora PostgreSQL erstellt eine Sequenz mit der Standardkonfiguration und fügt der Spalte eine Einschränkung NOT NULL hinzu. Die neu erstellte Sequenz verhält sich wie eine reguläre Sequenz (um 1 erhöht) und enthält keine zusammengesetzte Option SERIAL.

- Nachdem Sie Daten mit Tabellen migriert haben, die Spalten IDENTITY oder den Datentyp SERIAL verwenden, setzen Sie das PostgreSQL-basierte Sequenzobjekt auf der Grundlage des Maximalwerts für die Spalte zurück. Wenn die Tabellen vollständig geladen sind, verwenden Sie die folgende T-SQL-Abfrage, um Anweisungen zu generieren, um das zugehörige Sequenzobjekt zu setzen.

```
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(%)';
```

Die Abfrage generiert eine Reihe von SELECT-Anweisungen, die Sie ausführen, um die maximalen IDENTITY- und SERIAL-Werte zu aktualisieren.

- Bei Babelfish-Versionen vor 3.2 kann Vollständiger LOB-Modus zu einem Tabellenfehler führen. Erstellen Sie in diesem Fall eine separate Aufgabe für die Tabellen, die nicht geladen werden konnten. Verwenden Sie dann Limitierter LOB-Modus, um den entsprechenden Wert für Maximale

LOB-Größe (KB) anzugeben. Eine weitere Option besteht darin, für das SQL-Server-Endpoint-Verbindungsattribut die Einstellung `ForceFullLob=True` festzulegen.

- Bei Babelfish-Versionen vor 3.2 wird bei der Datenvalidierung mit Babelfish-Tabellen, die keine auf Ganzzahlen basierenden Primärschlüssel verwenden, eine Meldung generiert, dass kein geeigneter eindeutiger Schlüssel gefunden werden kann. Ab Aurora PostgreSQL (Version 15.3/14.8 und höher) und Babelfish (Version 3.2.0 und höher) wird die Datenvalidierung für nicht ganzzahlige Primärschlüssel unterstützt.
- Aufgrund von Genauigkeitsunterschieden bei der Anzahl der Dezimalstellen für Sekunden meldet DMS Datenvalidierungsfehler für Babelfish-Tabellen, die Datentypen DATETIME verwenden. Um diese Fehler zu unterdrücken, können Sie den folgenden Validierungsregeltyp für Datentypen DATETIME hinzufügen.

```
{
  "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"
}
```

Verwenden einer MySQL-kompatiblen Datenbank als Ziel für AWS Database Migration Service

Sie können Daten mit AWS DMS jeder der unterstützten Quell-Daten-Engines in jede MySQL-kompatible Datenbank migrieren. AWS DMS Wenn Sie zu einer lokalen MySQL-kompatiblen Datenbank migrieren, AWS DMS muss sich Ihre Quell-Engine innerhalb des Ökosystems befinden. AWS Die Engine kann sich in einem AWS verwalteten Service wie Amazon RDS, Amazon Aurora oder Amazon S3 befinden. Alternativ kann sich die Engine in einer selbstverwalteten Datenbank auf Amazon EC2 befinden.

Sie können mit SSL Verbindungen zwischen Ihrem MySQL-kompatiblen Endpunkt und der Replikations-Instance verschlüsseln. Weitere Informationen zur Verwendung von SSL mit einem MySQL-kompatiblen Endpunkt finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).

Hinweise zu Versionen von MySQL, die als Ziel AWS DMS unterstützen, finden Sie unter [Ziele für AWS DMS](#).

Sie können die folgenden MySQL-kompatiblen Datenbanken als Ziele verwenden für: 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

Unabhängig von der Quell-Speicher-Engine (MyISAM, MEMORY usw.) erstellt AWS DMS eine MySQL-kompatible Zieltabelle standardmäßig als InnoDB-Tabelle.

Wenn Sie eine Tabelle benötigen, die eine andere Speicher-Engine als InnoDB verwendet, können Sie die Tabelle in der MySQL-kompatiblen Zieldatenbank manuell erstellen und die Tabelle mithilfe des Modus Nichts unternehmen migrieren. Weitere Informationen finden Sie unter [Aufgabeneinstellungen für vollständiges Laden](#).

Weitere Details zum Arbeiten mit einer MySQL-kompatiblen Datenbank als Ziel für AWS DMS finden Sie in den folgenden Abschnitten.

Themen

- [Verwendung einer beliebigen MySQL-kompatiblen Datenbank als Ziel für AWS Database Migration Service](#)

- [Einschränkungen bei der Verwendung einer MySQL-kompatiblen Datenbank als Ziel für AWS Database Migration Service](#)
- [Endpunkteinstellungen bei Verwendung einer MySQL-kompatiblen Datenbank als Ziel für AWS DMS](#)
- [Ziel datentypen für MySQL](#)

Verwendung einer beliebigen MySQL-kompatiblen Datenbank als Ziel für AWS Database Migration Service

Bevor Sie mit einer MySQL-kompatiblen Datenbank als Ziel für AWS DMS arbeiten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllt haben:

- Geben Sie ein Benutzerkonto an AWS DMS, das über Lese-/Schreibberechtigungen für die MySQL-kompatible Datenbank verfügt. Führen Sie die folgenden Befehle aus, um die erforderlichen Berechtigungen zu erstellen.

```
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>'@'%';
```

- Während der Volllast-Migrationsphase müssen Sie Fremdschlüssel in Ihren Zieltabellen deaktivieren. Um Fremdschlüsselprüfungen in einer MySQL-kompatiblen Datenbank während eines vollen Ladevorgangs zu deaktivieren, können Sie den folgenden Befehl zum Abschnitt Zusätzliche Verbindungsattribute der AWS DMS Konsole für Ihren Zielendpunkt hinzufügen.

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

- Legen Sie den Datenbankparameter `local_infile = 1` fest, um AWS DMS das Laden von Daten in die Zieldatenbank zu ermöglichen.

Einschränkungen bei der Verwendung einer MySQL-kompatiblen Datenbank als Ziel für AWS Database Migration Service

Wenn Sie eine MySQL-Datenbank als Ziel verwenden, unterstützt Folgendes AWS DMS nicht:

- Die DDL-Anweisungen (Data Definition Language) TRUNCATE PARTITION, DROP TABLE und RENAME TABLE.
- Verwenden einer ALTER TABLE *table_name* ADD COLUMN *column_name*-Anweisung zum Hinzufügen von Spalten an den Anfang oder in die Mitte einer Tabelle
- Beim Laden von Daten in ein MySQL-kompatibles Ziel in einer Vollladeaufgabe werden AWS DMS keine Fehler gemeldet, die durch Einschränkungen in den Task-Logs verursacht werden, was zu doppelten Schlüsselfehlern oder Nichtübereinstimmungen mit der Anzahl der Datensätze führen kann. Dies ist auf die Art und Weise zurückzuführen, wie MySQL lokale Daten mit dem Befehl LOAD DATA behandelt. Während der Volllastphase müssen Sie folgende Schritte ausführen:
 - Einschränkungen deaktivieren
 - Stellen Sie mithilfe der AWS DMS Validierung sicher, dass die Daten konsistent sind.
- Wenn Sie den Wert einer Spalte auf den vorhandenen Wert aktualisieren, geben MySQL-kompatible Datenbanken die Warnung `0 rows affected` zurück. Obwohl dieses Verhalten technisch gesehen kein Fehler ist, unterscheidet es sich von der Art und Weise, wie andere Datenbank-Engines mit solchen Fällen umgehen. Oracle führt beispielsweise eine Aktualisierung einer Zeile durch. AWS DMS Generiert für MySQL-kompatible Datenbanken einen Eintrag in der Steuertabelle `awsdms_apply_exceptions` und protokolliert die folgende Warnung.

```
Some changes from the source database had no impact when applied to the target database. See awsdms_apply_exceptions table for details.
```

- Aurora Serverless ist als Ziel für Amazon Aurora Version 2, kompatibel mit MySQL Version 5.7, verfügbar. (Wählen Sie Aurora-MySQL-Version 2.07.1 aus, um Aurora Serverless mit MySQL-5.7-Kompatibilität verwenden zu können.) Weitere Informationen zu Aurora Serverless finden Sie unter [Using Aurora Serverless v2](#) im Amazon Aurora Benutzerhandbuch.
- AWS DMS unterstützt die Verwendung eines Reader-Endpunkts für Aurora oder Amazon RDS nicht, es sei denn, die Instances befinden sich im Schreibmodus, d. h. die `innodb_read_only` Parameter `read_only` und sind auf `0` oder `OFF` gesetzt. Weitere Informationen zur Verwendung von Amazon RDS und Aurora als Ziele finden Sie in folgenden Themen:
 - [Feststellen, mit welcher DB-Instance Sie verbunden sind](#)
 - [Aktualisieren von Lesereplikaten mit MySQL](#)

Endpunkteinstellungen bei Verwendung einer MySQL-kompatiblen Datenbank als Ziel für AWS DMS

Sie können Endpunkteinstellungen zum Konfigurieren Ihrer MySQL-kompatiblen Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen an, wenn Sie den Zielendpunkt mithilfe der AWS DMS Konsole oder mithilfe des `create-endpoint` Befehls in [AWS CLI](#), mit der `--my-sql-settings '{"EndpointSetting": "value", ...}'` JSON-Syntax erstellen.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit MySQL als Ziel verwenden können.

Name	Beschreibung
TargetDbType	<p>Gibt an, wo die Quelltabellen in der Zieldatenbank migriert werden, entweder in einer einzelnen Datenbank oder mehreren Datenbanken. Wenn Sie <code>SPECIFIC_DATABASE</code> angeben, müssen Sie den Datenbanknamen angeben, entweder wenn Sie den AWS CLI oder den verwenden AWS Management Console.</p> <p>Standardwert: <code>MULTIPLE_DATABASES</code></p> <p>Zulässige Werte: <code>{SPECIFIC_DATABASE , MULTIPLE_DATABASES }</code></p> <p>Beispiel: <code>--my-sql-settings '{"TargetDbType": "MULTIPLE_DATABASES"}</code></p>
ParallelLoadThreads	<p>Verbessert die Leistung beim Laden von Daten in die MySQL-kompatible Zieldatenbank. Gibt an, wie viele Threads zum Laden der Daten in die MySQL-kompatible Zieldatenbank verwendet werden sollen. Das Festlegen einer großen Anzahl von Threads kann die Datenbankleistung beeinträchtigen, da für jeden Thread eine separate Verbindung erforderlich ist.</p> <p>Standardwert: 1</p> <p>Zulässige Werte: 1 bis 5</p>

Name	Beschreibung
AfterConnectScript	<p data-bbox="686 212 1333 296">Beispiel: <code>--my-sql-settings '{"ParallelLoadThreads": 1}'</code></p> <p data-bbox="686 342 1507 468">Gibt ein Skript an, das sofort ausgeführt wird, nachdem AWS DMS eine Verbindung mit dem Endpunkt hergestellt hat.</p> <p data-bbox="686 514 1502 787">Sie können beispielsweise angeben, dass das MySQL-kompatible Ziel empfangene Anweisungen in den Latin1-Zeichensatz übersetzen soll. Dies ist der Zeichensatz, in den die Datenbank standardmäßig kompiliert wird. Dieser Parameter verbessert in der Regel die Leistung beim Umwandeln von UTF8-Clients.</p> <p data-bbox="686 833 1414 959">Beispiel: <code>--my-sql-settings '{"AfterConnectScript": "SET character_set_connection='latin1''}"</code></p>
MaxFileSize	<p data-bbox="686 1010 1495 1136">Gibt die maximale Größe (in KB) einer CSV-Datei an, die zum Übertragen von Daten in eine MySQL-kompatible Datenbank verwendet wird.</p> <p data-bbox="686 1182 1157 1220">Standardwert: 32768 KB (32 MB)</p> <p data-bbox="686 1266 1118 1304">Gültige Werte: 1 bis 1 048 576</p> <p data-bbox="686 1350 1450 1388"><code>--my-sql-settings '{"MaxFileSize": 512}'</code></p>

Name	Beschreibung
CleanSrcMetadataOnMismatch	<p>Bereinigt und erstellt Tabellenmetadaten-Informationen auf der Replikations-Instance, wenn eine fehlende Übereinstimmung auftritt. Zum Beispiel in einer Situation , in der das Ausführen einer alter DDL auf der Tabelle zu unterschiedlichen Informationen über die in der Replikations-Instance zwischengespeicherte Tabelle führen könnte. Boolesch.</p> <p>Standardwert: false</p> <p>Beispiel: <code>--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'</code></p>

Sie können auch zusätzliche Verbindungsattribute verwenden, um Ihre MySQL-kompatible Zieldatenbank zu konfigurieren.

Die folgende Tabelle zeigt die zusätzlichen Verbindungsattribute, die Sie verwenden können, wenn MySQL das Ziel ist.

Name	Beschreibung
Initstmt=SET FOREIGN_KEY_CHECKS=0;	<p>Deaktiviert die Fremdschlüsselprüfungen.</p> <p>Beispiel: <code>--extra-connection-attributes "Initstmt=SET FOREIGN_KEY_CHECKS=0;"</code></p>
Initstmt=SET time_zone	<p>Gibt die Zeitzone für die MySQL-kompatible Zieldatenbank an.</p> <p>Standardwert: UTC</p> <p>Gültige Werte: Die in der MySQL-Zieldatenbank verfügbaren Zeitzonennamen.</p> <p>Beispiel: <code>--extra-connection-attributes "Initstmt=SET time_zone= <i>US/Pacific</i> ;"</code></p>

Alternativ können Sie den Parameter `AfterConnectScript` des Befehls `--my-sql-settings` verwenden, um Fremdschlüsselprüfungen zu deaktivieren und die Zeitzone für Ihre Datenbank anzugeben.

Zieldatentypen für MySQL

Die folgende Tabelle zeigt die Zieldatentypen der MySQL-Datenbank, die bei der Verwendung unterstützt werden, AWS DMS und die Standardzuweisung von AWS DMS Datentypen.

Weitere Hinweise zu AWS DMS Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

AWS DMS Datentypen	MySQL-Datentypen
BOOLEAN	BOOLEAN
BYTES	Ist die Länge 1 bis 65.535, verwenden Sie VARBINARY (Länge). Ist die Länge 65.536 bis 2.147.483.647, verwenden Sie LONGLOB.
DATUM	DATUM
TIME	TIME
TIMESTAMP (ZEITSTEMPEL)	Ist die Skalierung => 0 und =< 6, dann: DATETIME (Skalierung) Ist die Skalierung => 7 und =< 9, dann: VARCHAR (37)
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (p,s)

AWS DMS Datentypen	MySQL-Datentypen
REAL4	FLOAT
REAL8	DOUBLE PRECISION
STRING	<p>Ist die Länge 1 bis 21.845, verwenden Sie VARCHAR (Länge).</p> <p>Ist die Länge 21.846 bis 2.147.483.647, verwenden Sie LONGTEXT.</p>
UINT1	UNSIGNED TINYINT
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	<p>Ist die Länge 1 bis 32.767, verwenden Sie VARCHAR (Länge).</p> <p>Ist die Länge 32.768 bis 2.147.483.647, verwenden Sie LONGTEXT.</p>
BLOB	<p>Ist die Länge 1 bis 65.535, verwenden Sie BLOB.</p> <p>Ist die Länge 65.536 bis 2.147.483.647, verwenden Sie LONGBLOB.</p> <p>Ist die Länge 0, verwenden Sie LONGBLOB (vollständige LOB-Unterstützung).</p>

AWS DMS Datentypen	MySQL-Datentypen
NCLOB	<p>Ist die Länge 1 bis 65.535, verwenden Sie TEXT.</p> <p>Ist die Länge 65.536 bis 2.147.483.647, verwenden Sie LONGTEXT mit ucs2 für CHARACTER SET.</p> <p>Ist die Länge 0, verwenden Sie LONGTEXT (vollständige LOB-Unterstützung) mit ucs2 für CHARACTER SET.</p>
CLOB	<p>Ist die Länge 1 bis 65.535, verwenden Sie TEXT.</p> <p>Ist die Länge 65.536 bis 2.147.483.647, verwenden Sie LONGTEXT.</p> <p>Ist die Länge 0, verwenden Sie LONGTEXT (vollständige LOB-Unterstützung).</p>

Verwenden einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service

Sie können Daten mithilfe von AWS Database Migration Service in Amazon-Redshift-Datenbanken migrieren. Amazon Redshift ist ein vollständig verwalteter Data-Warehouse-Service in Petabytegröße in der Cloud. Mit einer Amazon-Redshift-Datenbank als Ziel können Sie Daten aus allen anderen unterstützten Quelldatenbanken migrieren.

Sie können Amazon Redshift Serverless als Ziel für AWS DMS verwenden. Weitere Informationen finden Sie unter [Verwenden von AWS DMS mit Amazon Redshift Serverless als Ziel](#).

Der Amazon-Redshift-Cluster muss sich in demselben AWS-Konto und derselben AWS-Region wie die Replikations-Instance befinden.

Während einer Datenbankmigration zu Amazon Redshift verschiebt AWS DMS die Daten zunächst in einen Amazon-S3-Bucket. Wenn sich die Dateien in einem Amazon-S3-Bucket befinden, überträgt AWS DMS sie an die entsprechenden Tabellen im Amazon Redshift Data Warehouse. AWS DMS

erstellt den S3-Bucket in derselben AWS-Region, in der sich auch die Amazon-Redshift-Datenbank befindet. Die AWS DMS-Replikations-Instance muss sich in derselben AWS-Region befinden.

Wenn Sie mithilfe der AWS CLI oder der DMS-API Daten zu Amazon Redshift migrieren, müssen Sie eine AWS Identity and Access Management-(IAM)-Rolle einrichten, die den S3-Zugriff zulässt. Weitere Informationen zum Erstellen dieser IAM-Rolle finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Der Amazon-Redshift-Endpoint bietet eine vollständige Automatisierung für die folgenden Aufgaben:

- Schemagenerierung und Datentypzuweisung
- Vollständiges Laden von Quelldatenbanktabellen
- Inkrementelles Laden der an Quelltabellen vorgenommenen Änderungen
- Anwendung von Schemaänderungen in DDL (Data Definition Language), die an Quelltabellen vorgenommen wurden
- Synchronisierung zwischen vollständigen Ladevorgängen und CDC-Prozessen (Change Data Capture, Erfassung von Änderungsdaten)

AWS Database Migration Service unterstützt sowohl vollständige Ladevorgänge als auch Vorgänge zur Verarbeitung von Änderungen. AWS DMS liest die Daten aus der Quelldatenbank und erstellt eine Reihe von CSV-Dateien (Comma-Separated Value, durch Trennzeichen getrennt). Bei vollständigen Ladevorgängen erstellt AWS DMS Dateien für jede Tabelle. AWS DMS kopiert dann die Tabellendateien für jede Tabelle in einen separaten Ordner in Amazon S3. Wenn Dateien auf Amazon S3 hochgeladen werden, sendet AWS DMS einen Kopierbefehl und die in den Dateien enthaltenen Daten werden nach Amazon Redshift kopiert. Bei Vorgängen zur Verarbeitung von Änderungen kopiert AWS DMS die reinen Änderungen in die CSV-Dateien. Anschließend lädt AWS DMS die Dateien mit reinen Änderungen in Amazon S3 hoch und kopiert die Daten nach Amazon Redshift.

Weitere Informationen zum Arbeiten mit Amazon Redshift als Ziel für AWS DMS finden Sie in den folgenden Abschnitten:

Themen

- [Voraussetzungen für die Verwendung einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service](#)
- [Erforderliche Berechtigungen für die Verwendung von Redshift als Ziel](#)

- [Einschränkungen bei der Verwendung von Amazon Redshift als Ziel für AWS Database Migration Service](#)
- [Konfigurieren einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service](#)
- [Verwenden von Enhanced VPC Routing mit Amazon Redshift als Ziel für AWS Database Migration Service](#)
- [Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten](#)
- [Endpunkteinstellungen bei Verwendung von Amazon Redshift als Ziel für AWS DMS](#)
- [Verwenden eines Datenverschlüsselungsschlüssels und eines Amazon-S3-Buckets als Zwischenspeicher](#)
- [Multithread-Aufgabeneinstellungen für Amazon Redshift](#)
- [Zieldatentypen für Amazon Redshift](#)
- [Verwenden von AWS DMS mit Amazon Redshift Serverless als Ziel](#)

Voraussetzungen für die Verwendung einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service

In der folgenden Liste werden die erforderlichen Voraussetzungen für die Arbeit mit Amazon Redshift als Ziel für die Datenmigration beschrieben:

- Starten Sie einen Amazon-Redshift-Cluster über die AWS-Managementkonsole. Notieren Sie sich die grundlegenden Angaben zu Ihrem AWS-Konto und Ihrem Amazon-Redshift-Cluster, wie beispielsweise Ihr Passwort, Ihren Benutzernamen und den Datenbanknamen. Sie benötigen diese Werte beim Erstellen des Amazon-Redshift-Zielendpunkts.
- Der Amazon-Redshift-Cluster muss sich in demselben AWS-Konto und derselben AWS-Region wie die Replikations-Instance befinden.
- Die AWS DMS-Replikations-Instance benötigt Netzwerkkonnektivität zu dem Amazon-Redshift-Endpunkt (Hostname und Port), den Ihr Cluster verwendet.
- AWS DMS überträgt die Daten über einen Amazon-S3-Bucket an die Amazon-Redshift-Datenbank. Damit AWS DMS den Bucket erstellen kann, verwendet die Konsole eine IAM-Rolle, `dms-access-for-endpoint`. Wenn Sie die AWS CLI oder die DMS-API zum Erstellen einer Datenbankmigration mit Amazon Redshift als Zieldatenbank verwenden, müssen Sie diese IAM-Rolle erstellen. Weitere Informationen zum Erstellen dieser Rolle finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

- AWS DMS konvertiert BLOBs, CLOBs und NCLOBs in der Amazon-Redshift-Ziel-Instance in VARCHAR. Amazon Redshift unterstützt keine VARCHAR-Datentypen über 64 KB. Daher können Sie traditionelle LOBs in Amazon Redshift nicht speichern.
- Legen Sie die Ziel-Metadatenaufgabe [BatchApplyEnabled](#) auf `true` für AWS DMS fest, um Änderungen an Amazon Redshift-Zieltabellen während CDC zu verarbeiten. Ein Primärschlüssel für die Quell- und Zieltabelle ist erforderlich. Ohne Primärschlüssel werden Änderungen Anweisung für Anweisung angewendet. Dies kann sich negativ auf die Aufgabenleistung während der CDC auswirken, indem Ziellatenz verursacht wird und sich auf die Cluster-Commit-Warteschlange auswirkt.

Erforderliche Berechtigungen für die Verwendung von Redshift als Ziel

Verwenden Sie den Befehl `GRANT`, um Zugriffsberechtigungen für einen Benutzer oder eine Benutzergruppe zu definieren. Zu den Rechten gehören Zugriffsoptionen wie das Lesen von Daten in Tabellen und Ansichten, das Schreiben von Daten und das Erstellen von Tabellen. Weitere Informationen über die Verwendung von `GRANT` mit Amazon Redshift finden Sie unter [GRANT](#) im Datenbankentwicklerhandbuch zu Amazon Redshift.

Im Folgenden ist die Syntax zum Erteilen von bestimmten Berechtigungen für eine Tabelle, eine Datenbank, ein Schema, eine Funktion oder eine Prozedur oder von Berechtigungen auf Sprachebene für Amazon-Redshift-Tabellen und -Ansichten dargestellt.

```
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 } [, ...]
```

Im Folgenden finden Sie die Syntax für Berechtigungen auf Spaltenebene für Amazon-Redshift-Tabellen und -Ansichten.

```
GRANT { { SELECT | UPDATE } ( column_name [, ...] ) [, ...] | ALL [ PRIVILEGES ]
    ( column_name [, ...] ) }
    ON { [ TABLE ] table_name [, ...] }
    TO { username | GROUP group_name | PUBLIC } [, ...]
```

Im Folgenden finden Sie die Syntax für das ASSUMEROLE-Recht, das Benutzern und Gruppen mit einer spezifizierten Rolle gewährt wird.

```
GRANT ASSUMEROLE
    ON { 'iam_role' [, ...] | ALL }
    TO { username | GROUP group_name | PUBLIC } [, ...]
    FOR { ALL | COPY | UNLOAD } [, ...]
```

Einschränkungen bei der Verwendung von Amazon Redshift als Ziel für AWS Database Migration Service

Die folgenden Einschränkungen gelten, wenn Sie eine Amazon-Redshift-Datenbank als Ziel verwenden:

- Aktivieren Sie nicht die Versionsverwaltung für den S3-Bucket, den Sie als Zwischenspeicher für Ihr Amazon-Redshift-Ziel verwenden. Wenn Sie eine S3-Versionsverwaltung benötigen, verwenden Sie Lebenszyklusrichtlinien, um alte Versionen aktiv zu löschen. Andernfalls kann es aufgrund eines Timeouts eines `list-object`-Aufrufs in S3 zu Verbindungsfehlern beim Endpunkttest kommen. Informationen zum Erstellen einer Lebenszyklusrichtlinie für einen S3-Bucket finden Sie unter [Verwalten Ihres Speicher-Lebenszyklus](#). Informationen zum Löschen einer Version eines S3-Objekts finden Sie unter [Löschen von Objekten aus einem versionsverwaltungsfähigen Bucket](#).
- Die folgende DDL wird nicht unterstützt:


```
ALTER TABLE table name MODIFY COLUMN column name data type;
```

- AWS DMS kann keine Änderungen in ein Schema mit einem Namen migrieren oder replizieren, der mit einem Unterstrich (_) beginnt. Verwenden Sie für Schemata, deren Namen mit einem Unterstrich beginnen, Zuweisungstransformationen, um das Schema im Ziel umzubenennen.
- Amazon Redshift unterstützt keine VARCHARs über 64 KB. Daher können LOBs aus traditionellen Datenbanken in Amazon Redshift nicht gespeichert werden.
- Das Anwenden einer DELETE-Anweisung auf eine Tabelle mit einem mehrspaltigen Primärschlüssel wird nicht unterstützt, wenn einer der Primärschlüsselspaltenamen ein reserviertes Wort verwendet. Eine Liste der reservierten Wörter in Amazon Redshift finden Sie [hier](#).
- Es kann zu Leistungsproblemen kommen, wenn Ihr Quellsystem UPDATE-Operationen für den Primärschlüssel einer Quelltable ausführt. Diese Leistungsprobleme treten auf, wenn Änderungen auf das Ziel angewendet werden. Das liegt daran, dass UPDATE- (und DELETE-)Operationen auf den Primärschlüsselwert angewiesen sind, um die Zielzeile zu identifizieren. Wenn Sie den Primärschlüssel einer Quelltable aktualisieren, enthält Ihr Aufgabenprotokoll Meldungen wie die folgende:

```
Update on table 1 changes PK to a PK that was previously updated in the same bulk update.
```

- DMS unterstützt keine benutzerdefinierten DNS-Namen, wenn Sie einen Endpunkt für einen Redshift-Cluster konfigurieren. Sie müssen den von Amazon bereitgestellten DNS-Namen verwenden. Da sich der Amazon-Redshift-Cluster in demselben AWS-Konto und derselben Region wie die Replikations-Instance befinden muss, schlägt die Validierung fehl, wenn Sie einen benutzerdefinierten DNS-Endpunkt verwenden.
- In Amazon Redshift gilt ein standardmäßiges Timeout für inaktive Sitzungen von 4 Stunden. Wenn im Rahmen der DMS-Replikationsaufgabe keine Aktivität stattfindet, unterbricht Redshift die Sitzung nach 4 Stunden. Fehler können darauf zurückzuführen sein, dass DMS keine Verbindung herstellen kann und möglicherweise neu gestartet werden muss. Um das Problem zu umgehen, legen Sie für den DMS-Replikationsbenutzer einen Grenzwert für SESSION TIMEOUT von mehr als 4 Stunden fest. Sie können sich auch die Beschreibung von [ALTER USER](#) im Datenbankentwicklerhandbuch für Amazon Redshift ansehen.
- Wenn AWS DMS Quelltabellendaten ohne einen Primärschlüssel oder eindeutigen Schlüssel repliziert, kann die CDC-Latenz hoch und das Leistungsniveau damit inakzeptabel sein.

Konfigurieren einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service

AWS Database Migration Service muss für die Verwendung mit der Amazon-Redshift-Instance konfiguriert werden. Die folgende Tabelle beschreibt die verfügbaren Konfigurationseigenschaften für den Amazon-Redshift-Endpunkt.

Property (Eigenschaft)	Description (Beschreibung)
server	Der Name des verwendeten Amazon-Redshift-Clusters.
port	Die Portnummer für Amazon Redshift. Der Standardwert lautet 5439.
username	Ein Amazon-Redshift-Benutzername eines registrierten Benutzers.
password	Das Passwort für den Benutzer, der in der Eigenschaft "Benutzername" angegeben ist.
Datenbank	Der Name des Amazon Redshift Data Warehouse (Service), mit dem Sie arbeiten.

Wenn Sie zusätzliche Attribute für Verbindungszeichenfolgen zu Ihrem Amazon-Redshift-Endpunkt hinzufügen möchten, können Sie die Attribute `maxFileSize` und `fileTransferUploadStreams` festlegen. Weitere Informationen zu diesen Attributen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon Redshift als Ziel für AWS DMS](#).

Verwenden von Enhanced VPC Routing mit Amazon Redshift als Ziel für AWS Database Migration Service

Wenn Sie Enhanced VPC Routing mit Ihrem Amazon-Redshift-Ziel verwenden, wird der gesamte COPY-Datenverkehr zwischen Ihrem Amazon-Redshift-Cluster und Ihren Daten-Repositories über Ihre VPC geleitet. Da Enhanced VPC Routing sich darauf auswirkt, wie Amazon Redshift auf andere Ressourcen zugreift, schlagen COPY-Befehle möglicherweise fehl, wenn die VPC nicht ordnungsgemäß konfiguriert wurde.

AWS DMS kann durch dieses Verhalten beeinflusst werden, da es mit dem COPY-Befehl Daten von S3 in einen Amazon-Redshift-Cluster verschiebt.

Im Folgenden sind die von AWS DMS ausgeführten Schritte zum Laden von Daten in ein Amazon Redshift-Ziel angegeben:

1. AWS DMS kopiert Daten von der Quelle in CSV-Dateien auf dem Replikationsserver.
2. AWS DMS kopiert die CSV-Dateien mithilfe des AWS-SDK in einen S3-Bucket auf Ihrem Konto.
3. Anschließend verwendet AWS DMS den COPY-Befehl in Amazon Redshift zum Kopieren von Daten aus den CSV-Dateien in S3 in eine entsprechende Tabelle in Amazon Redshift.

Wenn Enhanced VPC Routing nicht aktiviert ist, leitet Amazon Redshift den Datenverkehr über das Internet weiter, einschließlich des Datenverkehrs an andere Services im AWS-Netzwerk. Bei deaktivierter Funktion müssen Sie den Netzwerkpfad nicht konfigurieren. Wenn die Funktion aktiviert ist, müssen Sie speziell einen Netzwerkpfad zwischen Ihrem VPC-Cluster und Ihren Datenressourcen erstellen. Weitere Informationen zu der erforderlichen Konfiguration finden Sie unter [Enhanced VPC Routing](#) in der Amazon-Redshift-Dokumentation.

Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten

Sie können Ihre an Amazon S3 übertragenen Zieldaten verschlüsseln, bevor Sie diese nach Amazon Redshift kopieren. Um dies zu tun, können Sie benutzerdefinierte AWS KMS-Schlüssel erstellen und verwenden. Sie können den Schlüssel verwenden, den Sie zur Verschlüsselung Ihrer Zieldaten erstellt haben. Verwenden Sie dazu beim Erstellen des Amazon-Redshift-Zielendpunkts einen der folgenden Mechanismen:

- Verwenden Sie die folgende Option beim Ausführen des `create-endpoint`-Befehls mithilfe der AWS CLI.

```
--redshift-settings '{"EncryptionMode": "SSE_KMS", "ServerSideEncryptionKmsKeyId":  
  "your-kms-key-ARN"}'
```

Hier handelt es sich bei *your-kms-key-ARN* um den Amazon-Ressourcennamen (ARN) für Ihren KMS-Schlüssel. Weitere Informationen finden Sie unter [Verwenden eines Datenverschlüsselungsschlüssels und eines Amazon-S3-Buckets als Zwischenspeicher](#).

- Richten Sie für das zusätzliche Verbindungsattribut `encryptionMode` den Wert `SSE_KMS` und für das zusätzliche Verbindungsattribut `serverSideEncryptionKmsKeyId` den ARN für Ihren KMS-Schlüssel ein. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon Redshift als Ziel für AWS DMS](#).

Um Amazon-Redshift-Zieldaten mithilfe eines KMS-Schlüssels zu verschlüsseln, benötigen Sie eine AWS Identity and Access Management (IAM)-Rolle mit Berechtigungen für den Zugriff auf Amazon-Redshift-Daten. Auf diese IAM-Rolle wird dann in einer Richtlinie (einer Schlüsselrichtlinie) zugegriffen, die dem von Ihnen erstellten Verschlüsselungsschlüssel angefügt ist. Sie können diesen Vorgang in Ihrer IAM-Konsole ausführen, indem Sie Folgendes erstellen:

- Eine IAM-Rolle mit einer von AWS verwalteten Richtlinie.
- Einen KMS-Schlüssel mit einer Schlüsselrichtlinie, die sich auf diese Rolle bezieht.


Die entsprechende Vorgehensweise wird nachfolgend beschrieben.

So erstellen Sie eine IAM-Rolle mit der erforderlichen von AWS verwalteten Richtlinie

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Rollen aus. Die Seite Roles (Rollen) wird geöffnet.
3. Wählen Sie Rolle erstellen aus. Die Seite Create role (Rolle erstellen) wird geöffnet.
4. Wenn AWS-Service als vertrauenswürdige Entität ausgewählt ist, wählen Sie DMS als den Service aus, der die Rolle verwendet.
5. Wählen Sie Weiter: Berechtigungen aus. Die Seite Attach permissions policies (Berechtigungsrichtlinien anfügen) wird angezeigt.
6. Suchen und wählen Sie die AmazonDMSRedshiftS3Role-Richtlinie aus.
7. Wählen Sie Next: Markierungen (Weiter: Markierungen). Die Seite Add tags (Tags hinzufügen) wird angezeigt. Hier können Sie alle gewünschten Tags hinzufügen.
8. Wählen Sie Next: Review (Weiter: Prüfen) aus und überprüfen Sie Ihre Ergebnisse.
9. Wenn die Einstellungen Ihren Vorstellungen entsprechen, geben Sie einen Namen für die Rolle (z. B. `DMS-Redshift-endpoint-access-role`) sowie eine zusätzliche Beschreibung ein und wählen Sie anschließend Create role (Rolle erstellen) aus. Die Seite Roles (Rollen) wird geöffnet und zeigt eine Mitteilung an, die darauf hinweist, dass Ihre Rolle erstellt wurde.


Sie haben jetzt die neue Rolle erstellt, über die Sie auf Amazon-Redshift-Ressourcen zugreifen können, um diese mit einem angegebenen Namen, z. B. `DMS-Redshift-endpoint-access-role`, zu verschlüsseln.

So erstellen Sie einen AWS KMS-Verschlüsselungsschlüssel mit einer Schlüsselrichtlinie, die sich auf Ihre IAM-Rolle bezieht

 Note

Weitere Hinweise zur Funktionsweise von AWS DMS mit AWS KMS-Verschlüsselungsschlüsseln finden Sie unter [Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen](#).

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter <https://console.aws.amazon.com/kms>.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key. Die Seite Configure key (Schlüssel konfigurieren) wird geöffnet.
5. Wählen Sie für Key type (Schlüsseltyp) Symmetric (Symmetrisch).

 Note

Wenn Sie diesen Schlüssel erstellen, können Sie nur einen symmetrischen Schlüssel erstellen, da alle AWS-Services wie z. B. Amazon Redshift, nur mit symmetrischen Verschlüsselungsschlüsseln arbeiten.

6. Wählen Sie Erweiterte Optionen aus. Stellen Sie für Key material origin (Schlüsselmaterialherkunft) sicher, dass KMS ausgewählt ist, und wählen Sie dann Next (Weiter). Die Seite Add Labels (Etiketten hinzufügen) wird geöffnet.
7. Geben Sie unter Create alias and description (Alias und Beschreibung erstellen) einen Alias für den Schlüssel (z. B. DMS-Redshift-endpoint-encryption-key) und eventuell eine weitere Beschreibung ein.
8. Fügen Sie unter Tags alle Tags hinzu, die Sie verwenden möchten, um den Schlüssel zu identifizieren und seine Nutzung zu verfolgen, und wählen Sie anschließend Next Step (Nächster Schritt) aus. Die Seite Define key administrative permissions (Schlüsselverwaltungsberechtigungen definieren) wird geöffnet und zeigt eine Liste der Benutzer und Rollen an, aus denen Sie auswählen können.

9. Fügen Sie die Benutzer und Rollen hinzu, die den Schlüssel verwalten sollen. Stellen Sie sicher, dass diese Benutzer und Rollen über die erforderlichen Berechtigungen verfügen, um den Schlüssel zu verwalten.
10. Wählen Sie unter Key deletion (Schlüssellöschung) aus, ob die Schlüsseladministratoren den Schlüssel löschen können, und klicken Sie anschließend auf Next Step (Nächster Schritt). Die Seite Define Key usage permissions (Schlüsselnutzungsberechtigungen definieren) wird geöffnet und zeigt eine weitere Liste mit Benutzern und Rollen an, aus denen Sie auswählen können.
11. Wählen Sie unter Dieses Konto die verfügbaren Benutzer aus, die kryptografische Operationen für Amazon-Redshift-Ziele ausführen sollen. Wählen Sie zudem die Rolle aus, die Sie zuvor unter Rollen erstellt haben, um den Zugriff zur Verschlüsselung von Amazon-Redshift-Zielobjekten zu ermöglichen (z. B. `DMS-Redshift-endpoint-access-role`).
12. Wenn Sie weitere Konten hinzufügen möchten, die nicht für denselben Zugriff aufgeführt sind, wählen Sie unter Other AWS accounts (Andere Konten) die Option Add another AWS account (Anderes Konto hinzufügen) und dann Next (Weiter). Die Seite Review and edit key policy (Schlüsselrichtlinie prüfen und bearbeiten) wird geöffnet. Hier wird das JSON für die Schlüsselrichtlinie angezeigt, die Sie überprüfen und bearbeiten können, indem Sie in den vorhandenen JSON-Inhalt schreiben. Hier können Sie sehen, wo die Schlüsselrichtlinie auf die Rolle und die Benutzer (z. B. `Admin` und `User1`) verweist, die Sie im vorherigen Schritt ausgewählt haben. Sie können auch die verschiedenen Schlüsselaktionen sehen, die für die verschiedenen Prinzipale (Benutzer und Rollen) zulässig sind, wie im folgenden Beispiel dargestellt.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
```

```

    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:role/Admin"
      ]
    },
    "Action": [
      "kms:Create*",
      "kms:Describe*",
      "kms:Enable*",
      "kms:List*",
      "kms:Put*",
      "kms:Update*",
      "kms:Revoke*",
      "kms:Disable*",
      "kms:Get*",
      "kms>Delete*",
      "kms:TagResource",
      "kms:UntagResource",
      "kms:ScheduleKeyDeletion",
      "kms:CancelKeyDeletion"
    ],
    "Resource": "*"
  },
  {
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:role/DMS-Redshift-endpoint-access-role",
        "arn:aws:iam::111122223333:role/Admin",
        "arn:aws:iam::111122223333:role/User1"
      ]
    },
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  },
  {

```

```

    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:role/DMS-Redshift-endpoint-access-role",
        "arn:aws:iam::111122223333:role/Admin",
        "arn:aws:iam::111122223333:role/User1"
      ]
    },
    "Action": [
      "kms:CreateGrant",
      "kms:ListGrants",
      "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {
      "Bool": {
        "kms:GrantIsForAWSResource": true
      }
    }
  }
]

```

13. Wählen Sie Finish (Abschließen). Die Seite Verschlüsselungsschlüssel wird geöffnet und zeigt eine Mitteilung an, die darauf hinweist, dass Ihr AWS KMS key erstellt wurde.

Damit haben Sie einen neuen KMS-Schlüssel mit einem angegebenen Alias (z. B. DMS-Redshift-endpoint-encryption-key) erstellt. Dieser Schlüssel ermöglicht es AWS DMS, Amazon-Redshift-Zieldaten zu verschlüsseln.

Endpunkteinstellungen bei Verwendung von Amazon Redshift als Ziel für AWS DMS


Sie können Endpunkteinstellungen zur Konfiguration Ihrer Amazon-Redshift-Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen beim Erstellen des Zielendpunkts über die AWS DMS-Konsole oder über den Befehl `create-endpoint` in der [AWS CLI](#) mit der JSON-Syntax `--redshift-settings '{"EndpointSetting": "value", ...}'` an.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Amazon Redshift als Ziel verwenden können.

Name	Beschreibung
MaxFileSize	<p>Gibt die maximale Größe (in KB) einer CSV-Datei zum Übertragen von Daten nach Amazon Redshift an.</p> <p>Standardwert: 32768 KB (32 MB)</p> <p>Gültige Werte: 1 bis 1 048 576</p> <p>Beispiel: <code>--redshift-settings '{"MaxFileSize": 512}'</code></p>
FileTransferUploadStreams	<p>Gibt die Anzahl von Threads an, die verwendet werden, um eine Datei hochzuladen.</p> <p>Standardwert: 10</p> <p>Gültige Werte: 1 bis 64</p> <p>Beispiel: <code>--redshift-settings '{"FileTransferUploadStreams": 20}'</code></p>
Acceptanydate	<p>Gibt an, ob ein beliebiges Datumsformat akzeptiert wird, einschließlich ungültiger Datumsformate wie 0000-00-00-00. Boolescher Wert:</p> <p>Standardwert: false</p> <p>Zulässige Werte: true false</p> <p>Beispiel: <code>--redshift-settings '{"Acceptanydate": true}'</code></p>
Dateformat	<p>Gibt das Datumsformat an. Dies ist eine Zeichenfolge-Eingabe und standardmäßig leer. Das Standardformat ist JJJJ-MM-TT, aber Sie können es ändern, z. B. in TT-MM-JJJJ. Wenn Ihre Datums- oder Zeitwerte unterschiedliche Formate verwenden, verwenden Sie das <code>auto</code>-Argument mit dem <code>Dateformat</code> -Parameter. Das Argument <code>auto</code> erkennt verschiedene Formate, die bei der Verwendung</p>

Name	Beschreibung
	<p>g einer Dateiformat Zeichenfolge nicht unterstützt werden. Das Schlüsselwort auto unterscheidet zwischen Groß- und Kleinschreibung.</p> <p>Standardwert: leer</p> <p>Gültige Werte: <i>dateformat_string</i> " oder auto</p> <p>Beispiel: <code>--redshift-settings '{"Dateformat": "auto"}'</code></p>
Timeformat	<p>Gibt das Zeitformat an. Dies ist eine Zeichenfolge-Eingabe und standardmäßig leer. Das Argument auto erkennt verschiedene Formate, die bei der Verwendung einer Timeformat -Zeichenfolge nicht unterstützt werden. Wenn Ihre Datums- und Zeitwerte unterschiedliche Formate verwenden, verwenden Sie das Argument auto mit dem Timeformat -Parameter.</p> <p>Standardwert: 10</p> <p>Gültige Werte: <i>Timeformat_string</i> " "auto" "epochsecs" "epochmillisecs"</p> <p>Beispiel: <code>--redshift-settings '{"Timeformat": "auto"}'</code></p>
Emptyasnull	<p>Gibt an, ob AWS DMS leere CHAR- und VARCHAR-Felder als Null migrieren soll. Ein Wert von true stellt leere CHAR- und VARCHAR-Felder auf Null ein.</p> <p>Standardwert: false</p> <p>Zulässige Werte: true false</p> <p>Beispiel: <code>--redshift-settings '{"Emptyasnull": true}'</code></p>

Name	Beschreibung
TruncateColumns	<p>Schneidet Daten in Spalten auf die entsprechende Zahl von Zeichen ab, sodass sie der Spaltenspezifikation entsprechen. Gilt nur für Spalten mit dem Datentyp VARCHAR und CHAR und Zeilen mit einer Größe von 4 MB oder weniger.</p> <p>Standardwert: false</p> <p>Zulässige Werte: true false</p> <p>Beispiel: <code>--redshift-settings '{"TruncateColumns": true}'</code></p>
RemoveQuotes	<p>Entfernt in den eingehenden Daten Anführungszeichen, die Zeichenfolgen umschließen. Alle Zeichen innerhalb der Anführungszeichen einschließlich Trennzeichen bleiben bewahrt. Weitere Informationen über das Entfernen von Anführungszeichen für ein Amazon-Redshift-Ziel finden Sie im Datenbankentwicklerhandbuch zu Amazon Redshift.</p> <p>Standardwert: false</p> <p>Zulässige Werte: true false</p> <p>Beispiel: <code>--redshift-settings '{"RemoveQuotes": true}'</code></p>
TrimBlanks	<p>Entfernt Leerzeichen am Ende von VARCHAR-Zeichenfolgen. Dieser Parameter gilt nur für die Spalten mit dem Datentyp VARCHAR.</p> <p>Standardwert: false</p> <p>Zulässige Werte: true false</p> <p>Beispiel: <code>--redshift-settings '{"TrimBlanks": true}'</code></p>

Name	Beschreibung
EncryptionMode	<p>Gibt den serverseitigen Verschlüsselungsmodus an, den Sie verwenden möchten, um Ihre Daten per Push nach S3 zu übertragen, bevor sie nach Amazon Redshift kopiert werden. Die gültigen Werte sind SSE_S3 (S3-serverseitige Verschlüsselung) oder SSE_KMS (KMS-Verschlüsselung). Wenn Sie SSE_KMS auswählen, legen Sie den <code>ServerSideEncryptionKmsKeyId</code> -Parameter auf den Amazon-Ressourcennamen (ARN) fest, so dass der KMS-Schlüssel für die Verschlüsselung verwendet wird.</p> <div data-bbox="688 737 1507 1146" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> Note</p><p>Sie können auch den CLI-Befehl <code>modify-endpoint</code> verwenden, um den Wert der Einstellung <code>EncryptionMode</code> für einen vorhandenen Endpunkt von SSE_KMS in SSE_S3 zu ändern. Sie können jedoch nicht den Wert für <code>EncryptionMode</code> von SSE_S3 in SSE_KMS ändern.</p></div> <p>Standardwert: SSE_S3</p> <p>Gültige Werte: SSE_S3 oder SSE_KMS.</p> <p>Beispiel: <code>--redshift-settings '{"EncryptionMode": "SSE_S3"}</code></p>

Name	Beschreibung
ServerSideEncryptionKmsKeyId	<p>Wenn Sie für EncryptionMode SSE_KMS festlegen , setzen Sie diesen Parameter auf den ARN für den KMS-Schlüssel. Sie finden diesen ARN, indem Sie den Schlüssel-Alias in der Liste der AWS KMS-Schlüssel, die für Ihr Konto erstellt wurden, auswählen. Wenn Sie den Schlüssel erstellen, müssen Sie ihm auch bestimmte Richtlinien und Rollen zuweisen. Weitere Informationen finden Sie unter Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Ziel Daten.</p> <p>Beispiel: <code>--redshift-settings '{"ServerSideEncryptionKmsKeyId":"arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'</code></p>
EnableParallelBatchInMemoryCSVFiles	<p>Die Einstellung EnableParallelBatchInMemoryCSVFiles verbessert die Leistung umfangreicherer Aufgaben für vollständige Multithread-Ladevorgänge, da DMS auf die Festplatte und nicht in den Arbeitsspeicher schreibt. Der Standardwert ist false.</p>
CompressCsvFiles	<p>Verwenden Sie dieses Attribut zum Komprimieren von Daten, die während der Migration an ein Amazon-Redshift-Ziel gesendet werden. Der Standardwert ist true und die Komprimierung ist standardmäßig aktiviert.</p>

Verwenden eines Datenverschlüsselungsschlüssels und eines Amazon-S3-Buckets als Zwischenspeicher

Sie können die Einstellungen für den Amazon-Redshift-Zielendpunkt verwenden, um Folgendes zu konfigurieren:

- Ein benutzerdefinierter AWS KMS-Datenverschlüsselungsschlüssel. Sie können dann diesen Schlüssel verwenden, um Ihre an Amazon S3 übertragenen Daten zu verschlüsseln, bevor sie nach Amazon Redshift kopiert werden.
- Einen benutzerdefinierten S3-Bucket als Zwischenspeicher für die zu Amazon Redshift migrierten Daten.
- Mapping von BOOLEAN als BOOLEAN aus einer PostgreSQL-Quelle. Standardmäßig wird ein BOOLEAN-Typ als varchar(1) migriert. Sie können `MapBooleanAsBoolean` angeben, damit das Redshift-Ziel den BOOLEAN-Typ als BOOLEAN migriert, wie im folgenden Beispiel gezeigt.

```
--redshift-settings '{"MapBooleanAsBoolean": true}'
```

Beachten Sie, dass Sie diese Einstellung sowohl am Quell- als auch am Zielendpunkt festlegen müssen, damit sie wirksam wird.

KMS-Schlüsseleinstellungen für die Datenverschlüsselung

Die folgenden Beispiele veranschaulichen die Konfiguration eines benutzerdefinierten KMS-Schlüssels, um Ihre per Push an S3 übertragenen Daten zu verschlüsseln. Um zu beginnen, können Sie den folgenden `create-endpoint`-Aufruf in der AWS CLI ausführen.

```
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"}'
```

Das durch die `--redshift-settings`-Option angegebene JSON-Objekt definiert zwei Parameter. Der eine ist ein `EncryptionMode`-Parameter mit dem Wert `SSE_KMS`. Der andere ist ein `ServerSideEncryptionKmsKeyId`-Parameter mit dem Wert `arn:aws:kms:us-east-1:111122223333:key/24c3c5a1-f34a-4519-a85b-2debbef226d1`. Bei diesem Wert handelt es sich um einen Amazon-Ressourcenname (ARN) für Ihren benutzerdefinierten KMS-Schlüssel.

Standardmäßig erfolgt S3-Datenverschlüsselung mithilfe von serverseitiger S3-Verschlüsselung. Für das Amazon-Redshift-Ziel im vorherigen Beispiel ist dies gleichbedeutend mit der Angabe der Endpunkteinstellungen, wie im folgenden Beispiel dargestellt.

```
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"}'
```

Weitere Informationen zur Verwendung der serverseitigen S3-Verschlüsselung finden Sie unter [Schutz von Daten mittels serverseitiger Verschlüsselung](#) im Amazon-Simple-Storage-Service-Benutzerhandbuch.

Note

Sie können auch den CLI-Befehl `modify-endpoint` verwenden, um den Wert des Parameters `EncryptionMode` für einen vorhandenen Endpunkt von `SSE_KMS` in `SSE_S3` zu ändern. Sie können jedoch nicht den Wert für `EncryptionMode` von `SSE_S3` in `SSE_KMS` ändern.

Einstellungen von Amazon-S3-Buckets

Wenn Sie Daten zu einem Amazon-Redshift-Zielendpunkt migrieren, verwendet AWS DMS einen standardmäßigen Amazon-S3-Bucket als Zwischenspeicher für Aufgaben, bevor die migrierten Daten nach Amazon Redshift kopiert werden. Beispielsweise wird in den dargelegten Beispielen für das Erstellen eines Amazon-Redshift-Zielendpunkts mit einem AWS KMS-Datenverschlüsselungsschlüssel dieser Standard-S3-Bucket verwendet (siehe [KMS-Schlüsseleinstellungen für die Datenverschlüsselung](#)).

Sie können stattdessen einen benutzerdefinierten S3-Bucket für diese Zwischenspeicherung angeben, indem Sie die folgenden Parameter in den Wert Ihrer `--redshift-settings`-Option des AWS CLI-Befehls „`create-endpoint`“ einschließen:

- `BucketName` – Eine Zeichenfolge, die Sie als Namen des S3-Bucket-Speichers angeben. Wenn Ihre Servicezugriffsrolle auf der Richtlinie `AmazonDMSRedshiftS3Role` basiert, muss dieser Wert das Präfix `dms-` haben, also beispielsweise `dms-my-bucket-name` lauten.
- `BucketFolder` – (Optional) Eine Zeichenfolge die Sie als Namen des Speicherordners im angegebenen S3-Bucket angeben können.
- `ServiceAccessRoleArn` – Der ARN einer IAM-Rolle, die administrativen Zugriff auf den S3-Bucket erlaubt. In der Regel erstellen Sie diese Rolle auf Grundlage der

AmazonDMSRedshiftS3Role-Richtlinie. Ein Beispiel finden Sie unter dem Vorgehen zum Erstellen einer IAM-Rolle mit der erforderlichen von AWS verwalteten Richtlinie in [Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten](#).

Note

Wenn Sie mithilfe der `--service-access-role-arn` Option des `create-endpoint`-Befehls den ARN einer anderen IAM-Rolle angeben, hat diese IAM-Rollenoption Priorität.

Das folgende Beispiel zeigt, wie Sie diese Parameter verwenden können, um einen benutzerdefinierten Amazon-S3-Bucket im folgenden Aufruf `create-endpoint` über die AWS CLI anzugeben.

```
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"}'
```

Multithread-Aufgabeneinstellungen für Amazon Redshift

Sie können die Leistung von Aufgaben zum vollständigen Laden und CDC-Aufgaben (CDC = Change Data Capture) für einen Amazon-Redshift-Zielendpunkt verbessern, indem Sie Multithread-Aufgabeneinstellungen verwenden. Auf diese Weise können Sie die Anzahl der gleichzeitigen Threads und die Anzahl der Datensätze angeben, die in einem Puffer gespeichert werden sollen.

Aufgabeneinstellungen für vollständige Multithread-Ladevorgänge für Amazon Redshift

Zur Förderung der Leistung beim vollständigen Laden können Sie die folgenden `ParallelLoad*`-Aufgabeneinstellungen verwenden:

- `ParallelLoadThreads` – Gibt die Anzahl der gleichzeitigen Threads an, die DMS während eines vollständigen Ladevorgangs verwendet, um Datensätze an einen Amazon-Redshift-Zielendpunkt zu übertragen. Der Standardwert ist Null (0) und der maximale Wert ist 32. Weitere Informationen finden Sie unter [Aufgabeneinstellungen für vollständiges Laden](#).

Das Attribut `enableParallelBatchInMemoryCSVFiles` kann auf `false` gesetzt sein, wenn Sie die `ParallelLoadThreads`-Aufgabeneinstellung verwenden. Das Attribut verbessert die

Leistung umfangreicherer Aufgaben für vollständige Multithread-Ladevorgänge, da DMS auf die Festplatte und nicht in den Arbeitsspeicher schreibt. Der Standardwert ist `true`.

- `ParallelLoadBufferSize` – Gibt die maximale Anzahl von Datensatzanforderungen bei Verwendung von Threads für paralleles Laden mit dem Redshift-Ziel an. Der Standardwert ist 100 und der maximale Wert 1 000. Wir empfehlen, diese Option zu verwenden, wenn `ParallelLoadThreads > 1` (größer als eins) ist.

Note

Unterstützung für die Verwendung von `ParallelLoad*`-Aufgabeneinstellungen während FULL-LOAD-Vorgängen zu Amazon-Redshift-Zielendpunkten ist in AWS DMS Version 3.4.5 und höher verfügbar.

Die Redshift-Endpunkteinstellung `ReplaceInvalidChars` wird nicht für die Verwendung während eines CDC-Vorgangs (Change Data Capture) oder während einer FULL-LOAD-Migrationsaufgabe mit aktiviertem parallelem Laden unterstützt. Sie wird für die FULL-LOAD-Migration unterstützt, wenn paralleles Laden nicht aktiviert ist. Weitere Informationen finden Sie unter [RedShiftSettings](#) in der API-Referenz zu AWS Database Migration Service

Multithread-CDC-Aufgabeneinstellungen für Amazon Redshift

Zur Förderung der CDC-Leistung können Sie die folgenden `ParallelApply*`-Aufgabeneinstellungen verwenden:

- `ParallelApplyThreads` – Gibt die Anzahl der gleichzeitigen Threads an, die AWS DMS während eines CDC-Ladevorgangs verwendet, um Datensätze an einen Amazon-Redshift-Zielendpunkt zu übertragen. Der Standardwert ist Null (0) und der maximale Wert ist 32. Der empfohlene Mindestwert entspricht der Anzahl der Slices in Ihrem Cluster.
- `ParallelApplyBufferSize` – Gibt die maximale Anzahl von Datensatzanforderungen bei Verwendung von Threads für parallele Anwendung mit dem Redshift-Ziel an. Der Standardwert ist 100 und der maximale Wert 1 000. Wir empfehlen, diese Option zu verwenden, wenn `ParallelApplyThreads > 1` (größer als eins) ist.

Um Redshift als Ziel optimal zu nutzen, sollte der Wert von `ParallelApplyBufferSize` mindestens dem Zweifachen der Anzahl von `ParallelApplyThreads` entsprechen (also mindestens doppelt so groß sein).

Note

Unterstützung für die Verwendung von ParallelApply*-Aufgabeneinstellungen während CDC-Vorgängen zu Amazon-Redshift-Zielendpunkten ist in AWS DMS Version 3.4.3 und höher verfügbar.

Der Grad der angewandten Parallelität hängt von der Korrelation zwischen der gesamten Stapelgröße und der maximalen Dateigröße ab, die für die Übertragung von Daten verwendet wird. Wenn Sie Multithread-CDC-Aufgabeneinstellungen mit einem Redshift-Ziel verwenden, erzielen Sie Vorteile, wenn die Stapelgröße im Verhältnis zur maximalen Dateigröße groß ist. Sie können beispielsweise die folgende Kombination aus Endpunkt- und Aufgabeneinstellungen verwenden, um eine optimale Leistung zu erzielen.

```
// Redshift endpoint setting

    MaxFileSize=250000;

// Task settings

    BatchApplyEnabled=true;
    BatchSplitSize =8000;
    BatchApplyTimeoutMax =1800;
    BatchApplyTimeoutMin =1800;
    ParallelApplyThreads=32;
    ParallelApplyBufferSize=100;
```

Bei Verwendung der Einstellungen im vorherigen Beispiel profitiert ein Kunde mit einer hohen Transaktions-Workload von der Tatsache, dass sein Stapelpuffer, der 8 000 Datensätze enthält, innerhalb von 1 800 Sekunden gefüllt wird und 32 parallele Threads mit einer maximalen Dateigröße von 250 MB verwendet werden.

Weitere Informationen finden Sie unter [Einstellungen für die Optimierung der Verarbeitung von Änderungen](#).

Note

DMS-Abfragen, die während der laufenden Replikation auf einen Redshift-Cluster ausgeführt werden, können dieselbe WLM-Warteschlange (WLM = Workload Management)

gemeinsam mit anderen laufenden Anwendungsabfragen nutzen. Daher sollten Sie die WLM-Eigenschaften ordnungsgemäß konfigurieren, um die Leistung während der laufenden Replikation auf ein Redshift-Ziel zu beeinflussen. Wenn beispielsweise andere parallele ETL-Abfragen ausgeführt werden, läuft DMS langsamer und die Leistungsverbesserungen gehen verloren.

Ziel datentypen für Amazon Redshift

Der Amazon-Redshift-Endpunkt für AWS DMS unterstützt die meisten Amazon-Redshift-Datentypen. In der folgenden Tabelle sind die Amazon Redshift-Ziel datentypen, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuordnung von AWS DMS-Datentypen aufgeführt.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

AWS DMS-Datentypen	Amazon-Redshift-Datentypen
BOOLEAN	BOOL
BYTES	VARCHAR (Länge)
DATUM	DATUM
TIME	VARCHAR(20)
DATETIME	<p>Wenn die Skala => 0 und =< 6 ist, je nach Redshift-Zielspalten typ einer der folgenden:</p> <p>TIMESTAMP (s)</p> <p>TIMESTAMPTZ (s) – Wenn der Quell-Zeitstempel einen Zonen-Offset enthält (wie in SQL Server oder Oracle), wird er beim Einfügen/Aktualisieren in UTC konvertiert. Wenn er keinen Offset enthält, wird die Zeit bereits in UTC berücksichtigt.</p> <p>Ist die Skalierung => 7 und =< 9, dann:</p>

AWS DMS-Datentypen	Amazon-Redshift-Datentypen
	VARCHAR (37)
INT1	INT2
INT2	INT2
INT4	INT4
INT8	INT8
NUMERIC	Ist die Skalierung => 0 und =< 37, dann: NUMERIC (p,s) Ist die Skalierung => 38 und =< 127, dann: VARCHAR (Länge)
REAL4	FLOAT4
REAL8	FLOAT8
STRING	Bei einer Länge von 1 bis 65 535 Verwendung von VARCHAR (Länge in Byte) Bei einer Länge von 65 536 bis 2 147 483 647 Verwendung von VARCHAR (65535)
UINT1	INT2
UINT2	INT2
UINT4	INT4
UINT8	NUMERIC (20,0)

AWS DMS-Datentypen	Amazon-Redshift-Datentypen
WSTRING	Bei einer Länge von 1 bis 65 535 Verwendung von NVARCHAR (Länge in Byte) Bei einer Länge von 65 536 bis 2 147 483 647 Verwendung von NVARCHAR (65535)
BLOB	VARCHAR (maximale LOB-Größe * 2) Die maximale LOB-Größe darf 31 KB nicht überschreiten. Amazon Redshift unterstützt keine VARCHARs über 64 KB.
NCLOB	NVARCHAR (maximale LOB-Größe) Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine VARCHARs über 64 KB.
CLOB	VARCHAR (maximale LOB-Größe) Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine VARCHARs über 64 KB.

Verwenden von AWS DMS mit Amazon Redshift Serverless als Ziel

AWS DMS unterstützt Amazon Redshift Serverless als Zielendpunkt. Informationen zur Verwendung von Amazon Redshift Serverless finden Sie unter [Amazon Redshift Serverless](#) im [Amazon-Redshift-Verwaltungshandbuch](#).

In diesem Thema wird die Verwendung eines Amazon-Redshift-Serverless-Endpunkts mit AWS DMS beschrieben.

Note

Wenn Sie einen Amazon-Redshift-Serverless-Endpunkt erstellen, verwenden Sie für das Feld DatabaseName Ihrer [RedshiftSettings](#)-Endpunktconfiguration entweder den Namen

des Amazon Redshift Data Warehouse oder den Namen des Arbeitsgruppenendpunkts. Verwenden Sie für das Feld Servername den Wert für den Endpunkt, der auf der Seite Arbeitsgruppe für den Serverless-Cluster angezeigt wird (z. B. default-workgroup.093291321484.us-east-1.redshift-serverless.amazonaws.com). Weitere Informationen zum Erstellen eines Endpunkts finden Sie unter [Erstellen der Quell- und Zielendpunkte](#). Informationen zum Arbeitsgruppenendpunkt finden Sie unter [Verbinden mit Amazon Redshift Serverless](#).

Vertrauensrichtlinie mit Amazon Redshift Serverless als Ziel

Wenn Sie Amazon Redshift Serverless als Zielendpunkt verwenden, müssen Sie Ihrer Vertrauensrichtlinie den folgenden hervorgehobenen Abschnitt hinzufügen. Diese Vertrauensrichtlinie wird der Rolle dms-access-for-endpoint angefügt.

```
{
  "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
}
}
```

Weitere Informationen zur Verwendung einer Vertrauensrichtlinie mit AWS DMS finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Einschränkungen bei Verwendung von Amazon Redshift Serverless als Ziel

Bei Verwendung von Redshift Serverless als Ziel gelten folgende Einschränkungen:

- AWS DMS unterstützt Amazon Redshift Serverless nur als Endpunkt in Regionen, die Amazon Redshift Serverless unterstützen. Informationen über die Regionen, die Amazon Redshift Serverless unterstützen, finden Sie unter Redshift Serverless API im Thema [Endpunkte und Kontingente von Amazon Redshift](#) in der [Allgemeinen AWS-Referenz](#).
- Wenn Sie Enhanced VPC Routing verwenden, stellen Sie sicher, dass Sie einen Amazon-S3-Endpunkt in derselben VPC erstellen, in der sich auch Ihr Redshift-Serverless-Cluster oder Ihr von Redshift bereitgestellter Cluster befindet. Weitere Informationen finden Sie unter [Verwenden von Enhanced VPC Routing mit Amazon Redshift als Ziel für AWS Database Migration Service](#).
- AWS DMS Serverless unterstützt Amazon Redshift Serverless nicht als Ziel.

Verwenden einer SAP ASE-Datenbank als Ziel für AWS Database Migration Service

Sie können Daten mithilfe von AWS DMS aus einer der unterstützten Datenbankquellen zu SAP-ASE-Datenbanken (Adaptive Server Enterprise) – ehemals als Sybase-Datenbanken bezeichnet – migrieren.

Informationen zu den Versionen von SAP ASE, die AWS DMS als Ziel unterstützt, finden Sie unter [Ziele für AWS DMS](#).

Voraussetzungen für die Verwendung einer SAP ASE-Datenbank als Ziel für AWS Database Migration Service

Bevor Sie eine SAP ASE-Datenbank als Ziel für AWS DMS verwenden, vergewissern Sie sich, dass Sie die folgenden Voraussetzungen erfüllen:

- Erteilen Sie dem AWS DMS-Benutzer Zugriff auf das SAP ASE-Konto. Dieser Benutzer muss über Lese-/Schreibberechtigung für die SAP ASE-Datenbank verfügen.
- In einigen Fällen kann es vorkommen, dass Sie auf eine SAP-ASE-Version 15.7 replizieren, die auf einer Amazon-EC2-Instance unter Microsoft Windows installiert ist, die mit nicht-lateinischen Zeichen (z. B. Chinesisch) konfiguriert ist. In solchen Fällen erfordert AWS DMS die Installation von SAP ASE 15.7 SP121 auf dem SAP ASE-Zielcomputer erforderlich.

Einschränkungen bei der Verwendung einer SAP-ASE-Datenbank als Ziel für AWS DMS

Die folgenden Einschränkungen gelten, wenn Sie eine SAP-ASE-Datenbank als Ziel für AWS DMS verwenden:

- AWS DMS unterstützt keine Tabellen, die Felder mit den folgenden Datentypen enthalten. Replizierte Spalten mit diesen Datentypen werden als Null angezeigt.
 - Benutzerdefinierter Typ (UDT)

Endpunkteinstellungen bei Verwendung von SAP ASE als Ziel für AWS DMS

Sie können Endpunkteinstellungen zum Konfigurieren Ihrer SAP-ASE-Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen beim Erstellen des Zielendpunkts über die AWS DMS-Konsole oder über den Befehl `create-endpoint` in der [AWS CLI](#) mit der JSON-Syntax `--sybase-settings '{"EndpointSetting": "value", ...}'` an.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit SAP ASE als Ziel verwenden können.

Name	Beschreibung
Driver	<p>Legen Sie dieses Attribut fest, wenn Sie TLS für ASE-Versionen 15.7 und höher verwenden möchten.</p> <p>Standardwert: Adaptive Server Enterprise</p> <p>Beispiel: driver=Adaptive Server Enterprise 16.03.06;</p> <p>Zulässige Werte: Adaptive Server Enterprise 16.03.06</p>
AdditionalConnectionProperties	Alle zusätzlichen ODBC-Verbindungsparameter, die Sie angeben möchten.

Zieldatentypen für SAP ASE

In der folgenden Tabelle sind die SAP ASE-Datenbank-Zieldatentypen aufgeführt, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung von AWS DMS-Datentypen.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

AWS DMS-Datentypen	SAP ASE-Datentypen
BOOLEAN	BIT
BYTES	VARBINARY (Länge)
DATUM	DATUM
TIME	TIME
TIMESTAMP	Ist die Skalierung => 0 und =< 6, dann: BIGDATETIME

AWS DMS-Datentypen	SAP ASE-Datentypen
	Ist die Skalierung => 7 und =< 9, dann: VARCHAR (37)
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	DOUBLE PRECISION
STRING	VARCHAR (Länge)
UINT1	TINYINT
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	VARCHAR (Länge)
BLOB	IMAGE
CLOB	UNITEXT
NCLOB	TEXT

Verwenden von Amazon S3 als Ziel für AWS Database Migration Service

Sie können Daten mithilfe von AWS DMS aus einer der unterstützten Datenbankquellen zu Amazon S3 migrieren. Bei Verwendung von Amazon S3 als Ziel in einer AWS DMS-Aufgabe werden

Daten des vollständigen Ladevorgangs sowie CDC-Daten (Change Data Capture, Erfassung von Änderungsdaten) standardmäßig in CSV-Dateien (Comma-Separated Value, durch Kommas getrennte Werte) geschrieben. Für kompaktere Speicher- und schnellere Abfrageoptionen haben Sie zudem die Möglichkeit, die Daten im Apache Parquet-Format (.parquet) zu schreiben.

AWS DMS benennt Dateien, die während eines vollständigen Ladevorgangs erstellt wurden, mithilfe eines inkrementellen Hexadezimalzählers – beispielsweise LOAD00001.csv, LOAD00002..., LOAD00009, LOAD0000A usw bei CSV-Dateien. AWS DMS benennt CDC-Dateien mithilfe von Zeitstempeln, beispielsweise 20141029-1134010000.csv. Für jede Quelltable mit Datensätzen erstellt AWS DMS einen Ordner unter dem angegebenen Zielordner (wenn die Quelltable nicht leer ist). AWS DMS schreibt alle Dateien des vollständigen Ladevorgangs und alle CDC-Dateien in den angegebenen Amazon-S3-Bucket. Sie können die Größe der von AWS DMS erstellten Dateien mithilfe der Endpunkteinstellung [MaxFileSize](#) steuern.

Der Parameter `bucketFolder` enthält den Speicherort, an dem die CSV- oder Parquet-Dateien vor dem Hochladen in den S3-Bucket gespeichert werden. Bei CSV-Dateien werden die Tabellendaten im folgenden Format im S3-Bucket gespeichert, dargestellt für Dateien des vollständigen Ladevorgangs.

```
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
...
```

Sie können mithilfe der zusätzlichen Verbindungsattribute die Trennzeichen für die Spalte und die Zeile sowie andere Parameter festlegen. Weitere Informationen zu den zusätzlichen Verbindungsattributen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#) am Ende dieses Abschnitts.

Sie können einen Bucket-Eigentümer angeben und Sniping verhindern, indem Sie die Amazon-S3-Endpunkteinstellung `ExpectedBucketOwner` verwenden, wie im Folgenden gezeigt. Wenn Sie dann eine Anforderung stellen, um eine Verbindung zu testen oder eine Migration durchzuführen, gleicht S3 die Konto-ID des Bucket-Eigentümers mit dem angegebenen Parameter ab.

```
--s3-settings='{ "ExpectedBucketOwner": "AWS_Account_ID" }'
```

Wenn Sie AWS DMS verwenden, um Datenänderungen unter Verwendung einer CDC-Aufgabe zu replizieren, gibt die erste Spalte der CSV- oder Parquet-Ausgabedatei an, wie die Zeilendaten geändert wurden. Dies wird in der folgenden CSV-Datei dargestellt.

```
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
```

Für dieses Beispiel nehmen wir an, dass es eine EMPLOYEE-Tabelle in der Quelldatenbank gibt. AWS DMS schreibt Daten in die CSV- oder Parquet-Datei als Reaktion auf die folgenden Ereignisse:

- Ein neuer Mitarbeiter (Bob Smith, Mitarbeiter-ID 101) wird am 04. Juni 2014 in der Niederlassung in New York eingestellt. In der CSV- oder Parquet-Datei gibt das I in der ersten Spalte an, dass eine neue Zeile in die MITARBEITER-Tabelle in der Quelldatenbank INSERTED (eingefügt) wurde.
- Am 08. Oktober 2015 wechselt Bob zur Niederlassung in Los Angeles. In der CSV- oder Parquet-Datei gibt das U an, dass die entsprechende Zeile in der MITARBEITER-Tabelle UPDATED (aktualisiert) wurde, um Bobs neuen Firmenstandort widerzuspiegeln. Der Rest der Linie spiegelt die Zeile in der MITARBEITER-Tabelle wider, wie sie nach dem UPDATE angezeigt wird.
- Am 13. März 2017 wechselt Bob erneut zur Niederlassung in Dallas. In der CSV- oder Parquet-Datei gibt das U an, dass diese Zeile erneut UPDATED (aktualisiert) wurde. Der Rest der Linie spiegelt die Zeile in der MITARBEITER-Tabelle wider, wie sie nach dem UPDATE angezeigt wird.
- Nach einiger Zeit der Arbeit in Dallas verlässt Bob die Firma. In der CSV- oder Parquet-Datei gibt das D an, dass die Zeile in der Quelltable DELETED (gelöscht) wurde. Der Rest der Linie spiegelt wider, wie die Zeile in der MITARBEITER-Tabelle vor der Löschung angezeigt wurde.

Beachten Sie, dass AWS DMS für CDC standardmäßig die Zeilenänderungen für jede Datenbanktabelle unabhängig von der Transaktionsreihenfolge speichert. Wenn Sie die Zeilenänderungen in CDC-Dateien der Transaktionsreihenfolge entsprechend speichern möchten, müssen Sie dies und den Ordnerpfad, in dem die CDC-Transaktionsdateien auf dem S3-Ziel gespeichert werden sollen, über S3-Endpunkteinstellungen angeben. Weitere Informationen finden Sie unter [Erfassen von Datenänderungen \(CDC\), einschließlich Transaktionsreihenfolge auf dem S3-Ziel](#).

Um die Häufigkeit von Schreibvorgängen auf ein Amazon-S3-Ziel während einer Datenreplikationsaufgabe zu steuern, können Sie die zusätzlichen Verbindungsattribute `cdcMaxBatchInterval` und `cdcMinFileSize` konfigurieren. Dies kann zu einer besseren Leistung bei der Analyse der Daten ohne zusätzliche überflüssige Aufgaben führen. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#)

Themen

- [Voraussetzungen für die Verwendung von Amazon S3 als Ziel](#)
- [Einschränkungen bei der Verwendung von Amazon S3 als Ziel](#)
- [Sicherheit](#)
- [Speichern von Amazon-S3-Objekten mithilfe von Apache Parquet](#)
- [Markieren von Amazon-S3-Objekten](#)
- [Erstellen von AWS KMS-Schlüsseln zum Verschlüsseln von Amazon-S3-Zielobjekten](#)
- [Verwenden einer datumsbasierten Ordnerpartitionierung](#)
- [Paralleles Laden partitionierter Quellen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#)
- [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#)
- [Verwenden von AWS Glue Data Catalog mit einem Amazon-S3-Ziel für AWS DMS](#)
- [Verwenden von Datenverschlüsselung, Parquet-Dateien und CDC auf Ihrem Amazon-S3-Ziel](#)
- [Angabe von Quelldatenbankoperationen in migrierten S3-Daten](#)
- [Zieldatentypen für S3 Parquet](#)

Voraussetzungen für die Verwendung von Amazon S3 als Ziel

Überprüfen Sie, ob folgende Bedingungen erfüllt sind, bevor Sie Amazon S3 als Ziel verwenden:

- Der S3-Bucket, den Sie als Ziel verwenden, befindet sich in derselben AWS-Region wie die DMS-Replikations-Instance, die Sie für die Migration der Daten verwenden.
- Das AWS-Konto, das Sie für die Migration verwenden, verfügt über eine IAM-Rolle mit Schreib- und Löschzugriff auf den S3-Bucket, den Sie als Ziel verwenden.
- Diese Rolle besitzt Tagging-Zugriff, sodass Sie alle S3-Objekte, die in den Ziel-Bucket geschrieben werden, markieren können.
- Der IAM-Rolle wurde DMS (`dms.amazonaws.com`) als vertrauenswürdige Entität hinzugefügt.

Um diesen Kontozugriff einzurichten, vergewissern Sie sich, dass die dem Benutzerkonto zugewiesene Rolle, mit der die Migrationsaufgabe erstellt wird, den folgenden Satz von Berechtigungen aufweist.

```
{
  "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"
      ]
    }
  ]
}
```

Die Voraussetzungen für die Verwendung der Validierung mit S3 als Ziel finden Sie unter [Voraussetzungen für die S3-Zielvalidierung](#).

Einschränkungen bei der Verwendung von Amazon S3 als Ziel

Bei der Verwendung von Amazon S3 als Ziel gelten die folgenden Einschränkungen:

- Aktivieren Sie die Versionsverwaltung für S3 nicht. Wenn Sie die Versionsverwaltung für S3 benötigen, verwenden Sie Lebenszyklusrichtlinien, um alte Versionen aktiv zu löschen. Andernfalls kann es aufgrund eines Timeouts eines `list-object`-Aufrufs in S3 zu Verbindungsfehlern beim Endpunkttest kommen. Informationen zum Erstellen einer Lebenszyklusrichtlinie für einen S3-Bucket finden Sie unter [Verwalten Ihres Speicher-Lebenszyklus](#). Informationen zum

Löschen einer Version eines S3-Objekts finden Sie unter [Löschen von Objekten aus einem versionsverwaltungsfähigen Bucket](#).

- Ein VPC-fähiger S3-Bucket (Gateway-VPC) wird in den Versionen 3.4.7 und höher unterstützt.
- Die folgenden DDL-Befehle (Data Definition Language) werden für Change Data Capture (CDC) unterstützt: Truncate Table (Tabelle kürzen), Drop Table (Tabelle löschen), Create Table (Tabelle erstellen), Rename Table (Tabelle umbenennen), Add Column (Spalte hinzufügen), Drop Column (Spalte löschen), Rename Column (Spalte umbenennen) und Change Column Data Type (Spaltentyp ändern). Beachten Sie, dass beim Hinzufügen, Löschen oder Umbenennen einer Spalte in der Quelldatenbank keine ALTER-Anweisung im Ziel-S3-Bucket aufgezeichnet wird und AWS DMS zuvor erstellte Datensätze nicht der neuen Struktur entsprechend ändert. Nach der Änderung erstellt AWS DMS alle neuen Datensätze unter Verwendung der neuen Tabellenstruktur.

Note

Bei einer DDL-Operation zum Kürzen werden alle Dateien und die entsprechenden Tabellenordner aus einem S3-Bucket entfernt. Mithilfe von Aufgabeneinstellungen können Sie dieses Verhalten deaktivieren und konfigurieren, wie DMS mit dem DDL-Verhalten bei Change Data Capture (CDC) umgehen soll. Weitere Informationen finden Sie unter [Aufgabeneinstellungen für den Umgang mit der DDL-Änderungsverarbeitung](#).

- Der vollständige LOB-Modus wird nicht unterstützt.
- Änderungen an der Quelltabellenstruktur während des vollständigen Ladevorgangs werden nicht unterstützt. Änderungen an Daten werden während des vollständigen Ladevorgangs unterstützt.
- Mehrere Aufgaben, die Daten aus derselben Quelltable in denselben S3-Ziel-Endpunkt-Bucket replizieren, führen dazu, dass diese Aufgaben in dieselbe Datei schreiben. Wir empfehlen, unterschiedliche Zielendpunkte (Buckets) anzugeben, wenn die Datenquelle aus derselben Tabelle stammt.
- BatchApply wird für einen S3-Endpunkt nicht unterstützt. Bei Verwendung von Batch Apply (z. B. der Zielmetadaten-Aufgabeneinstellung BatchApplyEnabled) für ein S3-Ziel kann es zu einem Datenverlust kommen.
- Sie können DatePartitionEnabled oder addColumnName nicht zusammen mit PreserveTransactions oder CdcPath verwenden.
- AWS DMS unterstützt das Umbenennen mehrerer Quelltabellen in denselben Zielordner mithilfe von Transformationsregeln nicht.

- Wenn während der Vollladephase intensiv in die Quelltable geschrieben wird, schreibt DMS möglicherweise doppelte Datensätze in den S3-Bucket oder zwischengespeicherte Änderungen.
- Wenn Sie die Aufgabe mit einem `TargetTablePrepMode` von `D0_NOTHING` konfigurieren, schreibt DMS möglicherweise doppelte Datensätze in den S3-Bucket, wenn die Aufgabe während der Vollladephase unterbrochen und abrupt wieder aufgenommen wird.
- Wenn Sie den Zielendpunkt mit der `PreserveTransactions`-Einstellung `true` konfigurieren, werden zuvor generierte CDC-Dateien beim erneuten Laden einer Tabelle nicht gelöscht. Weitere Informationen finden Sie unter [Erfassen von Datenänderungen \(CDC\), einschließlich Transaktionsreihenfolge auf dem S3-Ziel](#).

Die Einschränkungen für die Verwendung der Validierung mit S3 als Ziel finden Sie unter [Einschränkungen bei Verwendung der S3-Zielvalidierung](#).

Sicherheit

Um Amazon S3 als Ziel verwenden zu können, muss das für die Migration eingesetzte Konto über Schreib- und Löschzugriff auf den Amazon-S3-Bucket verfügen, den Sie als Ziel verwenden. Geben Sie den Amazon-Ressourcennamen (ARN) einer IAM-Rolle an, die über die erforderlichen Berechtigungen für den Zugriff auf Amazon S3 verfügt.

AWS DMS unterstützt eine Reihe vordefinierter Berechtigungen für Amazon S3, auch als vordefinierte Zugriffssteuerungslisten (ACLs) bezeichnet. Jede vordefinierte ACL umfasst eine Reihe von Empfängern und Berechtigungen, die Sie verwenden können, um Berechtigungen für den Amazon-S3-Bucket festzulegen. Sie können eine vordefinierte ACL mit dem `cannedAclForObjects` in dem Attribut der Verbindungszeichenfolge für Ihren S3-Zielendpunkt festlegen. Weitere Informationen zur Verwendung des zusätzlichen Oracle-Verbindungsattributs `cannedAclForObjects` finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#). Weitere Informationen zu vordefinierten ACLs von Amazon S3 finden Sie unter [Vordefinierte ACL](#).

Die IAM-Rolle, die Sie für die Migration verwenden, muss in der Lage sein, die API-Operation `s3:PutObjectAcl` durchzuführen.

Speichern von Amazon-S3-Objekten mithilfe von Apache Parquet

Das CSV-Format ist das Standardspeicherformat für Amazon-S3-Zielobjekte. Das Apache Parquet-Speicherformat (`.parquet`) bietet indessen kompaktere Speicheroptionen und schnellere Abfragen.

Apache Parquet ist ein Open-Source-Dateispeicherformat, das ursprünglich für Hadoop entwickelt wurde. Weitere Informationen über Apache Parquet finden Sie unter <https://parquet.apache.org/>.

Um .parquet als das Speicherformat für Ihre migrierten S3-Zielobjekte festzulegen, können Sie die folgenden Mechanismen verwenden:

- Endpunkteinstellungen, die Sie bei der Erstellung des Endpunkts mithilfe der AWS CLI oder der API für AWS DMS als Parameter für ein JSON-Objekt angeben. Weitere Informationen finden Sie unter [Verwenden von Datenverschlüsselung, Parquet-Dateien und CDC auf Ihrem Amazon-S3-Ziel](#).
- Zusätzliche Verbindungsattribute, die Sie bei der Erstellung des Endpunkts als eine durch Semikolons getrennte Liste bereitstellen. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#).

Markieren von Amazon-S3-Objekten



Sie können von einer Replikations-Instance erstellte Amazon-S3-Objekte markieren, indem Sie entsprechende JSON-Objekte als Teil der Aufgaben-Tabellen-Zuweisungsregeln angeben. Weitere Informationen über die Anforderungen und Optionen für das Markieren von S3-Objekten, einschließlich gültiger Tag-Namen, finden Sie unter [Markieren von Objekten](#) im Benutzerhandbuch für Amazon Simple Storage Service. Weitere Informationen über Tabellenzuweisungen mithilfe von JSON finden Sie unter [Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON](#).

Sie markieren S3-Objekte, die für bestimmte Tabellen und Schemata erstellt wurden, mithilfe eines oder mehrerer JSON-Objekte des `selection`-Regeltyps. Nach Ausführung dieses `selection`-Objekts (oder dieser Objekte) können Sie ein oder mehrere JSON-Objekte des `post-processing`-Regeltyps mithilfe der `add-tag`-Aktion hinzufügen. Diese Nachbearbeitungsregeln identifizieren die zu markierenden S3-Objekte und geben die Namen und Werte der Tags an, die Sie diesen S3-Objekten hinzufügen möchten.

Sie finden die Parameter zur Angabe in den JSON-Objekten des `post-processing`-Regeltyps in der folgenden Tabelle.

Parameter	Mögliche Werte	Beschreibung
<code>rule-type</code>	<code>post-processing</code>	Ein Wert, der die Aktionen zur Nachbereitung auf die generierten Zielobjekte anwendet. Sie können

Parameter	Mögliche Werte	Beschreibung
		eine oder mehrere Nachbereitungsregeln festlegen, um ausgewählte S3-Objekte zu markieren.
<code>rule-id</code>	Ein numerischer Wert.	Ein eindeutiger numerischer Wert zur Identifizierung der Regel.
<code>rule-name</code>	Ein alphanumerischer Wert.	Ein eindeutiger Name zur Identifizierung der Regel.
<code>rule-action</code>	<code>add-tag</code>	Die Nachbereitungsaktion, die Sie auf das S3-Objekt anwenden möchten. Mit einem einzigen JSON-Objekt für die Nachbereitung können Sie der <code>add-tag</code> -Aktion ein oder mehrere Tags hinzufügen.
<code>object-locator</code>	<p><code>schema-name</code> – Der Name des Tabellenschemas.</p> <p><code>table-name</code> – Der Name der Tabelle.</p>	<p>Der Name der einzelnen Schemen und Tabellen, für die die Regel gilt. Sie können das Prozentzeichen "%" als Platzhalter für alle oder einen Teil der Werte der einzelnen <code>object-locator</code>-Parameter verwenden. Auf diese Weise können Sie diese Elemente aufeinander abstimmen:</p> <ul style="list-style-type: none"> • Eine einzelne Tabelle in einem einzelnen Schema • Eine einzelne Tabelle in einigen oder allen Schemata • Einige oder alle Tabellen in einem einzelnen Schema • Einige oder alle Tabellen in einigen oder allen Schemata

Parameter	Mögliche Werte	Beschreibung
tag-set	<p>key – Ein beliebiger gültiger Name für ein einzelnes Tag.</p> <p>value – Ein beliebiger gültiger JSON-Wert für dieses Tag.</p>	<p>Die Namen und Werte für ein oder mehrere Tags, die für die einzelnen erstellten S3-Objekte festgelegt werden sollen, welche dem angegebenen <code>object-locator</code> entsprechen. Sie können bis zu 10 Schlüssel-Wert-Paare in einem einzigen <code>tag-set</code>-Parameterobjekt angeben. Weitere Informationen zum Markieren von S3-Objekten finden Sie unter Markieren von Objekten im Benutzerhandbuch für Amazon Simple Storage Service.</p> <p>Sie können auch einen dynamischen Wert für alle oder einen Teil des Werts für die <code>value</code>- und <code>key</code>-Parameter eines Tags mithilfe von <code>\${dyn-value }</code> angeben. Dabei kann <code>\${dyn-value }</code> entweder <code>\${schema-name}</code> oder <code>\${table-name}</code> sein. Auf diese Weise können Sie den Namen des aktuell ausgewählten Schemas oder der aktuell ausgewählten Tabelle als den gesamten oder einen Teil des Parameterwerts eingeben.</p> <div data-bbox="974 1533 1510 1869"><p> Note</p><div data-bbox="1055 1648 1477 1869"><p> Important</p><p>Wenn Sie einen dynamischen Wert für den key-</p></div></div>

Parameter	Mögliche Werte	Beschreibung
		<div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p>Parameter eingeben, können Sie Tags mit doppelten Namen für ein S3-Objekt generieren, je nachdem, wie Sie es verwenden möchten. In diesem Fall wird dem Objekt nur eine der doppelten Tag-Einstellungen hinzugefügt.</p> </div>

Wenn Sie mehrere `post-processing`-Regeltypen verwenden, um eine Reihe von S3-Objekten zu markieren, wird jedes S3-Objekt mit nur einem `tag-set`-Objekt aus einer Nachbearbeitungsregel markiert. Zum Markieren eines bestimmten S3-Objekts wird der Tag-Satz verwendet, dessen zugehöriger Objekt-Locator dem S3-Objekt am besten entspricht.

Nehmen Sie beispielsweise an, dass zwei Nachbearbeitungsregeln das gleiche S3-Objekt identifizieren. Darüber hinaus gehen wir davon aus, dass der Objekt-Locator der einen Regel Platzhalter verwendet und der Objekt-Locator der anderen Regel eine genaue Übereinstimmung zur Identifizierung des S3-Objekts einsetzt (ohne Platzhalter). In diesem Fall wird der Tag-Satz zur Markierung des S3-Objekts verwendet, dessen Nachbearbeitungsregel eine genaue Übereinstimmung aufweist. Wenn mehrere Nachbearbeitungsregeln einem bestimmten S3-Objekt gleich gut entsprechen, wird der Tag-Satz zur Markierung des Objekts verwendet, der der ersten dieser Nachbearbeitungsregeln zugewiesen ist.

Example Hinzufügen von statischen Tags zu einem S3-Objekt, das für ein einzelnes Schema und eine einzelne Tabelle erstellt wurde

Die folgenden Auswahl- und Nachbearbeitungsregeln fügen einem erstellten S3-Objekt drei Tags (`tag_1`, `tag_2` und `tag_3` mit den entsprechenden statischen Werten `value_1`, `value_2` und `value_3`) hinzu. Dieses S3-Objekt entspricht einer einzelnen Tabelle in der Quelle mit dem Namen `STOCK` mit einem Schema mit dem Namen `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 Hinzufügen von statischen und dynamischen Tags zu S3-Objekten, die für mehrere Tabellen und Schemata erstellt wurden

Das folgende Beispiel zeigt eine Auswahlregel und zwei Nachbearbeitungsregeln, wobei die Eingabe aus der Quelle alle Tabellen und deren zugehörigen Schemata umfasst.

```
{
  "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"
      }
    ]
  }
]
```

Die erste Nachbearbeitungsregel fügt die beiden Tags (`dw-schema-name` und `dw-schema-table`) sowie die dazugehörigen dynamischen Werte (`${schema-name}` und `my_prefix_${table-name}`) fast allen S3-Objekten hinzu, die in der Zieldatenbank erstellt wurden. Eine Ausnahme bildet das S3-Objekt, das über die zweite Nachbearbeitungsregel identifiziert und markiert wird. Daher wird jedes S3-Zielobjekt, das durch den Platzhalter-Objekt-Locator identifiziert wurde, mit Tags erstellt, die das Schema und die Tabelle erkennen lassen, dem es in der Quelle zugewiesen ist.

Die zweite Nachbearbeitungsregel fügt `tag_1` und `tag_2` mit den entsprechenden statischen Werten `value_1` und `value_2` einem erstellten S3-Objekt hinzu, welches von einem Objekt-Locator als eine genaue Übereinstimmung identifiziert wurde. Dieses erstellte S3-Objekt entspricht daher der einzigen Tabelle in der Quelle mit dem Namen `ITEM` mit einem Schema mit dem Namen `aat`. Aufgrund der genauen Entsprechung ersetzen diese Tags alle Tags für dieses Objekt, die von der ersten Nachbearbeitungsregel hinzugefügt wurden, da diese den S3-Objekten nur aufgrund der Platzhalter entsprechen.

Example Hinzufügen von dynamischen Tag-Namen und -Werten zu S3-Objekten

Das folgende Beispiel umfasst zwei Auswahlregeln und eine Nachbearbeitungsregel. Die Eingabe aus der Quelle schließt nur die `ITEM`-Tabelle entweder im `retail`- oder `wholesale`-Schema ein.

```
{
  "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"
      }
    ]
  }
}
```



```

    ]
  ]
}
```

Der Tag-Satz für die Nachbereitungsregel fügt allen S3-Objekten, die für die ITEM-Tabelle in der Zieldatenbank erstellt wurden, zwei Tags (`dw-schema-name` und `dw-schema-table`) hinzu. Das erste Tag hat den dynamischen Wert `"${schema-name}"` und das zweite Tag den statischen Wert `"my_prefix_ITEM"`. Daher wird jedes S3-Zielobjekt mit Tags erstellt, die das Schema und die Tabelle erkennen lassen, dem es in der Quelle zugewiesen ist.

Darüber hinaus fügt das Tag-Set zwei zusätzliche Tags mit dynamischen Namen hinzu (`${schema-name}_ITEM_tag_1` und `"${schema-name}_ITEM_tag_2"`). Diese haben die entsprechenden statischen Werte `value_1` und `value_2`. Aus diesem Grund werden diese Tags nach dem aktuellen Schema benannt, `retail` oder `wholesale`. In diesem Objekt ist es nicht möglich, einen doppelten dynamischen Tag-Namen zu verleihen, da jedes Objekt für einen einzelnen eindeutigen Schemanamen erstellt wurde. Der Schemaname wird dazu verwendet, einen anderweitig eindeutigen Tag-Namen zu erstellen.

Erstellen von AWS KMS-Schlüsseln zum Verschlüsseln von Amazon-S3-Zielobjekten

Sie können benutzerdefinierte AWS KMS-Schlüssel erstellen und zum Verschlüsseln Ihrer Amazon-S3-Zielobjekte verwenden. Nachdem Sie einen KMS-Schlüssel erstellt haben, können Sie ihn verwenden, um Objekte mithilfe eines der folgenden Ansätze zu verschlüsseln, wenn Sie den S3-Zielendpunkt erstellen:

- Verwenden Sie die folgenden Optionen für die S3-Zielobjekte (im standardmäßigen CSV-Dateispeicherformat), wenn Sie den `create-endpoint`-Befehl mithilfe der AWS CLI ausführen.

```
--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"}'
```

Hier handelt es sich bei `your-KMS-key-ARN` um den Amazon-Ressourcennamen (ARN) für Ihren KMS-Schlüssel. Weitere Informationen finden Sie unter [Verwenden von Datenverschlüsselung, Parquet-Dateien und CDC auf Ihrem Amazon-S3-Ziel](#).

- Richten Sie für das zusätzliche Verbindungsattribut `encryptionMode` den Wert `SSE_KMS` und für das zusätzliche Verbindungsattribut `serverSideEncryptionKmsKeyId` den ARN für Ihren KMS-

Schlüssel ein. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#).

Um Amazon-S3-Zielobjekte mithilfe eines KMS-Schlüssels zu verschlüsseln, benötigen Sie eine IAM-Rolle mit Berechtigungen für den Zugriff auf den Amazon-S3-Bucket. Auf diese IAM-Rolle wird dann in einer Richtlinie (einer Schlüsselrichtlinie) zugegriffen, die dem von Ihnen erstellten Verschlüsselungsschlüssel angefügt ist. Sie können diesen Vorgang in Ihrer IAM-Konsole ausführen, indem Sie Folgendes erstellen:

- Eine Richtlinie mit Berechtigungen für den Zugriff auf den Amazon-S3-Bucket
- Eine IAM-Rolle mit dieser Richtlinie
- Einen KMS-Verschlüsselungsschlüssel mit einer Schlüsselrichtlinie, die sich auf diese Rolle bezieht

Die entsprechende Vorgehensweise wird nachfolgend beschrieben.

So erstellen Sie eine IAM-Richtlinie mit Berechtigungen für den Zugriff auf den Amazon-S3-Bucket

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich die Option Policies (Richtlinien) aus. Die Seite Policies (Richtlinien) wird geöffnet.
3. Wählen Sie Richtlinie erstellen aus. Die Seite Create policy (Richtlinie erstellen) wird geöffnet.
4. Wählen Sie Service und anschließend die Option S3 aus. Es wird eine Liste mit den Aktionsberechtigungen angezeigt.
5. Wählen Sie Expand all (Alle maximieren) aus, um die Liste zu erweitern, und aktivieren Sie zumindest die folgenden Berechtigungen:
 - ListBucket
 - PutObject
 - DeleteObject

Wählen Sie alle anderen Berechtigungen aus, die Sie benötigen, und klicken Sie anschließend auf Collapse all (Alle minimieren), um die Liste zu reduzieren.

6. Wählen Sie Resources (Ressourcen) aus, um die Ressourcen festzulegen, auf die Sie Zugriff erlangen möchten. Wählen Sie zumindest Alle Ressourcen aus, um allgemeinen Zugriff auf die Amazon-S3-Ressourcen zu gewähren.

7. Fügen Sie alle anderen Bedingungen oder Berechtigungen hinzu, die Sie benötigen, und wählen Sie dann Review policy (Richtlinie überprüfen) aus. Überprüfen Sie die Ergebnisse auf der Seite Review policy (Richtlinie prüfen).
8. Wenn die Einstellungen Ihren Vorstellungen entsprechen, geben Sie einen Namen für die Richtlinie (z. B. DMS-S3-endpoint-access) sowie eine Beschreibung ein und wählen Sie anschließend Create policy (Richtlinie erstellen) aus. Die Seite Policies (Richtlinien) wird geöffnet und zeigt eine Mitteilung an, die darauf hinweist, dass Ihre Richtlinie erstellt wurde.
9. Wählen Sie den Richtliniennamen aus der Liste mit den Policies (Richtlinien) aus. Die Seite Summary (Übersicht) wird geöffnet und zeigt den JSON-Code für die Richtlinie an, der dem folgenden Beispiel ähnelt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Sie haben jetzt die neue Richtlinie erstellt, über die Sie auf Amazon-S3-Ressourcen zugreifen können, um diese mit einem angegebenen Namen, z. B. DMS-S3-endpoint-access, zu verschlüsseln.

So erstellen Sie eine IAM-Rolle mit dieser Richtlinie

1. Wählen Sie im Navigationsbereich Ihrer IAM-Konsole Rollen aus. Die Detailseite Roles (Rollen) wird geöffnet.
2. Wählen Sie Rolle erstellen aus. Die Seite Create role (Rolle erstellen) wird geöffnet.
3. Wenn AWS-Service als vertrauenswürdige Entität ausgewählt ist, wählen Sie DMS als den Service aus, der die IAM-Rolle verwendet.

4. Wählen Sie Weiter: Berechtigungen aus. Die Ansicht Attach permissions policies (Berechtigungsrichtlinien hinzufügen) wird auf der Seite Create role (Rolle erstellen) angezeigt.
5. Suchen Sie die IAM-Richtlinie für die IAM-Rolle, die Sie im vorherigen Verfahren (DMS-S3-endpoint-access) erstellt haben, und wählen Sie sie aus.
6. Wählen Sie Weiter: Markierungen. Die Ansicht Add tags (Tags hinzufügen) wird auf der Seite Create role (Rolle erstellen) angezeigt. Hier können Sie alle gewünschten Tags hinzufügen.
7. Wählen Sie Weiter: Prüfen aus. Die Ansicht Review (Überprüfen) wird auf der Seite Create role (Rolle erstellen) angezeigt. Hier können Sie die Ergebnisse überprüfen.
8. Wenn die Einstellungen Ihren Vorstellungen entsprechen, geben Sie einen Namen für die Rolle (Pflichtangabe, z. B. DMS-S3-endpoint-access-role) sowie eine zusätzliche Beschreibung ein, und wählen Sie anschließend Create role (Rolle erstellen) aus. Die Detailseite Roles (Rollen) wird geöffnet und zeigt eine Mitteilung an, die darauf hinweist, dass Ihre Rolle erstellt wurde.


Sie haben jetzt die neue Rolle erstellt, über die Sie auf Amazon-S3-Ressourcen zugreifen können, um diese mit einem angegebenen Namen, z. B. DMS-S3-endpoint-access-role, zu verschlüsseln.

So erstellen Sie einen KMS-Verschlüsselungsschlüssel mit einer Schlüsselrichtlinie, die sich auf Ihre IAM-Rolle bezieht

Note

Weitere Hinweise zur Funktionsweise von AWS DMS mit AWS KMS-Verschlüsselungsschlüsseln finden Sie unter [Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen](#).

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter <https://console.aws.amazon.com/kms>.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key. Die Seite Configure key (Schlüssel konfigurieren) wird geöffnet.
5. Wählen Sie für Key type (Schlüsseltyp) Symmetric (Symmetrisch).

 Note

Wenn Sie diesen Schlüssel erstellen, können Sie nur einen symmetrischen Schlüssel erstellen, da alle AWS-Services, wie z. B. Amazon S3, nur mit symmetrischen Verschlüsselungsschlüsseln arbeiten.

6. Wählen Sie Erweiterte Optionen aus. Stellen Sie für Key material origin (Schlüsselmaterialherkunft) sicher, dass KMS ausgewählt ist, und wählen Sie dann Next (Weiter). Die Seite Add Labels (Etiketten hinzufügen) wird geöffnet.
7. Geben Sie unter Create alias and description (Alias und Beschreibung erstellen) einen Alias für den Schlüssel (z. B. `DMS-S3-endpoint-encryption-key`) und eventuell eine weitere Beschreibung ein.
8. Fügen Sie unter Tags alle Tags hinzu, die Sie verwenden möchten, um den Schlüssel zu identifizieren und seine Nutzung zu verfolgen, und wählen Sie anschließend Next Step (Nächster Schritt) aus. Die Seite Define key administrative permissions (Schlüsselverwaltungsberechtigungen definieren) wird geöffnet und zeigt eine Liste der Benutzer und Rollen an, aus denen Sie auswählen können.
9. Fügen Sie die Benutzer und Rollen hinzu, die den Schlüssel verwalten sollen. Stellen Sie sicher, dass diese Benutzer und Rollen über die erforderlichen Berechtigungen verfügen, um den Schlüssel zu verwalten.
10. Wählen Sie unter Key deletion (Schlüssellöschung) aus, ob die Schlüsseladministratoren den Schlüssel löschen können, und klicken Sie anschließend auf Next Step (Nächster Schritt). Die Seite Define Key usage permissions (Schlüsselnutzungsberechtigungen definieren) wird geöffnet und zeigt eine weitere Liste mit Benutzern und Rollen an, aus denen Sie auswählen können.
11. Wählen Sie für Dieses Konto die verfügbaren Benutzer aus, die kryptografische Operationen auf Amazon-S3-Zielen ausführen sollen. Wählen Sie zudem die Rolle aus, die Sie zuvor unter Rollen erstellt haben, um den Zugriff zur Verschlüsselung von Amazon-S3-Zielobjekten zu aktivieren, wie z. B. `DMS-S3-endpoint-access-role`.
12. Wenn Sie weitere Konten hinzufügen möchten, die nicht für denselben Zugriff aufgeführt sind, wählen Sie unter Other AWS accounts (Andere Konten) die Option Add another AWS account (Anderes Konto hinzufügen) und dann Next (Weiter). Die Seite Review and edit key policy (Schlüsselrichtlinie prüfen und bearbeiten) wird geöffnet. Hier wird das JSON für die Schlüsselrichtlinie angezeigt, die Sie überprüfen und bearbeiten können, indem Sie in den vorhandenen JSON-Inhalt schreiben. Hier können Sie sehen, wo die Schlüsselrichtlinie auf die Rolle und die Benutzer (z. B. `Admin` und `User1`) verweist, die Sie im vorherigen Schritt

ausgewählt haben. Sie können auch die verschiedenen Schlüsselaktionen sehen, die für die verschiedenen Prinzipale (Benutzer und Rollen) zulässig sind, wie im folgenden Beispiel gezeigt.

```
{
  "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. Wählen Sie **Finish** (Abschließen). Die Seite Verschlüsselungsschlüssel wird geöffnet und zeigt eine Mitteilung an, die darauf hinweist, dass Ihr KMS-Schlüssel erstellt wurde.

Damit haben Sie einen neuen KMS-Schlüssel mit einem angegebenen Alias (z. B. `DMS-S3-endpoint-encryption-key`) erstellt. Dieser Schlüssel ermöglicht es AWS DMS, Amazon-S3-Zielobjekte zu verschlüsseln.

Verwenden einer datumsbasierten Ordnerpartitionierung

AWS DMS unterstützt S3-Ordnerpartitionen, die auf einem Transaktions-Commit-Datum basieren, wenn Sie Amazon S3 als Zielendpunkt verwenden. Bei Verwendung der datumsbasierten Ordnerpartitionierung können Sie Daten aus einer einzelnen Quelltable in eine Zeithierarchie-Ordnerstruktur in einem S3-Bucket schreiben. Die Partitionierung der Ordner beim Erstellen von S3-Zielendpunkten bietet Ihnen folgende Möglichkeiten:

- Bessere Verwaltung Ihrer S3-Objekte
- Größenbeschränkung der einzelnen S3-Ordner
- Optimierung von Data-Lake-Abfragen oder anderen nachfolgenden Operationen

Sie können die datumsbasierte Ordnerpartitionierung beim Erstellen eines S3-Zielendpunkts aktivieren. Sie können sie aktivieren, wenn Sie vorhandene Daten migrieren und laufende Änderungen replizieren (vollständiges Laden + CDC) oder wenn Sie nur Datenänderungen replizieren (nur CDC). Verwenden Sie die folgenden Einstellungen für den Zielendpunkt:

- `DatePartitionEnabled` – Gibt an, dass eine Partitionierung auf der Grundlage von Datumsangaben erfolgen soll. Setzen Sie diese boolesche Option auf `true`, um S3-Bucket-Ordner basierend auf den Commit-Daten von Transaktionen zu partitionieren.

Sie können diese Einstellung nicht zusammen mit `PreserveTransactions` oder `CdcPath` verwenden.

Der Standardwert ist `false`.

- `DatePartitionSequence` – Identifiziert die Reihenfolge des Datumsformats, das bei der Ordnerpartitionierung verwendet werden soll. Setzen Sie diese ENUM-Option auf `YYYYMMDD`, `YYYYMMDDHH`, `YYYYMM`, `MMYYYYDD` oder `DDMMYYYY`. Der Standardwert ist `YYYYMMDD`. Verwenden Sie diese Einstellung, wenn `DatePartitionEnabled` auf `true` gesetzt ist.

- `DatePartitionDelimiter` – Gibt ein Datumstrennzeichen für die Ordnerpartitionierung an. Setzen Sie diese ENUM-Option auf `SLASH`, `DASH`, `UNDERSCORE` oder `NONE`. Der Standardwert ist `SLASH`. Verwenden Sie diese Einstellung, wenn `DatePartitionEnabled` auf `true` festgelegt ist.

Das folgende Beispiel zeigt, wie die datumsbasierte Ordnerpartitionierung mit Standardwerten für die Reihenfolge der Datenpartitionen und das Trennzeichen aktiviert wird. Dabei wird die Option `--s3-settings '{json-settings}'` des AWS CLI-Befehls `create-endpoint` verwendet.

```
--s3-settings '{"DatePartitionEnabled": true, "DatePartitionSequence":  
"YYYYMMDD", "DatePartitionDelimiter": "SLASH"}
```

Paralleles Laden partitionierter Quellen bei Verwendung von Amazon S3 als Ziel für AWS DMS

Sie können einen parallelen vollständigen Ladevorgang partitionierter Datenquellen in Amazon-S3-Ziele konfigurieren. Durch dieses Vorgehen verbessern sich die Ladezeiten bei der Migration partitionierter Daten von unterstützten Quelldatenbank-Engines zum S3-Ziel. Um die Ladezeiten partitionierter Quelldaten zu verbessern, erstellen Sie S3-Zielunterordner, die den Partitionen jeder Tabelle in der Quelldatenbank zugeordnet sind. Diese partitionsgebundenen Unterordner ermöglichen AWS DMS die Ausführung paralleler Prozesse, um die einzelnen Unterordner im Ziel zu füllen.

Für die Konfiguration eines parallelen vollständigen Ladevorgangs für ein S3-Ziel unterstützt S3 drei `parallel-load`-Regeltypen für die `table-settings`-Regel der Tabellenzuweisung:

- `partitions-auto`
- `partitions-list`
- `ranges`

Weitere Informationen zu diesen Arten von Regeln für paralleles Laden finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Für die Regeltypen `partitions-auto` und `partitions-list` verwendet AWS DMS die einzelnen Partitionsnamen vom Quellendpunkt, um die Ziel-Unterordnerstruktur wie folgt zu identifizieren.

```
bucket_name/bucket_folder/database_schema_name/table_name/partition_name/  
LOADseq_num.csv
```

Hier umfasst der Unterordnerpfad, in den Daten migriert und in dem Daten auf dem S3-Ziel gespeichert werden, einen zusätzlichen Unterordner *partition_name*, der einer gleichnamigen Quellpartition entspricht. In diesem Unterordner *partition_name* werden dann eine oder mehrere Dateien *LOADseq_num.csv* mit Daten gespeichert, die von der angegebenen Quellpartition migriert wurden. *seq_num* ist hier der Sequenznummer-Nachsatz im Namen der CSV-Datei, also beispielsweise *00000001* in der CSV-Datei namens *LOAD00000001.csv*.

Bei einigen Datenbank-Engines, wie MongoDB und DocumentDB, gibt es das Konzept der Partitionen jedoch nicht. Für diese Datenbank-Engines fügt AWS DMS den laufenden Quellsegmentindex dem Namen der CSV-Zieldatei wie folgt als Präfix hinzu.

```
.../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
```

Hier sind die Dateien *SEGMENT1_LOAD00000001.csv* und *SEGMENT1_LOAD00000002.csv* mit demselben laufenden Quellsegmentindex-Präfix *SEGMENT1* benannt. Sie sind so benannt, da die migrierten Quelldaten für diese beiden CSV-Dateien demselben laufenden Quellsegmentindex zugeordnet sind. Andererseits sind die migrierten Daten, die in den Zieldateien *SEGMENT2_LOAD00000009.csv* und *SEGMENT3_LOAD0000000A.csv* gespeichert sind, unterschiedlichen laufenden Quellsegmentindizes zugeordnet. Jedem Dateinamen wird der Name des laufenden Segmentindex, *SEGMENT2* und *SEGMENT3*, als Präfix vorangestellt.

Für den Parallelladetyp *ranges* definieren Sie die Spaltennamen und Spaltenwerte mithilfe der Einstellungen *columns* und *boundaries* der *table-settings*-Regeln. Mit diesen Regeln können Sie wie folgt den Segmentnamen entsprechende Partitionen angeben.

```
"parallel-load": {  
  "type": "ranges",  
  "columns": [  
    "region",  
    "sale"  
  ],  
  "boundaries": [  

```

```

    [
      "NORTH",
      "1000"
    ],
    [
      "WEST",
      "3000"
    ]
  ],
  "segment-names": [
    "custom_segment1",
    "custom_segment2",
    "custom_segment3"
  ]
}

```

Die Einstellung `segment-names` definiert hier Namen für drei Partitionen, um Daten parallel zu dem S3-Ziel zu migrieren. Die migrierten Daten werden parallel geladen und der Reihenfolge nach in CSV-Dateien in den Unterordnern der Partition gespeichert, wie im Folgenden dargestellt.

```

.../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

```

Hier speichert AWS DMS eine Reihe von CSV-Dateien in jedem der drei Partitionsunterordner. Die Reihe von CSV-Dateien in jedem Partitionsunterordner wird inkrementell, beginnend bei `LOAD00000001.csv`, benannt, bis alle Daten migriert sind.

Es kann vorkommen, dass Sie die Partitionsunterordner für einen Parallelladetyp `ranges` nicht explizit über die Einstellung `segment-names` benennen. In diesen Fällen wendet AWS DMS das Standardverfahren an und erstellt jede Reihe von CSV-Dateien in ihrem `table_name`-Unterordner. Hier stellt AWS DMS den Dateinamen jeder Serie von CSV-Dateien den Namen des laufenden Quellsegmentindex voran, wie im Folgenden dargestellt.

```

.../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

```

Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS

Sie können Endpunkteinstellungen zur Konfiguration Ihrer Amazon-S3-Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen beim Erstellen des Zielendpunkts über die AWS DMS-Konsole oder über den Befehl `create-endpoint` in der [AWS CLI](#) mit der JSON-Syntax `--s3-settings '{"EndpointSetting": "value", ...}'` an.


Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Amazon S3 als Ziel verwenden können.


Option	Beschreibung
CsvNullValue	<p>Ein optionaler Parameter, der angibt, wie AWS DMS Nullwerte verarbeitet. Bei der Verarbeitung des Nullwerts können Sie diesen Parameter verwenden, um eine benutzerdefinierte Zeichenfolge als null zu übergeben, wenn Sie in das Ziel schreiben. Wenn Zielspalten beispielsweise auf Null gesetzt werden können, können Sie diese Option verwenden, um zwischen dem leeren Zeichenfolgenwert und dem Nullwert zu unterscheiden. Wenn Sie diesen Parameterwert also auf die leere Zeichenfolge (" " oder "") setzen, behandelt AWS DMS die leere Zeichenfolge als Nullwert und nicht als NULL.</p> <p>Standardwert: NULL</p> <p>Gültige Werte: jede gültige Zeichenfolge</p> <p>Beispiel: <code>--s3-settings '{"CsvNullValue": " "}'</code></p>
AddColumnName	<p>Ein optionaler Parameter, den sie, sofern er auf <code>true</code> oder <code>y</code> festgelegt ist, verwenden können, um der CSV-Ausgabedatei Informationen zu Spaltennamen hinzuzufügen.</p> <p>Sie können diesen Parameter nicht zusammen mit <code>PreserveTransactions</code> oder <code>CdcPath</code> verwenden.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p>

Option	Beschreibung
	Beispiel: <code>--s3-settings '{"AddColumnName": true}'</code>
AddTrailingPaddingCharacter	<p>Verwenden Sie die S3-Zielendpunkteinstellung AddTrailingPaddingCharacter , um Zeichenfolgedaten aufzufüllen. Der Standardwert ist false.</p> <p>Typ: Boolesch</p> <p>Beispiel: <code>--s3-settings '{"AddTrailingPaddingCharacter": true}'</code></p>
BucketFolder	<p>Ein optionaler Parameter zum Festlegen eines Ordernamens im S3-Bucket. Wenn angegeben, werden Zielobjekte als CSV- oder Parquet-Dateien im Pfad <i>BucketFolder /schema_name /table_name</i> / erstellt. Wenn dieser Parameter nicht festgelegt ist, wird der Pfad <i>schema_name /table_name</i> / verwendet.</p> <p>Beispiel: <code>--s3-settings '{"BucketFolder": "testFolder"}'</code></p>
BucketName	<p>Der Name des S3-Buckets, in dem S3-Zielobjekte als CSV- oder Parquet-Dateien erstellt werden.</p> <p>Beispiel: <code>--s3-settings '{"BucketName": "buckettest"}'</code></p>


Option	Beschreibung
CannedAc1For0bjects	<p>Ermöglicht es AWS DMS, eine vordefinierte Zugriffskontrollliste für Objekte festzulegen, die im S3-Bucket als CSV- oder Parquet-Dateien erstellt wurden. Weitere Informationen zu festgelegten ACLs von Amazon S3 finden Sie unter Festgelegte ACL im Amazon-S3-Entwicklerhandbuch.</p> <p>Standardwert: KEINE</p> <p>Gültige Werte für dieses Attribut sind: NONE; PRIVATE; PUBLIC_READ; PUBLIC_READ_WRITE; AUTHENTICATED_READ; AWS_EXEC_READ; BUCKET_OWNER_READ; BUCKET_OWNER_FULL_CONTROL.</p> <p>Beispiel: <code>--s3-settings '{"CannedAc1For0bjects": "PUBLIC_READ"}'</code></p>


Option	Beschreibung
CdcInsertsOnly	<p data-bbox="472 226 1484 695">Ein optionaler Parameter während der Erfassung von Änderungsdaten (Change Data Capture, CDC), um nur die INSERT-Operationen in die CSV- oder Parquet-Ausgabedateien zu schreiben. Standardmäßig (die <code>false</code>-Einstellung) enthält das erste Feld in einem CSV- oder PARTITION-Datensatz den Buchstaben I (INSERT), U (UPDATE) oder D (DELETE). Dieser Buchstabe gibt an, ob die Zeile für einen CDC-Ladenvorgang in das Ziel in der Quelldatenbank eingefügt, aktualisiert oder gelöscht wurde. Wenn <code>cdcInsertsOnly</code> auf <code>y</code> oder <code>true</code> festgelegt wurde, werden nur INSERTs aus der Quelldatenbank zur CSV- oder Parquet-Datei migriert.</p> <p data-bbox="472 737 1507 1157">Im Fall von CSV-Dateien ist die Art der Aufzeichnung dieser INSERTs vom Wert für <code>IncludeOpForFullLoad</code> abhängig. Wenn <code>IncludeOpForFullLoad</code> auf <code>true</code> festgelegt ist, wird das erste Feld jedes CDC-Datensatzes auf I festgelegt, um die INSERT-Operation an der Quelle anzugeben. Wenn <code>IncludeOpForFullLoad</code> auf <code>false</code> festgelegt ist, wird jeder CDC-Datensatz ohne erstes Feld geschrieben, um die INSERT-Operation an der Quelle anzugeben. Weitere Informationen darüber, wie diese Parameter miteinander funktionieren, finden Sie unter Angabe von Quelldatenbankoperationen in migrierten S3-Daten.</p> <p data-bbox="472 1199 776 1230">Standardwert: <code>false</code></p> <p data-bbox="472 1272 984 1304">Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p data-bbox="472 1346 1360 1377">Beispiel: <code>--s3-settings '{"CdcInsertsOnly": true}'</code></p>

Option	Beschreibung
<code>CdcInsertsAndUpdates</code>	<p>Aktiviert eine CDC (Change Data Capture)-Last zum Schreiben von INSERT- und UPDATE-Operationen in .csv- oder .parquet-Ausgabeteilen (Columnar Storage). Die Standardeinstellung ist <code>false</code>, wenn jedoch <code>cdcInsertsAndUpdates</code> auf <code>true</code> oder <code>y</code> festgelegt ist, werden INSERTs und UPDATEs aus der Quelldatenbank in die .csv- oder .parquet-Datei migriert.</p> <p>Nur für das CSV-Dateiformat hängt die Art und Weise, wie diese INSERTs und UPDATEs aufgezeichnet werden, vom Wert des <code>includeOpForFullLoad</code>-Parameters ab. Wenn <code>includeOpForFullLoad</code> auf <code>true</code> gesetzt ist, wird das erste Feld jedes CDC-Datensatzes entweder auf <code>I</code> oder <code>U</code> gesetzt, um INSERT- und UPDATE-Operationen an der Quelle anzuzeigen. Wenn <code>includeOpForFullLoad</code> jedoch auf <code>false</code> gesetzt ist, werden CDC-Datensätze ohne Anzeige von INSERT- oder UPDATE-Operationen an der Quelle geschrieben.</p> <p>Weitere Informationen darüber, wie diese Parameter miteinander funktionieren, finden Sie unter Angabe von Quelldatenbankoperationen in migrierten S3-Daten.</p> <div data-bbox="472 1182 1507 1545" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p><code>CdcInsertsOnly</code> und <code>cdcInsertsAndUpdates</code> können nicht beide für denselben Endpunkt auf „true“ gesetzt werden. Setzen Sie entweder <code>cdcInsertsOnly</code> oder <code>cdcInsertsAndUpdates</code> auf <code>true</code> für denselben Endpunkt, aber nicht beide.</p></div> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Beispiel: <code>--s3-settings '{"CdcInsertsAndUpdates": true}'</code></p>


Option	Beschreibung
CdcPath	<p>Gibt den Ordnerpfad von CDC-Dateien an. Für eine S3-Quelle ist diese Einstellung erforderlich, wenn eine Aufgabe Änderungsdaten erfasst; ansonsten ist sie optional. Wenn CdcPath festgelegt ist, liest DMS CDC-Dateien aus diesem Pfad und repliziert die Datenänderungen auf dem Zielpunkt. Wenn Sie für ein S3-Ziel <code>PreserveTransactions</code> auf „true“ gesetzt haben, prüft DMS, ob Sie für diesen Parameter einen Ordnerpfad auf Ihrem S3-Ziel festgelegt haben, in dem DMS die Transaktionsreihenfolge für den CDC-Ladevorgang speichern kann. DMS erstellt diesen CDC-Ordnerpfad in Ihrem S3-Zielarbeitsverzeichnis oder an dem durch <code>BucketFolder</code> und <code>BucketName</code> angegebenen S3-Zielort.</p> <p>Sie können diesen Parameter nicht zusammen mit <code>DatePartitionEnabled</code> oder <code>AddColumnName</code> verwenden.</p> <p>Typ: Zeichenfolge</p> <p>Wenn Sie beispielsweise <code>CdcPath</code> als <code>MyChangedData</code> und <code>BucketName</code> als <code>MyTargetBucket</code>, <code>BucketFolder</code> aber nicht angeben, erstellt DMS den folgenden CDC-Ordnerpfad: <code>MyTargetBucket/MyChangedData</code>.</p> <p>Wenn Sie denselben <code>CdcPath</code> und <code>BucketName</code> als <code>MyTargetBucket</code> sowie <code>BucketFolder</code> als <code>MyTargetData</code> angeben, erstellt DMS den folgenden CDC-Ordnerpfad: <code>MyTargetBucket/MyTargetData/MyChangedData</code>.</p> <div data-bbox="472 1438 1507 1808"><p> Note</p><p>Diese Einstellung wird in AWS DMS Version 3.4.2 und höher unterstützt.</p><p>Bei der Erfassung von Datenänderungen in der Reihenfolge der Transaktionen speichert DMS die Zeilenänderungen immer in CSV-Dateien, unabhängig vom Wert der S3-Einstellung „DataFormat“ auf dem Ziel. DMS speichert Datenänderungen bei</p></div>

Option	Beschreibung
	Verwendung von Parquet-Dateien nicht in der Reihenfolge der Transaktionen.
CdcMaxBatchInterval	<p>Bedingung für die maximale Intervalllänge, definiert in Sekunden, für die Ausgabe einer Datei in Amazon S3.</p> <p>Standardwert: 60 Sekunden</p> <p>Wenn <code>CdcMaxBatchInterval</code> und <code>CdcMinFileSize</code> angegeben sind, wird das Schreiben der Datei durch die Parameterbedingung ausgelöst, die zuerst erfüllt wird.</p>
CdcMinFileSize	<p>Bedingung für die minimale Dateigröße, definiert in Kilobyte, für die Ausgabe einer Datei in Amazon S3.</p> <p>Standardwert: 32000 KB</p> <p>Wenn <code>CdcMinFileSize</code> und <code>CdcMaxBatchInterval</code> angegeben sind, wird das Schreiben der Datei durch die Parameterbedingung ausgelöst, die zuerst erfüllt wird.</p>

Option	Beschreibung
PreserveTransactions	<p>Wenn hier <code>true</code> festgelegt ist, speichert DMS die Transaktionsreihenfolge für Change Data Capture (CDC) auf dem Amazon-S3-Ziel, das durch <code>CdcPath</code> angegeben ist.</p> <p>Sie können diesen Parameter nicht zusammen mit <code>DatePartitionEnabled</code> oder <code>AddColumnName</code> verwenden.</p> <p>Typ: Boolesch</p> <p>Bei der Erfassung von Datenänderungen in der Reihenfolge der Transaktionen speichert DMS die Zeilenänderungen immer in CSV-Dateien, unabhängig vom Wert der S3-Einstellung „DataFormat“ auf dem Ziel. DMS speichert Datenänderungen bei Verwendung von Parquet-Dateien nicht in der Reihenfolge der Transaktionen.</p> <div data-bbox="472 877 1507 1094"><p> Note</p><p>Diese Einstellung wird in AWS DMS Version 3.4.2 und höher unterstützt.</p></div>

Option	Beschreibung
<code>IncludeOpForFullLoad</code>	<p>Ein optionaler Parameter während eines vollständigen Ladevorgangs, um die INSERT-Operationen nur zu CSV-Ausgabedateien zu schreiben.</p> <p>Bei vollständigen Ladevorgängen können Datensätze nur eingefügt werden. Standardmäßig (Einstellung <code>false</code>) werden in diesen Ausgabedateien für einen vollständigen Ladevorgang keine Informationen aufgezeichnet, um anzugeben, dass die Zeilen in der Quelldatenbank eingefügt wurden. Wenn <code>IncludeOpForFullLoad</code> auf <code>true</code> oder <code>y</code> festgelegt ist, wird die INSERT-Operation als I-Anmerkung im ersten Feld der CSV-Datei aufgezeichnet.</p> <div data-bbox="472 716 1507 1081" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Dieser Parameter funktioniert mit <code>CdcInsertsOnly</code> oder <code>CdcInsertsAndUpdates</code> nur für die Ausgabe zu <code>.csv</code>-Dateien. Weitere Informationen darüber, wie diese Parameter miteinander funktionieren, finden Sie unter Angabe von Quelldatenbankoperationen in migrierten S3-Daten.</p></div> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Beispiel: <code>--s3-settings '{"IncludeOpForFullLoad": true}'</code></p>
<code>CompressionType</code>	<p>Ein optionaler Parameter verwendet, sofern er auf GZIP festgelegt ist, GZIP zum Komprimieren der CSV- oder Parquet-Zieldateien. Wenn dieser Parameter auf den Standardwert gesetzt ist, bleiben die Dateien unkomprimiert.</p> <p>Standardwert: <code>NONE</code></p> <p>Gültige Werte: <code>GZIP</code> oder <code>NONE</code>.</p> <p>Beispiel: <code>--s3-settings '{"CompressionType": "GZIP"}</code></p>

Option	Beschreibung
CsvDelimiter	<p>Das Trennzeichen, das zum Trennen von Spalten in den CSV-Quelldateien dient. Standardmäßig wird ein Komma (,) verwendet.</p> <p>Beispiel: <code>--s3-settings '{"CsvDelimiter": ","}'</code></p>
CsvRowDelimiter	<p>Das Trennzeichen, das zum Trennen von Zeilen in den CSV-Quelldateien dient. Standardmäßig wird ein Zeilenumbruch (\n) verwendet.</p> <p>Beispiel: <code>--s3-settings '{"CsvRowDelimiter": "\n"}'</code></p>
MaxFileSize	<p>Ein Wert, der die maximale Größe (in KB) der CSV-Datei angibt, die erstellt werden soll, während die Migration zum S3-Ziel in der Volllast-Phase ausgeführt wird.</p> <p>Standardwert: 1048576 KB (1 GB)</p> <p>Gültige Werte: 1 bis 1 048 576</p> <p>Beispiel: <code>--s3-settings '{"MaxFileSize": 512}'</code></p>
Rfc4180	<p>Ein optionaler Parameter, der verwendet wird, um das Verhalten so einzustellen, dass es RFC für Daten entspricht, die nur im CSV-Format zu Amazon S3 migriert werden. Wenn für diesen Wert bei Verwendung von Amazon S3 als Ziel <code>true</code> oder <code>y</code> eingestellt ist und die Daten Fragezeichen, Kommas oder Zeilenumbruchzeichen enthalten, umschließt AWS DMS die gesamte Spalte mit einem zusätzlichen Paar von Anführungszeichen ("). Jedes Anführungszeichen innerhalb der Daten wird zweimal wiederholt. Diese Formatierung entspricht RFC 4180.</p> <p>Standardwert: <code>true</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Beispiel: <code>--s3-settings '{"Rfc4180": false}'</code></p>


Option	Beschreibung
EncryptionMode	<p>Der serverseitige Verschlüsselungsmodus, mit dem Sie Ihre CSV- oder Parquet-Objektdateien, die Sie nach S3 kopiert haben, verschlüsseln möchten. Die gültigen Werte sind SSE_S3 (S3-serverseitige Verschlüsselung) oder SSE_KMS (KMS-Verschlüsselung). Wenn Sie SSE_KMS auswählen, legen Sie den <code>ServerSideEncryptionKmsKeyId</code> -Parameter auf den Amazon-Ressourcennamen (ARN) fest, so dass der KMS-Schlüssel für die Verschlüsselung verwendet wird.</p> <div data-bbox="472 590 1507 953" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Sie können auch den CLI-Befehl <code>modify-endpoint</code> verwenden, um den Wert des Attributs <code>EncryptionMode</code> für einen vorhandenen Endpunkt von SSE_KMS in SSE_S3 zu ändern. Sie können jedoch nicht den Wert für <code>EncryptionMode</code> von SSE_S3 in SSE_KMS ändern.</p> </div> <p>Standardwert: SSE_S3</p> <p>Gültige Werte: SSE_S3 oder SSE_KMS.</p> <p>Beispiel: <code>--s3-settings '{"EncryptionMode": SSE_S3}'</code></p>
ServerSideEncryptionKmsKeyId	<p>Wenn Sie <code>EncryptionMode</code> auf SSE_KMS setzen, legen Sie für diesen Parameter den Amazon-Ressourcennamen (ARN) für den KMS-Schlüssel fest. Sie finden diesen ARN, indem Sie den Schlüssel-Alias in der Liste der AWS KMS-Schlüssel, die für Ihr Konto erstellt wurden, auswählen. Wenn Sie den Schlüssel erstellen, müssen Sie ihm auch bestimmte Richtlinien und Rollen, die mit diesem KMS-Schlüssel verknüpft sind, zuweisen. Weitere Informationen finden Sie unter Erstellen von AWS KMS-Schlüsseln zum Verschlüsseln von Amazon-S3-Zielobjekten.</p> <p>Beispiel: <code>--s3-settings '{"ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'</code></p>

Option	Beschreibung
DataFormat	<p>Das Ausgabeformat für die Dateien, die AWS DMS zum Erstellen von S3-Objekten verwendet. Für Amazon-S3-Ziele unterstützt AWS DMS entweder CSV- oder Parquet-Dateien. Die Parquet-Dateien verfügen über ein binäres spaltenbasiertes Speicherformat mit effizienten Komprimierungsoptionen und schnellerer Abfrageleistung. Weitere Informationen über Parquet-Dateien finden Sie unter https://parquet.apache.org/.</p> <p>Standardwert: <code>csv</code></p> <p>Gültige Werte: <code>csv</code> oder <code>parquet</code>.</p> <p>Beispiel: <code>--s3-settings '{"DataFormat": "parquet"}'</code></p>
EncodingType	<p>Der Parquet-Kodierungstyp. Die Kodierungstypoptionen umfassen Folgendes:</p> <ul style="list-style-type: none">• <code>rle-dictionary</code> – Diese Verzeichniskodierung verwendet eine Kombination aus Bit-Packing und Run-Length-Kodierung, um sich wiederholende Werte effizienter zu speichern.• <code>plain</code> – Keine Kodierung.• <code>plain-dictionary</code> – Diese Verzeichniskodierung erstellt ein Verzeichnis der Werte, die in einer bestimmten Spalte vorkommen. Das Verzeichnis wird auf einer Verzeichnisseite für die einzelnen Spaltenblöcke gespeichert. <p>Standardwert: <code>rle-dictionary</code></p> <p>Gültige Werte: <code>rle-dictionary</code>, <code>plain</code> oder <code>plain-dictionary</code></p> <p>Beispiel: <code>--s3-settings '{"EncodingType": "plain-dictionary"}'</code></p>

Option	Beschreibung
<code>DictPageSizeLimit</code>	<p>Die zulässige Maximalgröße (in Bytes) für eine Verzeichnisseite in einer Parquet-Datei. Wenn eine Verzeichnisseite diesen Wert überschreitet, verwendet die Seite die einfache Kodierung.</p> <p>Standardwert: 1.024.000 (1 MB)</p> <p>Zulässige Werte: Jede gültige Ganzzahl</p> <p>Beispiel: <code>--s3-settings '{"DictPageSizeLimit": 2,048,000}'</code></p>
<code>RowGroupLength</code>	<p>Die Anzahl der Zeilen einer Zeilengruppe in einer Parquet-Datei.</p> <p>Standardwert: 10.024 (10 KB)</p> <p>Zulässige Werte: Jede gültige Ganzzahl</p> <p>Beispiel: <code>--s3-settings '{"RowGroupLength": 20,048}'</code></p>
<code>DataPageSize</code>	<p>Die zulässige Maximalgröße (in Bytes) für eine Datenseite in einer Parquet-Datei.</p> <p>Standardwert: 1.024.000 (1 MB)</p> <p>Zulässige Werte: Jede gültige Ganzzahl</p> <p>Beispiel: <code>--s3-settings '{"DataPageSize": 2,048,000}'</code></p>
<code>ParquetVersion</code>	<p>Die Version des Parquet-Dateiformats.</p> <p>Standardwert: <code>PARQUET_1_0</code></p> <p>Gültige Werte: <code>PARQUET_1_0</code> oder <code>PARQUET_2_0</code> .</p> <p>Beispiel: <code>--s3-settings '{"ParquetVersion": "PARQUET_2_0"}</code></p>

Option	Beschreibung
<code>EnableStatistics</code>	<p>Legen Sie <code>true</code> oder <code>y</code> fest, um Statistiken über Parquet-Dateiseiten und Zeilengruppen zu aktivieren.</p> <p>Standardwert: <code>true</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p>Beispiel: <code>--s3-settings '{"EnableStatistics": false}'</code></p>
<code>TimestampColumnName</code>	<p>Ein optionaler Parameter, um eine Zeitstempelspalte in die S3-Zieldatensatzpunkte einzufügen.</p> <p>AWS DMS enthält eine zusätzliche <code>STRING</code>-Spalte in den CSV- oder Parquet-Objektdateien der migrierten Daten, wenn Sie für <code>TimestampColumnName</code> einen nicht leeren Wert festlegen.</p> <p>Bei einem vollständigen Ladevorgang enthält jede Zeile der Zeitstempelspalte einen Zeitstempel für den Zeitpunkt, an dem die Daten von DMS von der Quelle zum Ziel übertragen wurden.</p> <p>Bei einem CDC-Ladevorgang enthält jede Zeile der Zeitstempelspalte den Zeitstempel für den Commit dieser Zeile in der Quelldatenbank.</p> <p>Das Zeichenfolgeformat für den Wert dieser Zeitstempelspalte ist <code>yyyy-MM-dd HH:mm:ss.SSSSSS</code>. Standardmäßig wird die Genauigkeit für diesen Wert in Mikrosekunden angegeben. Für einen CDC-Ladevorgang hängt die Rundung für die Genauigkeit vom Commit-Zeitstempel ab, der von DMS für die Quell-Datenbank unterstützt wird.</p> <p>Wenn für den Parameter <code>AddColumnName</code> <code>true</code> festgelegt ist, bezieht DMS auch den Namen für die Zeitstempelspalte ein, den Sie als den nicht leeren Wert von <code>TimestampColumnName</code> festlegen.</p> <p>Beispiel: <code>--s3-settings '{"TimestampColumnName": "TIMESTAMP"}</code></p>

Option	Beschreibung
UseTaskStartTimeForFullLoadTimestamp	<p>Bei Einstellung von <code>true</code> verwendet dieser Parameter die Startzeit der Aufgabe als Zeitstempel-Spaltenwert anstelle der Zeit, zu der Daten in das Ziel geschrieben werden. Bei einem vollständigen Ladevorgang, wenn <code>UseTaskStartTimeForFullLoadTimestamp</code> auf <code>true</code> gesetzt ist, enthält jede Zeile der Zeitstempel-Spalte die Startzeit der Aufgabe. Bei CDC-Ladevorgängen enthält jede Zeile der Spalte Zeitstempel die Übergabezeit der Transaktion.</p> <p>Wenn <code>UseTaskStartTimeForFullLoadTimestamp</code> auf <code>false</code> gesetzt ist, wird der Zeitstempel des vollständigen Ladevorgangs in der Zeitstempel-Spalte um den Zeitpunkt des Eintreffens der Daten im Ziel erhöht.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code></p> <p>Beispiel: <code>--s3-settings '{"UseTaskStartTimeForFullLoadTimestamp": true}'</code></p> <p><code>UseTaskStartTimeForFullLoadTimestamp: true</code> hilft dabei, das S3-Ziel <code>TimestampColumnName</code> für einen vollständigen Ladevorgang mit <code>TimestampColumnName</code> für einen CDC-Ladevorgang sortierbar zu machen.</p>

Option	Beschreibung
ParquetTimestampInMillisecond	<p data-bbox="467 226 1421 359">Ein optionaler Parameter, der die Genauigkeit der <code>TIMESTAMP</code> -Spaltenwerte angibt, die in eine S3-Objektdatei im <code>.parquet</code>-Format geschrieben werden.</p> <p data-bbox="467 405 1485 583">Wenn dieses Attribut auf <code>true</code> oder <code>y</code> festgelegt ist, schreibt AWS DMS alle <code>TIMESTAMP</code> -Spalten in eine <code>.parquet</code>-formatierte Datei mit einer Genauigkeit von Millisekunden. Andernfalls schreibt DMS sie mit einer Genauigkeit in Mikrosekunden.</p> <p data-bbox="467 627 1485 852">Derzeit können Amazon Athena und AWS Glue nur Millisekunden-Genauigkeit für <code>TIMESTAMP</code> -Werte verarbeiten. Setzen Sie dieses Attribut für <code>.parquet</code>-formatierte S3-Endpunkt-Objektdateien nur dann auf „<code>true</code>“, wenn Sie vorhaben, die Daten mithilfe von Athena oder AWS Glue abzufragen oder zu verarbeiten.</p> <div data-bbox="472 894 1507 1381" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p data-bbox="500 930 621 968"> Note</p><ul data-bbox="548 1010 1469 1339" style="list-style-type: none"><li data-bbox="548 1010 1469 1140">• AWS DMS verwendet für alle <code>TIMESTAMP</code> -Spaltenwerte, die in eine S3-Datei im <code>CSV</code>-Format geschrieben werden, eine Genauigkeit in Mikrosekunden.<li data-bbox="548 1161 1469 1339">• Die Einstellung dieses Attributs hat keine Auswirkung auf das Zeichenfolgeformat des Zeitstempelspaltenwerts; der durch Festlegen des <code>TimestampColumnName</code> -Attributs eingefügt wird.</div> <p data-bbox="467 1451 776 1488">Standardwert: <code>false</code></p> <p data-bbox="467 1533 984 1570">Zulässige Werte: <code>true</code>, <code>false</code>, <code>y</code>, <code>n</code></p> <p data-bbox="467 1614 1386 1696">Beispiel: <code>--s3-settings '{"ParquetTimestampInMillisecond": true}'</code></p>

Option	Beschreibung
GlueCatalogGeneration	<p>Um einen AWS Glue Data Catalog zu generieren, setzen Sie diese Endpunkteinstellung auf <code>true</code>.</p> <p>Standardwert: <code>false</code></p> <p>Zulässige Werte: <code>true</code>, <code>false</code></p> <p>Beispiel: <code>--s3-settings '{"GlueCatalogGeneration": true}'</code></p> <p>Hinweis: Verwenden Sie <code>GlueCatalogGeneration</code> nicht mit <code>PreserveTransactions</code> und <code>CdcPath</code>.</p>

Verwenden von AWS Glue Data Catalog mit einem Amazon-S3-Ziel für AWS DMS

AWS Glue ist ein Service, der einfache Möglichkeiten zur Kategorisierung von Daten bietet und aus einem Metadaten-Repository besteht, das als AWS Glue Data Catalog bezeichnet wird. Sie können AWS Glue Data Catalog in Ihren Amazon-S3-Zielendpunkt integrieren und Amazon-S3-Daten über andere AWS-Services wie Amazon Athena abfragen. Amazon Redshift kann mit AWS Glue verwendet werden, AWS DMS unterstützt dies jedoch nicht als vordefinierte Option.

Um den Datenkatalog zu generieren, setzen Sie die Endpunkteinstellung `GlueCatalogGeneration` auf `true`, wie im folgenden AWS CLI-Beispiel zu sehen ist.

```
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}'
```

Für eine Replikationsaufgabe mit vollständigem Ladevorgang, die Daten des Typs `csv` umfasst, legen Sie für `IncludeOpForFullLoad` `true` fest.

Verwenden Sie `GlueCatalogGeneration` nicht mit `PreserveTransactions` und `CdcPath`. Der AWS Glue-Crawler kann die verschiedenen Schemas von Dateien, die unter dem angegebenen `CdcPath` gespeichert sind, nicht abgleichen.

Damit Amazon Athena Ihre Amazon-S3-Daten indexiert und Sie Ihre Daten mithilfe von Standard-SQL-Abfragen über Amazon Athena abfragen können, muss die dem Endpunkt zugeordnete IAM-Rolle über die folgende Richtlinie verfügen:

```
{
  "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_*"
}
]
```

Referenzen

- Weitere Informationen zum Konfigurieren einer Regel in AWS Glue finden Sie unter <https://docs.aws.amazon.com/glue/latest/dg/components-key-concepts.html> im AWS Glue-Entwicklerhandbuch.
- Weitere Informationen zu AWS Glue Data Catalog finden Sie unter [Komponenten](#) im AWS Glue-Entwicklerhandbuch.

Verwenden von Datenverschlüsselung, Parquet-Dateien und CDC auf Ihrem Amazon-S3-Ziel

Sie können die Einstellungen für den S3-Zielendpunkt verwenden, um Folgendes zu konfigurieren:

- Einen benutzerdefinierten KMS-Schlüssel für die Verschlüsselung Ihrer S3-Zielobjekte
- Parquet-Dateien als Speicherformat für S3-Zielobjekte
- Change Data Capture (CDC), einschließlich Transaktionsreihenfolge auf dem S3-Ziel
- Integration von AWS Glue Data Catalog in Ihren Amazon-S3-Zielendpunkt und Abfrage von Amazon-S3-Daten über andere Services wie Amazon Athena

AWS KMS-Schlüsseleinstellungen für die Datenverschlüsselung

Die folgenden Beispiele veranschaulichen die Konfiguration eines benutzerdefinierten KMS-Schlüssels, um Ihre S3-Zielobjekte zu verschlüsseln. Um zu starten, können Sie den folgenden `create-endpoint-CLI`-Befehl ausführen.

```
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"}'
```

Das durch die `--s3-settings`-Option angegebene JSON-Objekt definiert zwei Parameter. Der eine ist ein `EncryptionMode`-Parameter mit dem Wert `SSE_KMS`. Der andere ist ein `ServerSideEncryptionKmsKeyId`-Parameter mit dem Wert `arn:aws:kms:us-east-1:111122223333:key/72abb6fb-1e49-4ac1-9aed-c803dfcc0480`. Bei diesem Wert handelt es sich um einen Amazon-Ressourcenname (ARN) für Ihren benutzerdefinierten KMS-Schlüssel. Bei einem S3-Ziel legen Sie auch zusätzliche Einstellungen fest. Diese identifizieren die Zugriffsrolle des Servers, legen die Trennzeichen für das CSV-Objektspeicherformat fest und geben den Speicherort sowie den Namen des Buckets zum Speichern von S3-Zielobjekten an.

Standardmäßig erfolgt S3-Datenverschlüsselung mithilfe von serverseitiger S3-Verschlüsselung. Für das S3-Ziel des vorherigen Beispiels ist dies gleichbedeutend mit der Angabe der Endpunkteinstellungen, wie im folgenden Beispiel dargestellt.

```
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"}'
```

Weitere Informationen über das Arbeiten mit serverseitiger S3-Verschlüsselung finden Sie unter [Schützen von Daten mithilfe serverseitiger Verschlüsselung](#).

Note

Sie können auch den CLI-Befehl `modify-endpoint` verwenden, um den Wert des Parameters `EncryptionMode` für einen vorhandenen Endpunkt von `SSE_KMS` in `SSE_S3` zu ändern. Sie können jedoch nicht den Wert für `EncryptionMode` von `SSE_S3` in `SSE_KMS` ändern.

Einstellungen für das Speichern von S3-Zielobjekten mithilfe von Parquet-Dateien

Das Standardformat für die Erstellung von S3-Zielobjekten ist CSV-Dateien. Die folgenden Beispiele zeigen einige Endpunkteinstellungen, um Parquet-Dateien als Format für die Erstellung von S3-Zielobjekten festzulegen. Sie können das Parquet-Dateiformat mit allen Standardeinstellungen angeben, wie im folgenden Beispiel gezeigt.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "DataFormat":
"parquet"}'
```

Hier wird der `DataFormat`-Parameter als `parquet` festgelegt, um das Format mit allen S3-Standards zu ermöglichen. Diese Standardwerte umfassen eine Verzeichniskodierung (`"EncodingType": "rle-dictionary"`), die eine Kombination aus Bit-Packing und Run-Length-Kodierung verwendet, um sich wiederholende Werte effizienter zu speichern.

Sie können zusätzliche Einstellungen für Optionen festlegen, die nicht zu den Standardeinstellungen gehören, wie im folgenden Beispiel gezeigt.

```
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 }'
```

Hier werden zusätzlich zu den Parametern für mehrere S3-Bucket-Standardoptionen und dem `DataFormat`-Parameter die folgenden zusätzlichen Parquet-Dateiparameter festgelegt:

- `EncodingType` – Auf eine Verzeichniskodierung (`plain-dictionary`) festgelegt, die die in den einzelnen Spalten auftretenden Werte in einem Block auf der Verzeichnisseite speichert, der auch nach Spalten organisiert ist.
- `DictPageSizeLimit` – Auf eine maximale Verzeichnisseitengröße von 3 MB festgelegt.
- `EnableStatistics` – Deaktiviert den Standard, der die Sammlung von Statistiken über Parquet-Dateiseiten und -Zeilengruppen ermöglicht.

Erfassen von Datenänderungen (CDC), einschließlich Transaktionsreihenfolge auf dem S3-Ziel

Bei Ausführung einer CDC-Aufgabe speichert AWS DMS standardmäßig alle in Ihrer Quelldatenbank (oder Ihren Quelldatenbanken) protokollierten Zeilenänderungen in einer oder mehreren Dateien für jede Tabelle. Jede Gruppe von Dateien mit Änderungen für dieselbe Tabelle befindet sich in einem einzelnen Zielverzeichnis, das dieser Tabelle zugeordnet ist. AWS DMS erstellt so viele Zielverzeichnisse, wie Datenbanktabellen auf den Amazon-S3-Zielendpunkt migriert werden. Die Dateien werden auf dem S3-Ziel in diesen Verzeichnissen ohne Berücksichtigung der Transaktionsreihenfolge gespeichert. Weitere Informationen zu den Dateinamenskonventionen, zu den Dateninhalten und zum Format finden Sie unter [Verwenden von Amazon S3 als Ziel für AWS Database Migration Service](#).

Um Änderungen an der Quelldatenbank so zu erfassen, dass auch die Transaktionsreihenfolge erfasst wird, können Sie S3-Endpunkteinstellungen angeben, die AWS DMS anweisen, die Zeilenänderungen für alle Datenbanktabellen in einer oder mehreren CSV-Dateien zu speichern, die je nach Transaktionsgröße erstellt werden. In diesen CSV-Transaktionsdateien sind alle Zeilenänderungen sequentiell in der Transaktionsreihenfolge für alle an den einzelnen Transaktionen beteiligten Tabellen aufgeführt. Diese Transaktionsdateien befinden sich alle in einem einzelnen Transaktionsverzeichnis, das Sie auch auf dem S3-Ziel angeben. In jeder Transaktionsdatei werden der Transaktionsvorgang und die Identität der Datenbank und der Quelltable für jede Zeilenänderung wie folgt als Teil der Zeilendaten gespeichert.

```
operation,table_name,database_schema_name,field_value,...
```

operation ist hier der Transaktionsvorgang für die geänderte Zeile, *table_name* der Name der Datenbanktabelle, in der die Zeile geändert wurde, *database_schema_name* der Name des Datenbankschemas, in dem sich die Tabelle befindet, und *field_value* der erste von einem oder mehreren Feldwerten, die die Daten für die Zeile angeben.

Im folgenden Beispiel einer Transaktionsdatei sind geänderte Zeilen für eine oder mehrere Transaktionen zu sehen, die sich auf zwei Tabellen beziehen.

```
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
```

Hier wird der Transaktionsvorgang für jede Zeile in der ersten Spalte durch I („Insert“ – Einfügen), U („Update“ – Aktualisieren) oder D („Delete“ – Löschen) angezeigt. Der Tabellennamen ist der zweite Spaltenwert (z. B. Names_03cdcad11a). Der Name des Datenbankschemas ist der Wert der dritten Spalte (z. B. rdsTempsdb). Die verbleibenden Spalten werden mit Ihren eigenen Zeilendaten gefüllt (z. B. 13, Daniel).

AWS DMS benennt die Transaktionsdateien, die auf dem Amazon-S3-Ziel erstellt werden, zudem mithilfe eines Zeitstempels gemäß der folgenden Namenskonvention.

```
CDC_TXN-timestamp.csv
```

timestamp ist hier der Zeitpunkt, zu dem die Transaktionsdatei erstellt wurde, wie im folgenden Beispiel dargestellt.

```
CDC_TXN-20201117153046033.csv
```

Dieser Zeitstempel im Dateinamen stellt sicher, dass die Transaktionsdateien in der Reihenfolge der Transaktionen erstellt und aufgeführt werden, wenn Sie sie in ihrem Transaktionsverzeichnis auflisten.

Note

Bei der Erfassung von Datenänderungen in der Reihenfolge der Transaktionen speichert AWS DMS die Zeilenänderungen immer in CSV-Dateien, unabhängig vom Wert der S3-Einstellung DataFormat auf dem Ziel. Bei Verwendung von Parquet-Dateien speichert AWS DMS Datenänderungen nicht in der Transaktionsreihenfolge.

Um die Häufigkeit von Schreibvorgängen auf ein Amazon-S3-Ziel während einer Datenreplikationsaufgabe zu steuern, können Sie die Einstellungen `CdcMaxBatchInterval` und `CdcMinFileSize` konfigurieren. Dies kann zu einer besseren Leistung bei der Analyse der Daten ohne zusätzliche überflüssige Aufgaben führen. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#)

So weisen Sie AWS DMS an, alle Zeilenänderungen in der Reihenfolge der Transaktionen zu speichern

1. Legen Sie für die S3-Einstellung `PreserveTransactions` auf dem Ziel `true` fest.
2. Legen Sie für die S3-Einstellung `CdcPath` auf dem Ziel einen relativen Ordnerpfad fest, in dem AWS DMS die CSV-Transaktionsdateien speichern soll.

AWS DMS erstellt diesen Pfad entweder unter dem standardmäßigen S3-Ziel-Bucket und -Arbeitsverzeichnis oder unter dem Bucket und Bucket-Ordner, die Sie über die S3-Einstellungen `BucketName` und `BucketFolder` auf dem Ziel angeben.

Angabe von Quelldatenbankoperationen in migrierten S3-Daten

Wenn AWS DMS Datensätze zu einem S3-Ziel migriert, kann es in jedem migrierten Datensatz ein zusätzliches Feld erstellen. Dieses zusätzliche Feld gibt die Operation an, die auf den Datensatz in der Quelldatenbank angewendet wird. Wie AWS DMS dieses erste Feld erstellt und festlegt, hängt vom Typ der Migrationsaufgabe und den Einstellungen von `includeOpForFullLoad`, `cdcInsertsOnly` und `cdcInsertsAndUpdates` ab.

Wenn im Falle eines vollständigen Ladevorgangs `includeOpForFullLoad` auf `true` gesetzt ist, erstellt AWS DMS in jedem CSV-Datensatz stets ein zusätzliches erstes Feld. Dieses Feld enthält den Buchstaben I (INSERT), um anzugeben, dass die Zeile in der Quelldatenbank eingefügt wurde. Wenn im Falle eines CDC-Ladevorgangs `cdcInsertsOnly` auf `false` (Standardeinstellung) gesetzt ist, erstellt AWS DMS ebenfalls in jedem CSV- oder Parquet-Datensatz stets ein zusätzliches erstes Feld. Dieses Feld enthält die Buchstaben I (INSERT), U (UPDATE) oder D (DELETE), um anzugeben, ob die Zeile in der Quelldatenbank eingefügt, aktualisiert oder gelöscht wurde.

In der folgenden Tabelle sehen Sie, wie sich die Einstellungen der Attribute `includeOpForFullLoad` und `cdcInsertsOnly` gemeinsam auf die Einstellung migrierter Datensätze auswirken.

Mit diesen Parametereinstellungen		Legt DMS die Zieldatensätze für CSV- und Parquet-Ausgaben wie folgt fest	
includeOpForFullLoad	cdcInsertsOnly	Für vollständige Ladevorgänge	Für CDC-Ladevorgänge
true	true	Wert des hinzugefügten ersten Felds ist auf I festgelegt	Wert des hinzugefügten ersten Felds ist auf I festgelegt
false	false	Kein hinzugefügtes Feld	Wert des hinzugefügten ersten Felds ist auf I, U oder D festgelegt
false	true	Kein hinzugefügtes Feld	Kein hinzugefügtes Feld
true	false	Wert des hinzugefügten ersten Felds ist auf I festgelegt	Wert des hinzugefügten ersten Felds ist auf I, U oder D festgelegt

Wenn `includeOpForFullLoad` und `cdcInsertsOnly` auf denselben Wert festgelegt sind, werden die Zieldatensätze entsprechend dem Attribut festgelegt, das die Datensatzeinstellungen für den aktuellen Migrationstyp steuert. Das Attribut ist `includeOpForFullLoad` für vollständige Ladevorgänge und `cdcInsertsOnly` für CDC-Ladevorgänge.

Wenn `includeOpForFullLoad` und `cdcInsertsOnly` auf verschiedene Werte festgelegt sind, stellt AWS DMS Konsistenz bei den Zieldatensatzeinstellungen sowohl für vollständige als auch für CDC-Ladevorgänge her. Hierzu werden die Datensatzeinstellungen des CDC-Ladevorgangs an die Datensatzeinstellungen früherer vollständiger Ladevorgänge angepasst, wie durch `includeOpForFullLoad` angegeben.

Nehmen Sie beispielsweise an, dass für einen vollständigen Ladevorgang ein erstes Feld hinzugefügt wird, um einen eingefügten Datensatz anzuzeigen. In diesem Fall wird einem folgenden CDC-Ladevorgang ein erstes Feld hinzugefügt, das einen eingefügten, aktualisierten oder gelöschten Datensatz in der Quelle anzeigt wie zutreffend. Nehmen Sie nun an, dass für einen vollständigen

Ladevorgang nicht festgelegt wurde, dass zur Anzeige eines eingefügten Datensatzes ein erstes Feld hinzugefügt wird. In diesem Fall wird für diesen CDC-Ladevorgang ebenfalls festgelegt, dass jedem Datensatz unabhängig von der entsprechenden Datensatzoperation in der Quelle ein erstes Feld hinzugefügt wird.

Ebenso hängt die Art und Weise, wie DMS ein zusätzliches erstes Feld erstellt und einstellt, von den Einstellungen von `includeOpForFullLoad` und `cdcInsertsAndUpdates` ab. In der folgenden Tabelle können Sie sehen, wie sich die Einstellungen der Attribute `includeOpForFullLoad` und `cdcInsertsAndUpdates` gemeinsam auf die Einstellung migrierter Datensätze in diesem Format auswirken.

Mit diesen Parametereinstellungen		DMS legt Zieldatensätze für CSV-Ausgaben wie folgt fest	
<code>includeOpForFullLoad</code>	<code>cdcInsertsAndUpdates</code>	Für vollständige Ladevorgänge	Für CDC-Ladevorgänge
<code>true</code>	<code>true</code>	Wert des hinzugefügten ersten Felds ist auf I festgelegt	Wert des hinzugefügten ersten Felds ist auf I oder U gesetzt
<code>false</code>	<code>false</code>	Kein hinzugefügtes Feld	Wert des hinzugefügten ersten Felds ist auf I, U oder D festgelegt
<code>false</code>	<code>true</code>	Kein hinzugefügtes Feld	Wert des hinzugefügten ersten Felds ist auf I oder U gesetzt
<code>true</code>	<code>false</code>	Wert des hinzugefügten ersten Felds ist auf I festgelegt	Wert des hinzugefügten ersten Felds ist auf I, U oder D festgelegt

Zieldatentypen für S3 Parquet

In der folgenden Tabelle sind die Parquet-Zieldatentypen, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuordnung von AWS DMS-Datentypen aufgeführt.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

AWS DMS-Datentyp	S3-Parquet-Datentyp
BYTES	BINARY
DATE	DATE32
TIME	TIME32
DATETIME	TIMESTAMP
INT1	INT8
INT2	INT16
INT4	INT32
INT8	INT64
NUMERIC	DECIMAL
REAL4	FLOAT
REAL8	DOUBLE
STRING	STRING
UINT1	UINT8
UINT2	UINT16
UINT4	UINT32
UINT8	UINT64

AWS DMS-Datentyp	S3-Parquet-Datentyp
WSTRING	STRING
BLOB	BINARY
NCLOB	STRING
CLOB	STRING
BOOLEAN	BOOL

Verwenden einer Amazon-DynamoDB-Datenbank als Ziel für AWS Database Migration Service

Mit AWS DMS können Sie Daten in eine Amazon-DynamoDB-Tabelle migrieren. Amazon DynamoDB ist ein vollständig verwalteter NoSQL-Datenbank-Service, der schnelle und planbare Leistung mit nahtloser Skalierbarkeit bietet. AWS DMS unterstützt die Verwendung von relationalen Datenbanken oder MongoDB als Quelle.

In DynamoDB sind Tabellen, Elemente und Attribute die zentralen Komponenten, mit denen Sie arbeiten. Eine Tabelle ist eine Sammlung von Elementen und jedes Element wiederum eine Sammlung von Attributen. DynamoDB nutzt die als Partitionsschlüssel bezeichneten Primärschlüssel, um jedes Element in einer Tabelle eindeutig zu identifizieren. Sie können auch Schlüssel und sekundäre Indizes verwenden, um die Abfrage flexibler zu gestalten.

Verwenden Sie Objektzuordnungen zum Migrieren Ihrer Daten aus einer Quelldatenbank in eine DynamoDB-Zieltabelle. Anhand von Objektzuweisungen können Sie bestimmen, wo sich die Quelldaten in der Zieltabelle befinden.

Wenn AWS DMS Tabellen auf einem DynamoDB-Zielendpunkt erstellt, werden dabei so viele Tabellen wie im Quelldatenbank-Endpunkt erstellt. AWS DMS legt auch mehrere DynamoDB-Parameterwerte fest. Die Kosten für die Erstellung der Tabelle sind abhängig von der Datenmenge und der Anzahl der zu migrierenden Tabellen.


Note

Die Option SSL-Modus in der AWS DMS-Konsole oder -API ist für einige Daten-Streaming- und NoSQL-Services wie Kinesis und DynamoDB nicht anwendbar. Diese

sind standardmäßig sicher, daher zeigt AWS DMS für den SSL-Modus „Keine“ an (SSL-Modus=Keine). Sie müssen keine zusätzliche Konfiguration für Ihren Endpunkt angeben, um SSL verwenden zu können. Wenn Sie beispielsweise DynamoDB als Zielendpunkt verwenden, ist dies standardmäßig sicher. Alle API-Aufrufe an DynamoDB verwenden SSL, sodass keine zusätzliche SSL-Option am AWS DMS-Endpunkt erforderlich ist. Mithilfe des HTTPS-Protokolls, das AWS DMS standardmäßig verwendet, wenn eine Verbindung mit einer DynamoDB-Datenbank hergestellt wird, können Sie Daten sicher über SSL-Endpunkte einfügen und abrufen.

Um die Übertragungsgeschwindigkeit zu erhöhen, unterstützt AWS DMS Multi-Thread-Vollstadvorgänge in eine DynamoDB-Ziel-Instance. DMS unterstützt Multithreading u. a. mithilfe der folgenden Aufgabeneinstellungen:

- `MaxFullLoadSubTasks` – Geben Sie diese Option an, um die maximale Anzahl von Quelltabellen festzulegen, die parallel geladen werden sollen. DMS lädt jede Tabelle mithilfe einer dedizierten Unteraufgabe in die entsprechende DynamoDB-Zieltabelle. Der Standardwert ist 8. Der Höchstwert ist 49.
- `ParallelLoadThreads` – Verwenden Sie diese Option, um die Anzahl der Threads anzugeben, die AWS DMS zum Laden der einzelnen Tabellen in die DynamoDB-Zieltabellen verwendet. Der Standardwert ist 0 (Single-Thread). Der maximale Wert beträgt 200. Sie können eine Erhöhung dieses Höchstwerts anfordern.

 Note

DMS weist jedem Segment einer Tabelle einen eigenen Thread zum Laden zu. Geben Sie aus diesem Grund für `ParallelLoadThreads` die maximale Anzahl der Segmente an, die Sie für eine Tabelle in der Quelle verwenden werden.

- `ParallelLoadBufferSize` – Verwenden Sie diese Option, um die maximale Anzahl der Datensätze anzugeben, die in dem Puffer gespeichert werden sollen, den die parallelen Ladethreads zum Laden von Daten in das DynamoDB-Ziel verwenden. Der Standardwert lautet 50. Die maximale Wert ist 1.000. Verwenden Sie diese Einstellung mit `ParallelLoadThreads`; `ParallelLoadBufferSize` ist nur gültig, wenn es mehr als einen Thread gibt.
- Table-Mapping-Einstellungen für einzelne Tabellen – Verwenden Sie `table-settings`-Regeln, um einzelne Tabellen von der Quelle zu unterscheiden, die Sie parallel laden möchten. Verwenden Sie diese Regeln darüber hinaus, um festzulegen, wie die Zeilen jeder Tabelle für das

Multithread-Laden aufgegliedert werden sollen. Weitere Informationen finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Note

Wenn AWS DMS DynamoDB-Parameterwerte für eine Migrationsaufgabe festlegt, wird der Parameterwert für Lesekapazitätseinheiten (Read Capacity Units, RCU) standardmäßig auf 200 festgelegt.

Der Schreibkapazitätseinheiten (WCU)-Parameterwert ist auch festgelegt, aber sein Wert hängt von mehreren anderen Einstellungen ab:

- Der Standardwert für den WCU-Parameter lautet 200.
- Wenn die Einstellung für die `ParallelLoadThreads`-Aufgabe größer als 1 ist (der Standard ist 0), dann wird der WCU-Parameter auf das 200-fache des `ParallelLoadThreads`-Werts festgelegt.
- Für die genutzten Ressourcen fallen AWS DMS-Standardnutzungsgebühren an.

Migrieren aus einer relationalen Datenbank in eine DynamoDB-Tabelle

AWS DMS unterstützt die Migration von Daten in skalare DynamoDB-Datentypen. Bei der Migration von Daten aus einer relationalen Datenbank wie Oracle oder MySQL zu DynamoDB sollten Sie die Art und Weise, wie Sie diese Daten speichern, umstrukturieren.

Derzeit unterstützt AWS DMS die Umstrukturierung einzelner Tabellen in die skalaren DynamoDB-Datentypattribute. Wenn Sie Daten aus einer relationalen Datenbanktabelle zu DynamoDB migrieren, formatieren Sie die Daten aus einer Tabelle in die skalaren DynamoDB-Datentypattribute um. Diese Attribute können Daten aus mehreren Spalten aufnehmen und Sie können eine Spalte direkt einem Attribut zuweisen.

AWS DMS unterstützt die folgenden skalaren DynamoDB-Datentypen:

- String
- Zahl
- Boolesch

Note

NULL-Daten aus der Quelle werden auf dem Ziel ignoriert.

Voraussetzungen für die Verwendung von DynamoDB als Ziel für AWS Database Migration Service

Bevor Sie eine DynamoDB-Datenbank als Ziel für AWS DMS einsetzen können, müssen Sie eine IAM-Rolle erstellen. Diese IAM-Rolle muss AWS DMS zum Zugriff und zum Gewähren von Zugriff auf die DynamoDB-Tabellen berechtigen, in die migriert wird. In der folgenden IAM-Richtlinie sind die Mindestzugriffsberechtigungen dargestellt.

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

Die Rolle, die Sie für die Migration zu DynamoDB verwenden, muss die folgenden Berechtigungen aufweisen:

```
{
  "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": "*"
}
]
```

Einschränkungen bei Verwendung von DynamoDB als Ziel für AWS Database Migration Service

Die folgenden Einschränkungen gelten bei Verwendung von DynamoDB als Ziel:

- DynamoDB begrenzt die Genauigkeit des Number-Datentyps auf 38 Stellen. Speichern Sie alle Datentypen mit einer höheren Genauigkeit als Zeichenfolge. Sie müssen dies explizit mithilfe der Objektzuweisungsfunktion festlegen.
- Da DynamoDB keinen Date-Datentyp hat, werden die Daten, die diesen Datentyp verwenden, in Zeichenfolgen umgewandelt.
- DynamoDB lässt keine Aktualisierungen der Primärschlüssel-Attribute zu. Diese Einschränkung ist wichtig, wenn Sie die laufende Replikation mit Change Data Capture (CDC) verwenden, da sie zu unerwünschten Daten im Ziel führen kann. Je nachdem, wie Sie das Objekt zuweisen, kann eine CDC-Operation, die den Primärschlüssel aktualisiert, Folgendes tun. Es kann entweder fehlschlagen oder ein neues Element mit dem aktualisierten Primärschlüssel und unvollständigen Daten einfügen.
- AWS DMS unterstützt nur die Replikation von Tabellen mit nicht zusammengesetzten Primärschlüsseln. Die Ausnahme ist, wenn Sie eine Objektzuweisung für die Zieltabelle mit einem benutzerdefinierten Partitionsschlüssel oder Sortierschlüssel (oder beiden) festlegen.

- AWS DMS unterstützt keine LOB-Daten, es sei denn, es handelt sich um CLOB-Daten. Beim Migrieren von Daten wandelt AWS DMS CLOB-Daten in eine DynamoDB-Zeichenfolge um.
- Bei der Verwendung von DynamoDB als Ziel wird nur die Steuerungstabelle „Apply Exceptions (Ausnahmen anwenden)“ (`dmslogs.aws_dms_apply_exceptions`) unterstützt. Weitere Informationen zu Steuerungstabellen finden Sie unter [Kontrolltabellen-Aufgabeneinstellungen](#).
- AWS DMS unterstützt die Aufgabeneinstellung `TargetTablePrepMode=TRUNCATE_BEFORE_LOAD` für DynamoDB als Ziel nicht.
- AWS DMS unterstützt die Aufgabeneinstellung `TaskRecoveryTableEnabled` für DynamoDB als Ziel nicht.

Verwenden der Objektzuordnung zum Migrieren von Daten zu DynamoDB

AWS DMS nutzt Tabellenzuweisungsregeln zum Zuweisen von Daten von der Quelle zur DynamoDB-Zieltabelle. Um Daten zu einem DynamoDB-Ziel zuzuweisen, verwenden Sie eine Art von Tabellenzuordnungsregel mit der Bezeichnung `object-mapping`. Mit Objektzuweisungen können Sie Attributnamen und die zu ihnen zuzumigrierenden Daten definieren. Bei der Verwendung der Objektzuweisung müssen Sie über Auswahlregeln verfügen.

DynamoDB hat keine voreingestellte Struktur, mit Ausnahme eines Partitionsschlüssels und eines optionalen Sortierschlüssels. Wenn Sie über einen nicht zusammengesetzten Primärschlüssel verfügen, wird dieser von AWS DMS verwendet. Wenn Sie über einen zusammengesetzten Primärschlüssel verfügen oder einen Sortierschlüssel verwenden möchten, müssen Sie diese Schlüssel und die anderen Attribute in Ihrer DynamoDB-Zieltabelle definieren.

Zum Erstellen einer Objektzuweisungsregel legen Sie `rule-type` als `object-mapping` fest. Diese Regel gibt an, welchen Objektzuweisungstyp Sie verwenden möchten.

Die Struktur für die Regel lautet wie folgt:

```
{ "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 unterstützt derzeit `map-record-to-record` und `map-record-to-document` als einzige gültige Werte für den `rule-action`-Parameter. Diese Werte geben an, wie AWS DMS standardmäßig Datensätze handhabt, die als Teil der `exclude-columns`-Attributliste nicht ausgeschlossen sind. Diese Werte wirken sich in keiner Weise auf die Attributzuweisungen aus.

- Sie können `map-record-to-record` beim Migrieren aus einer relationalen Datenbank zu DynamoDB verwenden. Dabei wird der Primärschlüssel aus der relationalen Datenbank als Partitionsschlüssel in DynamoDB verwendet und für jede Spalte in der Quelldatenbank wird ein Attribut erstellt. Wenn Sie `map-record-to-record` verwenden, erstellt AWS DMS für jede Spalte in der Quelltable, die nicht in der Attributliste `exclude-columns` aufgeführt ist, ein entsprechendes Attribut auf der DynamoDB-Ziel-Instance. Dies geschieht unabhängig davon, ob diese Quellspalte in einer Attributzuweisung verwendet wird oder nicht.
- Verwenden Sie `map-record-to-document`, um Quellspalten mithilfe des Attributnamens „_doc“ in einer einzelnen, flachen DynamoDB-Zuordnung auf dem Ziel zu platzieren. Wenn Sie `map-record-to-document` verwenden, platziert AWS DMS die Daten in einem einzigen, flachen DynamoDB-Zuordnungsattribut in der Quelle. Dieses Attribut hat den Namen „_doc“. Diese Platzierung gilt für jede Spalte der Quelltable, die nicht in der `exclude-columns`-Attributliste aufgeführt ist.

Der Unterschied zwischen den `rule-action`-Parametern `map-record-to-record` und `map-record-to-document` lässt sich am besten verstehen, wenn Sie die beiden Parameter in Aktion erleben. In diesem Beispiel wird davon ausgegangen, dass Sie mit einer Tabellenzeile einer relationalen Datenbank beginnen, die die folgende Struktur aufweist und die folgenden Daten enthält:

FirstName	LastName	NickName	WorkAddress	WorkPhone	HomeAddress	HomePhone	income
▶ Daniel	Sheridan	Dan	101 Main St Cambridge, MA	800-867-5309	100 Secret St, Unknowntown, MA	123-456-7890	12345678

Um diese Informationen zu DynamoDB zu migrieren, erstellen Sie Regeln zum Zuweisen der Daten in ein DynamoDB-Tabellenelement. Beachten Sie die für den `exclude-columns`-Parameter aufgeführten Spalten. Diese Spalten werden nicht direkt dem Ziel zugewiesen. Stattdessen werden die Daten mithilfe der Attributzuweisung in neue Elemente kombiniert, z. B. werden `FirstName` und `LastName` im DynamoDB-Ziel in `CustomerName` gruppiert. `NickName` und `income` sind nicht ausgeschlossen.

```
{
  "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}"
            }
          }
        }
      }
    }
  ]
}
```

Bei Verwendung des `rule-action`-Parameters `map-record-to-record` werden die Daten für `NickName` und `income` zu den Elementen mit demselben Namen im DynamoDB-Ziel zugewiesen.

```

  ▼ Item {4}
  + 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
  + income Number : 12345678

```

Angenommen, Sie verwenden die gleichen Regeln, ändern aber den `rule-action`-Parameter in `map-record-to-document`. In diesem Fall werden die Spalten `exclude-columnsNickName` und `income`, die nicht im `-Parameter` aufgeführt sind, einem `_doc`-Element zugeordnet.

```

  ▼ Item {3}
  + 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
  + ▼ _doc Map {2}
  +   NickName String : Dan
  +   income Number : 12345678

```

Verwenden von benutzerdefinierten Bedingungsausdrücken mit Objektzuweisung

Sie können das DynamoDB-Feature für Bedingungsausdrücke verwenden, um Daten zu bearbeiten, die in eine DynamoDB-Tabelle geschrieben werden. Weitere Informationen zu Bedingungsausdrücken in DynamoDB finden Sie unter [Bedingungsausdrücke](#).

Ein Bedingungsausdruckselement besteht aus:

- einem Ausdruck (erforderlich)

- Ausdrucksattributwerten (optional). Gibt eine DynamoDB-JSON-Struktur des Attributwerts an.
- Ausdrucksattributnamen (optional)
- Optionen für die Verwendung des Bedingungsausdrucks (optional). Der Standardwert lautet "apply-during-cdc = false and apply-during-full-load = true".

Die Struktur für die Regel lautet wie folgt:

```
"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>
    }
  }
```

Im folgenden Beispiel werden die für den Bedingungsausdruck verwendeten Abschnitte hervorgehoben.



Verwenden der Attributzuweisung mit Objektzuweisung

Mit Attributzuweisungen können Sie eine Vorlagenzeichenfolge festlegen, indem Sie anhand von Quellspaltennamen Daten auf dem Ziel umstrukturieren. Es wird keine Formatierung vorgenommen, außer den Angaben, die der Benutzer für die Vorlage festlegt.

Das folgende Beispiel zeigt die Struktur der Quelldatenbank und die gewünschte Struktur des DynamoDB-Ziels. Zuerst wird die Struktur der Quelle dargestellt, in diesem Fall eine Oracle-Datenbank, und dann die gewünschte Struktur der Daten in DynamoDB. Am Ende des Beispiels ist das JSON-Format dargestellt, mit dem die gewünschte Zielstruktur erstellt wird.

Die Struktur der Oracle-Daten ist im Folgenden angegeben:

First	Last	Street	Home/SS	HomeF	WorkAddress	Work	DateOfBirth
Primärschlüssel				N/A			
Randy	Sh	5	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876541230	02/29/1988

Die Struktur der DynamoDB-Daten ist im Folgenden angegeben:

Customer Name	StoreId	ContactDetails	DateOfBirth
Partitionsschlüssel		Sortierschlüssel	
		N/A	
Randy Sh	5	<pre>{ "Name": "Randy", "Home": { "Address": "221B Baker Street", "Phone": 1234567890 }, "Work": { "Address": "31 Spooner Street, Quahog", "Phone": 9876541230 } }</pre>	02/29/1988

Im folgenden JSON-Beispiel sind die Objekt- und Spaltenzuordnungen dargestellt, mit denen die DynamoDB-Struktur erzielt wird:

```
{
  "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}\\"}}}"}
    }
]
}
]
}
}

```

Alternativ kann bei der Spaltenzuordnung das DynamoDB-Format als Dokumenttyp verwendet werden. Im folgenden Codebeispiel wird `dynamodb-map` als `attribute-sub-type` für die Attributzuweisung verwendet.

```

{
  "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}"
      }
    }
  }
}
}
```

Alternativ zu `dynamodb-map` können Sie `dynamodb-list` als Attribut-Untertyp für die Attributzuordnung verwenden, wie im folgenden Beispiel gezeigt.

```
{
  "target-attribute-name": "ContactDetailsList",
  "attribute-type": "document",
  "attribute-sub-type": "dynamodb-list",
  "value": {
    "L": [
      {
        "N": "${FirstName}"
      },
      {
        "N": "${HomeAddress}"
      },
      {
        "N": "${HomePhone}"
      },
      {
        "N": "${WorkAddress}"
      }
    ]
  }
}
```

```

    },
    {
      "N": "${WorkPhone}"
    }
  ]
}
}

```

Beispiel 1: Verwenden der Attributzuweisung mit Objektzuweisung

Im folgenden Beispiel werden Daten aus den zwei MySQL-Datenbanktabellen `nfl_data` und `sport_team` in die zwei DynamoDB-Tabellen `NFLTeams` und `SportTeams` migriert. Unten stehend ist die Tabellenstruktur und das JSON-Format angegeben, das zum Zuweisen der Daten aus den MySQL-Datenbanktabellen zu den DynamoDB-Tabellen verwendet wird.

Die Struktur der MySQL-Datenbanktabelle `nfl_data` ist im Folgenden dargestellt:

```
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    |       |
+-----+-----+-----+-----+-----+-----+
```

Die Struktur der MySQL-Datenbanktabelle `sport_team` ist im Folgenden dargestellt:


```
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    |                |
```

Im Folgenden sind die Tabellenzuordnungsregeln für die Zuordnung der zwei Tabellen zu den zwei DynamoDB-Tabellen aufgeführt:

```
{
  "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}"
      }
    ]
  }
}

```

Die Beispielausgabe für die DynamoDB-Tabelle NFLTeams ist im Folgenden dargestellt:

```

  "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"
}

```

Die Beispielausgabe für die DynamoDB-Tabelle SportsTeams ist im Folgenden dargestellt:

```

{
  "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"
}

```

Zieldatentypen für DynamoDB

Der DynamoDB-Endpunkt für AWS DMS unterstützt die meisten DynamoDB-Datentypen. In der folgenden Tabelle sind die Amazon AWS DMS-Zieldatentypen aufgeführt, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung von AWS DMS-Datentypen.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

Wenn AWS DMS Daten aus heterogenen Datenbanken migriert, weisen wir Datentypen aus der Quelldatenbank zu Zwischendatentypen zu, die als AWS DMS-Datentypen bezeichnet werden. Wir weisen dann die Zwischendatentypen zu den Zieldatentypen zu. In der folgenden Tabelle sind die jeweiligen AWS DMS-Datentypen sowie der Datentyp aufgeführt, zu dem die Zuordnung in DynamoDB erfolgt:

AWS DMS-Datentyp	DynamoDB-Datentyp
String	String
WString	String

AWS DMS-Datentyp	DynamoDB-Datentyp
Boolesch	Boolesch
Datum	String
DateTime	String
INT1	Zahl
INT2	Zahl
INT4	Zahl
INT8	Zahl
Numerischer Wert	Zahl
Real4	Zahl
Real8	Zahl
UINT1	Zahl
UINT2	Zahl
UINT4	Zahl
UINT8	Zahl
CLOB	String

Verwenden von Amazon Kinesis Data Streams als Ziel für AWS Database Migration Service

Sie können verwenden AWS DMS , um Daten in einen Amazon Kinesis Data Stream zu migrieren. Amazon-Kinesis-Datenströme sind Teil des Service Amazon Kinesis Data Streams. Sie können Kinesis-Datenströme zum Erfassen und Verarbeiten großer Ströme von Datensätzen in Echtzeit nutzen.

Ein Kinesis-Datenstrom besteht aus Shards. Shards sind eindeutig identifizierbare Sequenzen von Datensätzen in einem Stream. Weitere Informationen zu Shards in Amazon Kinesis Data Streams finden Sie unter [Shard](#) im Amazon-Kinesis-Data-Streams-Entwicklerhandbuch.

AWS Database Migration Service veröffentlicht Datensätze mithilfe von JSON in einem Kinesis-Datenstrom. Während der Konvertierung serialisiert AWS DMS jeden Datensatz aus der Quelldatenbank in ein Attributwertpaar im JSON-Format oder in ein JSON_UNFORMATTED-Nachrichtenformat. Ein JSON_UNFORMATTED-Nachrichtenformat ist eine einzeilige JSON-Zeichenfolge mit neuem Zeilenbegrenzer. Es ermöglicht Amazon Data Firehose, Kinesis-Daten an ein Amazon S3-Ziel zu liefern und sie dann mit verschiedenen Abfrage-Engines abzufragen, einschließlich Amazon Athena .

Sie verwenden die Objektzuweisung zum Migrieren Ihrer Daten von einer unterstützten Datenquelle zu einem Ziel-Stream. Mit der Objektzuweisung bestimmen Sie, wie die Datensätze im Stream zu strukturieren sind. Außerdem definieren Sie einen Partitionsschlüssel für jede Tabelle, die Kinesis Data Streams zum Gruppieren der Daten in Shards verwendet.

Wenn Tabellen auf einem Kinesis-Data-Streams-Zielendpunkt AWS DMS erstellt werden, werden so viele Tabellen wie im Quelldatenbankendpunkt erstellt. legt AWS DMS auch mehrere Parameterwerte für Kinesis Data Streams fest. Die Kosten für die Erstellung der Tabelle sind abhängig von der Datenmenge und der Anzahl der zu migrierenden Tabellen.

Note

Die Option SSL-Modus auf der AWS DMS Konsole oder API gilt nicht für einige Daten-Streaming- und NoSQL-Services wie Kinesis und DynamoDB . Sie sind standardmäßig sicher, daher AWS DMS zeigt an, dass die SSL-Moduseinstellung gleich keiner ist (SSL-Modus=Keine). Sie müssen keine zusätzliche Konfiguration für Ihren Endpunkt angeben, um SSL verwenden zu können. Wenn Sie beispielsweise Kinesis als Zielendpunkt verwenden, ist dies standardmäßig sicher. Alle API-Aufrufe an Kinesis verwenden SSL, sodass keine zusätzliche SSL-Option im AWS DMS Endpunkt erforderlich ist. Mithilfe des HTTPS-Protokolls, das AWS DMS standardmäßig verwendet, wenn eine Verbindung zu einem Kinesis-Datenstrom hergestellt wird, können Sie Daten sicher über SSL-Endpunkte einfügen und abrufen.

Kinesis-Data-Streams-Endpunkteinstellungen

Wenn Sie Zielendpunkte von Kinesis Data Streams verwenden, können Sie Transaktions- und Kontrolldetails mit der `KinesisSettings` Option in der AWS DMS API abrufen.

Sie können die Verbindungseinstellungen auf eine der folgenden Weisen festlegen:

- Verwenden Sie in der - AWS DMS Konsole Endpunkteinstellungen.
- Verwenden Sie in der CLI die `kinesis-settings` Option des [CreateEndpoint](#) Befehls .

Verwenden Sie in der CLI die folgenden Anforderungsparameter der Option `kinesis-settings`:

Note

Unterstützung für die Endpunkteinstellung `IncludeNullAndEmpty` ist in AWS DMS -Version 3.4.1 und höher verfügbar. Die Unterstützung für die anderen folgenden Endpunkteinstellungen für Kinesis-Data-Streams-Ziele ist jedoch in verfügbar AWS DMS.

- `MessageFormat` – Das Ausgabeformat für die Datensätze, die auf dem Endpunkt erstellt wurden. Das Nachrichtenformat ist `JSON` (Standard) oder `JSON_UNFORMATTED` (eine einzelne Zeile ohne Tabulator).
- `IncludeControlDetails` – Zeigt detaillierte Steuerungsinformationen für Tabellendefinition, Spaltendefinition und Tabellen- und Spaltenänderungen in der Kinesis-Nachrichtenausgabe an. Der Standardwert ist `false`.
- `IncludeNullAndEmpty` – Schließt `NULL`-Spalten und leere Spalten in das Ziel ein. Der Standardwert ist `false`.
- `IncludePartitionValue` – Zeigt den Partitionswert innerhalb der Kinesis-Nachrichtenausgabe an, es sei denn, der Partitionstyp ist `schema-table-type`. Der Standardwert ist `false`.
- `IncludeTableAlterOperations` – Enthält alle DDL-Operationen (DDL = Data Definition Language), die die Tabelle in den Steuerungsdaten ändern, wie etwa `rename-table`, `drop-table`, `add-column`, `drop-column` und `rename-column`. Der Standardwert ist `false`.
- `IncludeTransactionDetails` – Stellt detaillierte Transaktionsinformationen aus der Quelldatenbank bereit. Diese Informationen beinhalten einen Durchführungszeitstempel, eine Protokollposition sowie Werte für `transaction_id`, `previous_transaction_id` und `transaction_record_id` (den Datensatzoffset innerhalb einer Transaktion). Der Standardwert ist `false`.

- `PartitionIncludeSchemaTable` – Fügt Schema- und Tabellennamen zu Partitionswerten als Präfix hinzu, wenn der Partitionstyp `primary-key-type` ist. Dadurch wird die Datenverteilung zwischen Kinesis-Shards erhöht. Angenommen, ein SysBench-Schema hat Tausende von Tabellen und jede davon hat nur einen begrenzten Bereich für einen Primärschlüssel. In diesem Fall wird derselbe Primärschlüssel von Tausenden von Tabellen an denselben Shard gesendet, was zu einer Drosselung führt. Der Standardwert ist `false`.

Das folgende Beispiel zeigt die Option `kinesis-settings` mit einem Beispielbefehl `create-endpoint`, der über die AWS CLI ausgegeben wird.

```
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
```

Aufgabeneinstellungen für vollständige Multithread-Ladevorgänge

Um die Geschwindigkeit der Übertragung zu erhöhen, AWS DMS unterstützt ein Multithread-Vollladen in eine Ziel-Instance von Kinesis Data Streams. DMS unterstützt Multithreading u. a. mithilfe der folgenden Aufgabeneinstellungen:

- `MaxFullLoadSubTasks` – Geben Sie diese Option an, um die maximale Anzahl von Quelltabellen festzulegen, die parallel geladen werden sollen. DMS lädt jede Tabelle in die entsprechende Kinesis-Zieltabelle mithilfe einer dedizierten Unteraufgabe. Der Standardwert beträgt 8; der Maximalwert beträgt 49.
- `ParallelLoadThreads` – Verwenden Sie diese Option, um die Anzahl der Threads anzugeben, die AWS DMS verwendet, um jede Tabelle in ihre Kinesis-Zieltabelle zu laden. Der Höchstwert für ein Kinesis-Data-Streams-Ziel ist 32. Sie können eine Erhöhung dieses Höchstwerts anfordern.
- `ParallelLoadBufferSize` – Verwenden Sie diese Option, um die maximale Anzahl der Datensätze anzugeben, die in dem Puffer gespeichert werden sollen, den die parallelen Lade-Threads zum Laden von Daten in das Kinesis-Ziel verwenden. Der Standardwert lautet 50. Die maximale Wert ist 1.000. Verwenden Sie diese Einstellung mit `ParallelLoadThreads`; `ParallelLoadBufferSize` ist nur gültig, wenn es mehr als einen Thread gibt.

- `ParallelLoadQueuesPerThread` – Verwenden Sie diese Option, um die Anzahl der Warteschlangen anzugeben, auf die jeder gleichzeitige Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und eine Stapellast für das Ziel zu generieren. Der Standardwert ist 1. Für Kinesis-Ziele mit verschiedenen Nutzlastgrößen beträgt der gültige Bereich jedoch 5–512 Warteschlangen pro Thread.

Aufgabeneinstellungen für Multithreaded CDC-Ladevorgänge

Sie können die Leistung der Änderungsdatenerfassung (Change Data Capture, CDC) für Echtzeitdaten-Streaming-Zielendpunkte wie Kinesis mithilfe von Aufgabeneinstellungen verbessern, die das Verhalten des API-Aufrufs `PutRecords` ändern. Dazu können Sie die Anzahl der gleichzeitigen Threads, der Warteschlangen pro Thread und die Anzahl der Datensätze angeben, die in einem Puffer unter Verwendung von `ParallelApply*`-Aufgabeneinstellungen gespeichert werden sollen. Beispiel: Sie möchten eine CDC-Last durchführen und 128 Threads parallel anwenden. Außerdem möchten Sie auf 64 Warteschlangen pro Thread zugreifen, wobei 50 Datensätze pro Puffer gespeichert sind.

Um die CDC-Leistung hochzustufen, AWS DMS unterstützt die folgenden Aufgabeneinstellungen:

- `ParallelApplyThreads` – Gibt die Anzahl der gleichzeitigen Threads an, die während eines CDC-Ladevorgangs AWS DMS verwendet, um Datensätze an einen Kinesis-Zielendpunkt zu übertragen. Der Standardwert ist Null (0) und der maximale Wert ist 32.
- `ParallelApplyBufferSize` – Gibt die maximale Anzahl von Datensätzen an, die in jeder Pufferwarteschlange für gleichzeitige Threads gespeichert werden sollen, um sie während einer CDC-Last an einen Kinesis-Zielendpunkt zu übertragen. Der Standardwert ist 100 und der maximale Wert 1 000. Verwenden Sie diese Option, wenn `ParallelApplyThreads` mehrere Threads angibt.
- `ParallelApplyQueuesPerThread` – Gibt die Anzahl der Warteschlangen an, auf die jeder Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und während des CDC eine Stapellast für einen Kinesis-Endpunkt zu generieren. Der Standardwert ist 1 und der maximale Wert 512.

Wenn Sie `ParallelApply*`-Aufgabeneinstellungen verwenden, ist der `primary-key` der Tabelle der `partition-key-type`-Standardwert, nicht `schema-name.table-name`.

Verwenden eines Vorher-Abbilds zum Anzeigen von Originalwerten von CDC-Zeilen für einen Kinesis-Datenstrom als Ziel

Wenn Sie CDC-Aktualisierungen in ein Data-Streaming-Ziel wie Kinesis schreiben, können Sie die ursprünglichen Werte einer Quelldatenbankzeile vor der Änderung durch eine Aktualisierung anzeigen. Um dies zu ermöglichen, AWS DMS füllt ein Vorher-Image von Aktualisierungsereignissen basierend auf Daten aus, die von der Quelldatenbank-Engine bereitgestellt werden.

Verschiedene Quelldatenbank-Engines liefern unterschiedliche Mengen an Informationen für ein Vorher-Abbild:

- Oracle stellt für Spalten nur dann Aktualisierungen bereit, wenn sie sich ändern.
- PostgreSQL stellt nur Daten für Spalten bereit, die Teil des Primärschlüssels sind (geändert oder nicht). Um Daten für alle Spalten (geändert oder nicht) bereitzustellen, müssen Sie FULL statt DEFAULT für REPLICA_IDENTITY festlegen. Beachten Sie, dass Sie die Einstellung REPLICA_IDENTITY für jede Tabelle sorgfältig auswählen sollten. Wenn Sie für REPLICA_IDENTITY FULL festlegen, werden alle Spaltenwerte kontinuierlich in das Write-Ahead Logging (WAL) geschrieben. Dies kann bei Tabellen, die häufig aktualisiert werden, zu Leistungs- oder Ressourcenproblemen führen.
- MySQL stellt generell Daten für alle Spalten außer für BLOB- und CLOB-Datentypen bereit (geändert oder nicht).

Verwenden Sie entweder die BeforeImageSettings-Aufgabeneinstellung oder den add-before-image-columns-Parameter, um die Erstellung von Vorher-Abbildern zum Hinzufügen von Originalwerten aus der Quelldatenbank zur AWS DMS -Ausgabe zu aktivieren. Dieser Parameter wendet eine Spalten-Transformationsregel an.

BeforeImageSettings fügt jeder Aktualisierungsoperation ein neues JSON-Attribut mit Werten hinzu, die aus dem Quelldatenbanksystem erfasst werden, wie nachfolgend gezeigt.

```
"BeforeImageSettings": {
  "EnableBeforeImage": boolean,
  "FieldName": string,
  "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

Note

`BeforeImageSettings` Gilt nur für AWS DMS Aufgaben, die eine CDC-Komponente enthalten, z. B. Volllast- und CDC-Aufgaben (die vorhandene Daten migrieren und laufende Änderungen replizieren) oder nur für CDC-Aufgaben (die nur Datenänderungen replizieren). Wenden Sie `BeforeImageSettings` nicht auf Nur-Volllast-Aufgaben an.

Für `BeforeImageSettings`-Optionen gilt Folgendes:

- Legen Sie die `EnableBeforeImage`-Option vor dem Imaging auf `true` fest. Der Standardwert ist `false`.
- Verwenden Sie die `FieldName`-Option, um dem neuen JSON-Attribut einen Namen zuzuweisen. Wann `EnableBeforeImage` `true` ist, ist `FieldName` erforderlich und darf nicht leer sein.
- Die `ColumnFilter`-Option gibt eine Spalte an, die vor dem Imaging hinzugefügt werden soll. Wenn Sie nur Spalten hinzufügen möchten, die Teil der Primärschlüssel der Tabelle sind, verwenden Sie den Standardwert `pk-only`. Wenn Sie eine Spalte hinzufügen möchten, die einen Vorher-Abbild-Wert hat, verwenden Sie `all`. Beachten Sie, dass das Vorher-Abbild keine Spalten mit LOB-Datentypen wie CLOB oder BLOB enthält.

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```

Note

Amazon-S3-Ziele unterstützen `BeforeImageSettings` nicht. Verwenden Sie für S3-Ziele nur die `add-before-image-columns`-Transformationsregel, die vor dem Imaging während des CDC-Vorgangs ausgeführt werden soll.

Verwenden einer Vorher-Abbild-Transformationsregel

Alternativ zu den Aufgabeneinstellungen können Sie den `add-before-image-columns`-Parameter verwenden, der eine Spalten-Transformationsregel anwendet. Mit diesem Parameter können Sie das Vorher-Abbild während des CDC-Vorgangs auf Data-Streaming-Zielen wie Kinesis aktivieren.

Wenn Sie `add-before-image-columns` in einer Transformationsregel verwenden, können Sie eine feinere Steuerung der Ergebnisse für das Vorher-Abbild anwenden. Mit Transformationsregeln können Sie einen Objekt-Locator verwenden, der Ihnen die Kontrolle über die für die Regel ausgewählten Tabellen gibt. Außerdem können Sie Transformationsregeln miteinander verketteten, wodurch verschiedene Regeln auf verschiedene Tabellen angewendet werden können. Anschließend können Sie die erzeugten Spalten mithilfe anderer Regeln bearbeiten.

Note

Verwenden Sie den `add-before-image-columns`-Parameter nicht zusammen mit der `BeforeImageSettings`-Aufgabeneinstellung innerhalb derselben Aufgabe. Verwenden Sie stattdessen entweder den Parameter oder die Einstellung, aber nicht beide, für eine einzelne Aufgabe.

Ein `transformation`-Regeltyp mit dem `add-before-image-columns`-Parameter für eine Spalte muss einen `before-image-def`-Abschnitt bereitstellen. Es folgt ein Beispiel.

```
{
  "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,
  }
}
```

Der Wert von `column-prefix` wird einem Spaltennamen vorangestellt, und der Standardwert von `column-prefix` ist `BI_`. Der Wert von `column-suffix` wird an den Spaltennamen angehängt, und der Standardwert ist leer. Setzen Sie nicht `column-prefix` und `column-suffix` auf leere Zeichenfolgen.

Wählen Sie einen Wert für `column-filter`. Wenn Sie nur Spalten hinzufügen möchten, die Teil der Primärschlüssel der Tabelle sind, wählen Sie `pk-only`. Wählen Sie `non-lob`, um nur Spalten hinzuzufügen, die nicht vom LOB-Typ sind. Oder lassen Sie eine Spalte hinzufügen, die einen Vorher-Abbild-Wert hat.

Beispiel für eine Vorher-Abbild-Transformationsregel

Die Transformationsregel im folgenden Beispiel fügt eine neue Spalte mit dem Namen `BI_emp_no` auf dem Ziel hinzu. Eine Anweisung wie `UPDATE employees SET emp_no = 3 WHERE emp_no = 1;` füllt daher das `BI_emp_no` Feld mit 1. Wenn Sie CDC-Aktualisierungen für Amazon-S3-Ziele schreiben, ermöglicht die `BI_emp_no`-Spalte, zu erkennen, welche ursprüngliche Zeile aktualisiert wurde.

```
{
  "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"
      }
    }
  ]
}
```

```
}
```

Weitere Informationen zur Verwendung der `add-before-image-columns`-Regelaktion finden Sie unter [Transformationsregeln und Aktionen](#).

Voraussetzungen für die Verwendung eines Kinesis-Datenstroms als Ziel für AWS Database Migration Service

IAM-Rolle für die Verwendung eines Kinesis-Datenstroms als Ziel für AWS Database Migration Service

Bevor Sie einen Kinesis-Datenstrom als Ziel für einrichten AWS DMS, stellen Sie sicher, dass Sie eine IAM-Rolle erstellen. Diese Rolle muss es ermöglichen AWS DMS , den Zugriff auf die Kinesis-Datenströme zu übernehmen und zu gewähren, in die migriert wird. In der folgenden IAM-Richtlinie sind die Mindestzugriffsberechtigungen dargestellt.

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

Die Rolle, die Sie für die Migration zu einem Kinesis-Datenstrom verwenden, muss über die folgenden Berechtigungen verfügen.

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

```
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:PutRecord",
      "kinesis:PutRecords"
    ],
    "Resource": "arn:aws:kinesis:region:accountID:stream/streamName"
  }
]
```

Zugreifen auf einen Kinesis-Datenstrom als Ziel für AWS Database Migration Service

In AWS DMS Version 3.4.7 und höher müssen Sie einen der folgenden Schritte ausführen, um eine Verbindung zu einem Kinesis-Endpunkt herzustellen:

- Konfigurieren von DMS für die Verwendung von VPC-Endpunkten. Informationen zur Konfiguration von VPC-Endpunkten finden Sie in [Konfigurieren von VPC-Endpunkten als Quell- und Zielendpunkte in AWS DMS](#).
- Konfigurieren Sie DMS so, dass es öffentliche Routen verwendet, d. h. Ihre Replikations-Instance öffentlich zugänglich macht. Informationen zu Replikations-Instances finden Sie unter [Öffentliche und private Replikations-Instances](#).

Einschränkungen bei der Verwendung von Kinesis Data Streams als Ziel für AWS Database Migration Service

Bei der Verwendung von Kinesis Data Streams als Ziel gelten die folgenden Einschränkungen:

- AWS DMS veröffentlicht jede Aktualisierung für einen einzelnen Datensatz in der Quelldatenbank als einen Datensatz in einem bestimmten Kinesis-Datenstrom, unabhängig von Transaktionen. Sie können jedoch Transaktionsdetails für jeden Datensatz hinzufügen, indem Sie relevante Parameter der `KinesisSettings`-API verwenden.
- Der vollständige LOB-Modus wird nicht unterstützt.
- Die maximale unterstützte LOB-Größe beträgt 1 MB.
- Kinesis Data Streams unterstützt keine Deduplizierung. Anwendungen, die Daten aus einem Stream aufnehmen, müssen doppelte Datensätze verarbeiten können. Weitere Informationen finden Sie unter [Umgang mit doppelten Datensätzen](#) im Amazon-Kinesis-Data-Streams-Entwicklerhandbuch.
- AWS DMS unterstützt die folgenden beiden Formen für Partitionsschlüssel:

- `SchemaName.TableName`: Eine Kombination des Schema- und Tabellennamens.
- `${AttributeName}`: Der Wert in einem der Felder in der JSON-Datei oder der Primärschlüssel der Tabelle in der Quelldatenbank.
- Weitere Informationen zum Verschlüsseln Ihrer Daten im Ruhezustand in Kinesis Data Streams finden Sie unter [Datenschutz in Kinesis Data Streams](#) im AWS Key Management Service - Entwicklerhandbuch.
- `BatchApply` wird für einen Kinesis-Endpunkt nicht unterstützt. Die Verwendung von Batch Apply (z. B. die Zielmetadaten-Aufgabeneinstellung `BatchApplyEnabled`) für ein Kinesis-Ziel kann zu Datenverlust führen.
- Kinesis-Ziele werden nur für einen Kinesis-Datenstrom im selben AWS Konto und derselben AWS-Region wie die Replikations-Instance unterstützt.
- Bei der Migration von einer MySQL-Quelle enthalten die BeforeImage Daten keine CLOB- und BLOB-Datentypen. Weitere Informationen finden Sie unter [Verwenden eines Vorher-Abbilds zum Anzeigen von Originalwerten von CDC-Zeilen für einen Kinesis-Datenstrom als Ziel](#).
- AWS DMS unterstützt nicht die Migration von Werten des `BigInt` Datentyps mit mehr als 16 Ziffern. Um diese Einschränkung zu umgehen, können Sie die folgende Transformationsregel verwenden, um die `BigInt`-Spalte in eine Zeichenfolge zu konvertieren. Informationen zu Transformationsregeln finden Sie unter [Transformationsregeln und Aktionen](#).

```
{
  "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
  }
}
```


Verwenden der Objektzuweisung zum Migrieren von Daten zu einem Kinesis-Datenstrom

AWS DMS verwendet Tabellenzuordnungsregeln, um Daten aus der Quelle dem Kinesis-Zieldatenstrom zuzuordnen. Um Daten einem Ziel-Stream zuzuweisen, verwenden Sie eine Art von Tabellenzuweisungsregel, die als Objektzuweisung bezeichnet wird. Durch die Objektzuweisung legen Sie fest, wie Datensätze in der Quelle den im Kinesis-Datenstrom veröffentlichten Datensätzen zugewiesen werden.

Kinesis-Datenströme verfügen bis auf einen Partitionsschlüssel über keine voreingestellte Struktur. In einer Objektzuweisungsregel sind die möglichen Werte eines `partition-key-type` für Datensätze `schema-table`, `transaction-id`, `primary-key`, `constant` und `attribute-name`.

Um eine Objektzuweisungsregel zu erstellen, legen Sie `rule-type` als `object-mapping` fest. Diese Regel gibt an, welchen Objektzuweisungstyp Sie verwenden möchten.

Die Struktur für die Regel lautet wie folgt.

```
{
  "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 unterstützt derzeit `map-record-to-record` und `map-record-to-document` als einzige gültige Werte für den `-rule-action` Parameter. Diese Einstellungen wirken sich auf Werte aus, die nicht in der `exclude-columns`-Attributliste ausgeschlossen sind. Die `map-record-to-document` Werte `map-record-to-record` und geben an, wie diese Datensätze standardmäßig AWS DMS verarbeitet. Diese Werte wirken sich in keiner Weise auf die Attributzuweisungen aus.

Verwenden Sie `map-record-to-record` beim Migrieren aus einer relationalen Datenbank zu einem Kinesis-Datenstrom. Dieser Regeltyp verwendet den

`taskResourceId.schemaName.tableName`-Wert aus der relationalen Datenbank als Partitionsschlüssel im Kinesis-Datenstrom und erstellt ein Attribut für jede Spalte in der Quelldatenbank.

Beachten Sie bei Verwendung von `map-record-to-record` Folgendes:

- Diese Einstellung wirkt sich nur auf Spalten aus, die durch die `exclude-columns`-Liste ausgeschlossen wurden.
- Für jede dieser Spalten AWS DMS erstellt ein entsprechendes Attribut im Zielthema.
- AWS DMS erstellt dieses entsprechende Attribut unabhängig davon, ob die Quellspalte in einer Attributzuordnung verwendet wird.

Verwenden Sie `map-record-to-document`, um Quellspalten in ein einziges, flaches Dokument im entsprechenden Ziel-Stream unter Verwendung des Attributnamens „_doc“ einzufügen. AWS DMS platziert die Daten in einer einzigen, flachen Zuweisung in der Quelle mit dem Namen „_doc“. Diese Platzierung gilt für jede Spalte der Quelltable, die nicht in der `exclude-columns`-Attributliste aufgeführt ist.

Eine Möglichkeit, `map-record-to-record` zu verstehen, besteht darin, sich die praktische Anwendung zu veranschaulichen. In diesem Beispiel wird davon ausgegangen, dass Sie mit einer Tabellenzeile einer relationalen Datenbank beginnen, die die folgende Struktur aufweist und die folgenden Daten enthält.

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

Um diese Informationen von einem Schema mit dem Namen `Test` zu einem Kinesis-Datenstrom zu migrieren, erstellen Sie Regeln zur Zuordnung der Daten zu dem Zieldatenstrom. Die folgende Regel veranschaulicht die Zuweisung.

```
{
  "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"
  }
}
]
```

Im Folgenden ist das resultierende Datensatzformat im Kinesis-Datenstrom dargestellt:

- StreamName: XXX
- PartitionKey: Test.Customers //schmaName.tableName
- Data: //Die folgende JSON-Meldung

```
{
  "FirstName": "Randy",
  "LastName": "Marsh",
  "StoreId": "5",
  "HomeAddress": "221B Baker Street",
  "HomePhone": "1234567890",
  "WorkAddress": "31 Spooner Street, Quahog",
  "WorkPhone": "9876543210",
  "DateOfBirth": "02/29/1988"
}
```

Nehmen wir jedoch an, dass Sie dieselben Regeln verwenden, den `rule-action`-Parameter jedoch in `map-record-to-document` ändern und bestimmte Spalten ausschließen. Die folgende Regel veranschaulicht die Zuweisung.

```
{
  "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"
        ]
      }
    }
  ]
}
```

In diesem Fall werden die nicht im `exclude-columns`-Parameter aufgeführten Spalten `FirstName`, `LastName`, `StoreId` und `DateOfBirth` zu `_doc` zugeordnet. Im Folgenden ist das resultierende Datensatzformat dargestellt.

```
{
  "data":{
    "_doc":{
      "FirstName": "Randy",
      "LastName": "Marsh",
      "StoreId": "5",
      "DateOfBirth": "02/29/1988"
    }
  }
}
```

Umstrukturieren von Daten mit Attributzuweisung

Sie können die Daten während der Migration zu einem Kinesis-Datenstrom mithilfe einer Attributzuordnung umstrukturieren. So möchten Sie zum Beispiel vielleicht mehrere Felder in der Quelle in einem einzigen Feld im Ziel vereinen. Die folgenden Attributzuordnung veranschaulicht, wie die Daten umstrukturiert werden.

```
{
  "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}"
        }
      }
    }
  ]
}

```

Um einen konstanten Wert für `partition-key` festzulegen, geben Sie einen `partition-key`-Wert an. So könnten Sie auf diese Weise beispielsweise erzwingen, dass alle Daten in einem einzigen Shard gespeichert werden. Die folgende Darstellung veranschaulicht dieses Konzept.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "Test",
        "table-name": "%"
      },
      "rule-action": "include"
    },
    {
      "rule-type": "object-mapping",
      "rule-id": "1",
      "rule-name": "TransformToKinesis",
      "rule-action": "map-record-to-document",
      "object-locator": {
        "schema-name": "Test",
        "table-name": "Customer"
      },
      "mapping-parameters": {
        "partition-key": {
          "value": "ConstantPartitionKey"
        },
        "exclude-columns": [
          "FirstName",
          "LastName",
          "HomeAddress",
          "HomePhone",
          "WorkAddress",
          "WorkPhone"
        ],
        "attribute-mappings": [
          {
            "attribute-name": "CustomerName",
            "value": "${FirstName},${LastName}"
          },
          {
            "attribute-name": "ContactDetails",
            "value": {
              "Home": {
                "Address": "${HomeAddress}",

```

```

        "Phone": "${HomePhone}"
    },
    "Work": {
        "Address": "${WorkAddress}",
        "Phone": "${WorkPhone}"
    }
},
{
    "attribute-name": "DateOfBirth",
    "value": "${DateOfBirth}"
}
]
}
]
}
}

```

Note

Der `partition-key`-Wert für einen Steuerungsdatensatz, der für eine spezifische Tabelle bestimmt ist, lautet `TaskId.SchemaName.TableName`. Der `partition-key`-Wert für einen Steuerungsdatensatz, der für eine spezifische Aufgabe bestimmt ist, ist die `TaskId` des betreffenden Datensatzes. Wenn Sie einen `partition-key`-Wert in der Objektzuweisung angeben, hat dies keine Auswirkungen auf den `partition-key` für einen Steuerungsdatensatz.

Nachrichtenformat für Kinesis-Datenströme

Die JSON-Ausgabe ist einfach eine Liste von Schlüssel-Wert-Paaren. Ein `JSON_UNFORMATTED`-Nachrichtenformat ist eine einzeilige JSON-Zeichenfolge mit neuem Zeilenbegrenzer.

AWS DMS stellt die folgenden reservierten Felder bereit, um die Nutzung der Daten aus den Kinesis Data Streams zu erleichtern:

RecordType

Der Datensatztyp kann entweder für Daten oder zur Steuerung bestimmt sein. Datensätze für Datenrepräsentieren die tatsächlichen Zeilen in der Quelle. Steuerungsdatensätze sind für wichtige Ereignisse im Stream bestimmt, z. B. einen Neustart der Aufgabe.

Operation

Mögliche Operationen für Datensätze sind `load`, `insert`, `update` oder `delete`.

Mögliche Operationen für Steuerungsdatensätze sind `create-table`, `rename-table`, `drop-table`, `change-columns`, `add-column`, `drop-column`, `rename-column` oder `column-type-change`.

SchemaName

Das Quellschema für den Datensatz. Dieses Feld kann für einen Steuerungsdatensatz leer sein.

TableName

Die Quelltable für den Datensatz. Dieses Feld kann für einen Steuerungsdatensatz leer sein.

Zeitstempel

Der Zeitstempel für den Zeitpunkt, an dem die JSON-Nachricht erstellt wurde. Das Feld ist mit dem ISO-8601-Format formatiert.

Verwenden von Apache Kafka als Ziel für AWS Database Migration Service

Sie können verwenden AWS DMS , um Daten zu einem Apache-Kafka-Cluster zu migrieren. Apache Kafka ist eine verteilte Streaming-Plattform. Mit Apache Kafka können Sie Streaming-Daten in Echtzeit erfassen und verarbeiten.

AWS bietet auch Amazon Managed Streaming für Apache Kafka (Amazon MSK) zur Verwendung als AWS DMS Ziel. Amazon MSK ist ein vollständig verwalteter Apache-Kafka-Streaming-Service, der die Implementierung und Verwaltung von Apache-Kafka-Instances vereinfacht. Es funktioniert mit Open-Source-Apache-Kafka-Versionen und Sie greifen genau wie jede Apache-Kafka-Instance als AWS DMS Ziele auf Amazon-MSK-Instances zu. Weitere Informationen finden Sie unter [Was ist Amazon MSK?](#) im Entwicklerhandbuch für Amazon Managed Streaming für Apache Kafka.

Ein Kafka-Cluster speichert Streams von Datensätzen in Kategorien, die als Themen bezeichnet und in Partitionen unterteilt werden. Partitionen sind eindeutig identifizierte Sequenzen von Datensätzen (Nachrichten) in einem Thema. Partitionen können über mehrere Broker in einem Cluster verteilt werden, um die parallele Verarbeitung der Datensätze eines Themas zu ermöglichen. Weitere Informationen zu Themen und Partitionen und ihrer Verteilung in Apache Kafka finden Sie unter [Themen und Protokolle](#) und [Verteilung](#).

Bei Ihrem Kafka-Cluster kann es sich entweder um eine Amazon-MSK-Instance, einen Cluster, der auf einer Amazon-EC2-Instance ausgeführt wird, oder einen On-Premises-Cluster handeln. Eine Amazon-MSK-Instance oder ein Cluster auf einer Amazon-EC2-Instance kann sich in derselben oder einer anderen VPC befinden. Wenn es sich um einen On-Premises-Cluster handelt, können Sie Ihren eigenen On-Premises-Namensserver für Ihre Replikations-Instance verwenden, um den Host-Namen des Clusters aufzulösen. Informationen zum Einrichten eines Namensservers für Ihre Replikations-Instance finden Sie unter [Verwenden Ihres eigenen Vor-Ort-Nameservers](#). Weitere Informationen zum Einrichten eines Netzwerks finden Sie unter [Einrichten eines Netzwerks für eine Replikations-Instance](#).

Wenn Sie einen Amazon-MSK-Cluster verwenden, stellen Sie sicher, dass dessen Sicherheitsgruppe den Zugriff von Ihrer Replikations-Instance aus ermöglicht. Informationen zum Ändern der Sicherheitsgruppe für einen Amazon-MSK-Cluster finden Sie unter [Ändern der Sicherheitsgruppe eines Amazon-MSK-Clusters](#).

AWS Database Migration Service veröffentlicht Datensätze in einem Kafka-Thema mithilfe von JSON. Während der Konvertierung serialisiert AWS DMS jeden Datensatz aus der Quelldatenbank in ein Attribut/Wert-Paar im JSON-Format.

Zum Migrieren Ihrer Daten von einer unterstützten Datenquelle zu einem Kafka-Ziel-Cluster verwenden Sie die Objektzuweisung. Mit der Objektzuweisung bestimmen Sie, wie die Datensätze im Zielthema strukturiert werden sollen. Außerdem definieren Sie einen Partitionsschlüssel für jede Tabelle, die Apache Kafka zum Gruppieren der Daten in seine Partitionen verwendet.

Derzeit AWS DMS unterstützt ein einzelnes Thema pro Aufgabe. Bei einer einzelnen Aufgabe mit mehreren Tabellen beziehen sich alle Nachrichten auf ein einzelnes Thema. Jede Nachricht enthält einen Metadatenabschnitt, der das Zielschema und die AWS DMS Tabellenversionen 3.4.6 und höher identifiziert. unterstützt die Replikation mehrerer Themen mithilfe der Objektzuordnung. Weitere Informationen finden Sie unter [Replikation für mehrere Themen mithilfe der Objektzuweisung](#).

Apache Kafka-Endpunkteinstellungen

Sie können Verbindungsdetails über Endpunkteinstellungen in der AWS DMS Konsole oder die `--kafka-settings` Option in der CLI angeben. Es folgen die Anforderungen für jede Einstellung:

- `Broker` – Geben Sie die Speicherorte eines oder mehrerer Broker in Ihrem Kafka-Cluster jeweils in Form einer durch Kommata getrennten Liste mit `broker-hostname:port` an. Ein Beispiel ist `"ec2-12-345-678-901.compute-1.amazonaws.com:2345,ec2-10-987-654-321.compute-1..."`. Mit dieser Einstellung können Sie die Speicherorte einiger oder aller Broker im Cluster angeben.

Die Cluster-Broker kommunizieren miteinander, um die Partitionierung von Datensätzen zu verarbeiten, die zu dem Thema migriert wurden.

- **Topic** – (Optional) Geben Sie den Themennamen mit einer maximalen Länge von 255 Buchstaben und Symbolen an. Sie können Punkt (.), Unterstrich (_) und Minuszeichen (-) verwenden. Themennamen mit einem Punkt (.) oder einem Unterstrich (_) können in internen Datenstrukturen kollidieren. Verwenden Sie entweder eines, aber nicht beide dieser Symbole im Namen des Themas. Wenn Sie keinen Themennamen angeben, AWS DMS verwendet "kafka-default-topic" als Migrationsthema.

Note

Um entweder ein von Ihnen angegebenes Migrationsthema oder das Standardthema AWS DMS erstellen zu lassen, legen Sie `auto.create.topics.enable = true` als Teil Ihrer Kafka-Cluster-Konfiguration fest. Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von Apache Kafka als Ziel für AWS Database Migration Service](#).

- **MessageFormat** – Das Ausgabeformat für die Datensätze, die auf dem Endpunkt erstellt wurden. Das Nachrichtenformat ist JSON (Standard) oder JSON_UNFORMATTED (eine einzelne Zeile ohne Tabulator).
- **MessageMaxBytes** – Die maximale Größe in Byte für Datensätze, die auf dem Endpunkt erstellt wurden. Der Standardwert ist 1.000.000.

Note

Sie können die AWS CLI/das SDK nur verwenden, um zu einem nicht standardmäßigen Wert `MessageMaxBytes` zu wechseln. Verwenden Sie beispielsweise den folgenden Befehl, um Ihren vorhandenen Kafka-Endpunkt und `MessageMaxBytes` zu ändern.

```
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** – Stellt detaillierte Transaktionsinformationen aus der Quelldatenbank bereit. Diese Informationen beinhalten einen Durchführungszeitstempel, eine Protokollposition sowie Werte für `transaction_id`, `previous_transaction_id` und

`transaction_record_id` (den Datensatzoffset innerhalb einer Transaktion). Der Standardwert ist `false`.

- `IncludePartitionValue` – Zeigt den Partitionswert innerhalb der Kafka-Nachrichtenausgabe an, es sei denn, der Partitionstyp ist `schema-table-type`. Der Standardwert ist `false`.
- `PartitionIncludeSchemaTable` – Fügt Schema- und Tabellennamen zu Partitionswerten als Präfix hinzu, wenn der Partitionstyp `primary-key-type` ist. Dadurch wird die Datenverteilung zwischen Kafka-Partitionen erhöht. Angenommen, ein SysBench-Schema hat Tausende von Tabellen und jede davon hat nur einen begrenzten Bereich für einen Primärschlüssel. In diesem Fall wird derselbe Primärschlüssel von Tausenden von Tabellen an dieselbe Partition gesendet, was zu einer Drosselung führt. Der Standardwert ist `false`.
- `IncludeTableAlterOperations` – Enthält alle DDL-Operationen (DDL = Data Definition Language), die die Tabelle in den Steuerungsdaten ändern, wie etwa `rename-table`, `drop-table`, `add-column`, `drop-column` und `rename-column`. Der Standardwert ist `false`.
- `IncludeControlDetails` – Zeigt detaillierte Steuerungsinformationen für Tabellendefinition, Spaltendefinition und Tabellen- und Spaltenänderungen in der Kafka-Nachrichtenausgabe an. Der Standardwert ist `false`.
- `IncludeNullAndEmpty` – Schließt NULL-Spalten und leere Spalten in das Ziel ein. Der Standardwert ist `false`.
- `SecurityProtocol` – Richtet eine sichere Verbindung mit einem Kafka-Zielendpunkt mit Transport Layer Security (TLS) ein. Optionen: `ssl-authentication`, `ssl-encryption` und `sasl-ssl`. Für die Verwendung von `sasl-ssl` sind `SaslUsername` und `SaslPassword` erforderlich.
- `SslEndpointIdentificationAlgorithm` – Legt die Hostnamenüberprüfung für das Zertifikat fest. Diese Einstellung wird in AWS DMS Version 3.5.1 und höher unterstützt. Es gibt die folgenden Optionen:
 - `NONE`: Deaktivieren Sie die Hostnamenverifizierung des Brokers in der Clientverbindung.
 - `HTTPS`: Aktivieren Sie die Hostnamenüberprüfung des Brokers in der Client-Verbindung.

Sie können Einstellungen verwenden, um die Übertragungsgeschwindigkeit zu erhöhen. Dazu unterstützt AWS DMS Multi-Thread-Volllastvorgänge in einen Ziel-Cluster in Apache Kafka. AWS DMS unterstützt Multi-Threading u. a. mithilfe der folgenden Aufgabeneinstellungen:

- `MaxFullLoadSubTasks` – Verwenden Sie diese Option, um die maximale Anzahl von Quelltabellen anzugeben, die parallel geladen werden sollen. AWS DMS lädt jede Tabelle mithilfe

einer dedizierten Unteraufgabe in die entsprechende Kafka-Zieltabelle. Der Standardwert beträgt 8; der Maximalwert beträgt 49.

- `ParallelLoadThreads` – Verwenden Sie diese Option, um die Anzahl der Threads anzugeben, die AWS DMS verwendet, um jede Tabelle in ihre Kafka-Zieltabelle zu laden. Der maximale Wert für ein Apache Kafka-Ziel ist 32. Sie können eine Erhöhung dieses Höchstwerts anfordern.
- `ParallelLoadBufferSize` – Verwenden Sie diese Option, um die maximale Anzahl der Datensätze anzugeben, die in dem Puffer gespeichert werden sollen, den die parallelen Lade-Threads zum Laden von Daten in das Kafka-Ziel verwenden. Der Standardwert lautet 50. Die maximale Wert ist 1.000. Verwenden Sie diese Einstellung mit `ParallelLoadThreads`; `ParallelLoadBufferSize` ist nur gültig, wenn es mehr als einen Thread gibt.
- `ParallelLoadQueuesPerThread` – Verwenden Sie diese Option, um die Anzahl der Warteschlangen anzugeben, auf die jeder gleichzeitige Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und eine Stapellast für das Ziel zu generieren. Der Standardwert ist 1. Der Höchstwert ist 512.

Sie können die Leistung der Erfassung von Datenänderungen (Change Data Capture, CDC) für Kafka-Endpunkte verbessern, indem Sie die Aufgabeneinstellungen für parallele Threads und Massenoperationen optimieren. Dazu können Sie die Anzahl der gleichzeitigen Threads, der Warteschlangen pro Thread und die Anzahl der Datensätze angeben, die in einem Puffer unter Verwendung von `ParallelApply*`-Aufgabeneinstellungen gespeichert werden sollen. Beispiel: Sie möchten eine CDC-Last durchführen und 128 Threads parallel anwenden. Außerdem möchten Sie auf 64 Warteschlangen pro Thread zugreifen, wobei 50 Datensätze pro Puffer gespeichert sind.

Um die CDC-Leistung hochzustufen, AWS DMS unterstützt die folgenden Aufgabeneinstellungen:

- `ParallelApplyThreads` – Gibt die Anzahl der gleichzeitigen Threads an, die während eines CDC-Ladevorgangs AWS DMS verwendet, um Datensätze an einen Kafka-Zielendpunkt zu übertragen. Der Standardwert ist Null (0) und der maximale Wert ist 32.
- `ParallelApplyBufferSize` – Gibt die maximale Anzahl von Datensätzen an, die in jeder Pufferwarteschlange für gleichzeitige Threads gespeichert werden sollen, um sie während einer CDC-Last an einen Kafka-Zielendpunkt zu übertragen. Der Standardwert ist 100 und der Höchstwert 1 000. Verwenden Sie diese Option, wenn `ParallelApplyThreads` mehrere Threads angibt.
- `ParallelApplyQueuesPerThread` – Gibt die Anzahl der Warteschlangen an, auf die jeder Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und während CDC eine Stapellast für einen Kafka-Endpunkt zu generieren. Der Standardwert ist 1. Der Höchstwert ist 512.

Wenn Sie ParallelApply*-Aufgabeneinstellungen verwenden, ist der primary-key der Tabelle der partition-key-type-Standardwert, nicht schema-name.table-name.

Herstellen einer Verbindung zu Kafka mit Transport Layer Security (TLS)

Ein Kafka-Cluster akzeptiert sichere Verbindungen mit Transport Layer Security (TLS). Mit DMS können Sie eine der folgenden drei Sicherheitsprotokoll-Optionen verwenden, um eine Verbindung mit einem Kafka-Endpunkt zu sichern.

SSL-Verschlüsselung (**server-encryption**)

Clients validieren die Serveridentität anhand des Serverzertifikats. Anschließend wird eine verschlüsselte Verbindung zwischen Server und Client hergestellt.

SSL-Authentifizierung (**mutual-authentication**)

Server und Client validieren die Identität untereinander durch ihre eigenen Zertifikate. Anschließend wird eine verschlüsselte Verbindung zwischen Server und Client hergestellt.

SASL-SSL (**mutual-authentication**)

Bei der Methode Simple Authentication and Security Layer (SASL) wird das Zertifikat des Clients durch einen Benutzernamen und ein Passwort ersetzt, um die Identität eines Clients zu überprüfen. Sie geben einen Benutzernamen und ein Passwort an, die der Server registriert hat, damit der Server die Identität eines Clients überprüfen kann. Anschließend wird eine verschlüsselte Verbindung zwischen Server und Client hergestellt.

Important

Apache Kafka und Amazon MSK akzeptieren aufgelöste Zertifikate. Dies ist eine bekannte Einschränkung von Kafka und Amazon MSK, die behoben werden muss. Weitere Informationen finden Sie unter [Apache Kafka issues, KAFKA-3700](#).

Wenn Sie Amazon MSK verwenden, sollten Sie die Verwendung von Zugriffssteuerungslisten (Access Control Lists, ACLs) als Problemumgehung für diese bekannte Einschränkung in Betracht ziehen. Weitere Informationen zur Verwendung von ACLs finden Sie im Abschnitt [Apache Kafka ACLs](#) im Entwicklerhandbuch für Amazon Managed Streaming für Apache Kafka.

Wenn Sie einen selbstverwalteten Kafka-Cluster verwenden, finden Sie Informationen zur Konfiguration des Clusters im [Kommentar vom 21. Oktober 18](#).

Verwenden von SSL-Verschlüsselung mit Amazon MSK oder einem selbstverwalteten Kafka-Cluster

Sie können SSL-Verschlüsselung verwenden, um eine Endpunktverbindung mit Amazon MSK oder einem selbstverwalteten Kafka-Cluster zu sichern. Wenn Sie die SSL-Verschlüsselung als Authentifizierungsmethode verwenden, validieren Clients die Identität eines Servers anhand des Serverzertifikats. Anschließend wird eine verschlüsselte Verbindung zwischen Server und Client hergestellt.

So verwenden Sie SSL-Verschlüsselung für die Verbindung mit Amazon MSK

- Legen Sie die Sicherheitsprotokoll-Endpunkteinstellung (`SecurityProtocol`) mithilfe der Option `ssl-encryption` fest, wenn Sie Ihren Kafka-Zielendpunkt erstellen.

Im folgenden JSON-Beispiel wird das Sicherheitsprotokoll als SSL-Verschlüsselung festgelegt.

```
"KafkaSettings": {
  "SecurityProtocol": "ssl-encryption",
}
```

So verwenden Sie SSL-Verschlüsselung für einen selbstverwalteten Kafka-Cluster

1. Wenn Sie eine private Zertifizierungsstelle (Certification Authority, CA) in Ihrem On-Premises-Kafka-Cluster verwenden, laden Sie Ihr privates CA-Zertifikat hoch, um einen Amazon-Ressourcennamen (ARN) abzurufen.
2. Legen Sie die Sicherheitsprotokoll-Endpunkteinstellung (`SecurityProtocol`) mithilfe der Option `ssl-encryption` fest, wenn Sie Ihren Kafka-Zielendpunkt erstellen. Im folgenden JSON-Beispiel wird das Sicherheitsprotokoll als `ssl-encryption` festgelegt.

```
"KafkaSettings": {
  "SecurityProtocol": "ssl-encryption",
}
```

3. Wenn Sie eine private CA verwenden, legen Sie `SslCaCertificateArn` im ARN fest, den Sie im ersten Schritt abgerufen haben.

Verwenden von SSL-Authentifizierung

Sie können SSL-Authentifizierung verwenden, um eine Endpunktverbindung mit Amazon MSK oder einem selbstverwalteten Kafka-Cluster zu sichern.

Gehen Sie wie folgt vor, um die Client-Authentifizierung und -Verschlüsselung mithilfe von SSL-Authentifizierung für die Verbindung mit Amazon MSK zu aktivieren:

- Bereiten Sie einen privaten Schlüssel und ein öffentliches Zertifikat für Kafka vor.
- Laden Sie die Zertifikate in den DMS-Zertifikatmanager hoch.
- Erstellen Sie einen Kafka-Zielendpunkt mit den entsprechenden in den Kafka-Endpunkteinstellungen angegebenen Zertifikat-ARNs.

So bereiten Sie einen privaten Schlüssel und ein öffentliches Zertifikat für Amazon MSK vor

1. Erstellen Sie eine EC2-Instance und richten Sie einen Client zur Verwendung der Authentifizierung ein, wie in den Schritten 1 bis 9 im Abschnitt zur [Client-Authentifizierung](#) im Entwicklerhandbuch für Amazon Managed Streaming für Apache Kafka beschrieben.

Nachdem Sie diese Schritte abgeschlossen haben, verfügen Sie über einen Zertifikat-ARN (den in ACM gespeicherten öffentlichen Zertifikat-ARN) und einen privaten Schlüssel, der in einer Datei mit dem Namen `kafka.client.keystore.jks` enthalten ist.

2. Rufen Sie das öffentliche Zertifikat ab und kopieren Sie das Zertifikat mit dem folgenden Befehl in die Datei `signed-certificate-from-acm.pem`:

```
aws acm-pca get-certificate --certificate-authority-arn Private_CA_ARN --  
certificate-arn Certificate_ARN
```

Dieser Befehl gibt ähnliche Informationen wie im folgenden Beispiel zurück:

```
{"Certificate": "123", "CertificateChain": "456"}
```

Anschließend kopieren Sie Ihr Äquivalent von "123" in die Datei `signed-certificate-from-acm.pem`.

3. Rufen Sie den privaten Schlüssel ab, indem Sie den Schlüssel `msk-rsa` aus `kafka.client.keystore.jks` to `keystore.p12` importieren, wie im folgenden Beispiel gezeigt.


```
keytool -importkeystore \  
-srckeystore kafka.client.keystore.jks \  
-destkeystore keystore.p12 \  
-deststoretype PKCS12 \  
-srcalias msk-rsa-client \  
-deststorepass test1234 \  
-destkeypass test1234
```

4. Verwenden Sie den folgenden Befehl, um `keystore.p12` in das Format `.pem` zu exportieren.

```
openssl pkcs12 -in keystore.p12 -out encrypted-private-client-key.pem -nocerts
```

Die Meldung PEM-Passphrase eingeben wird angezeigt und gibt den Schlüssel an, der zur Verschlüsselung des Zertifikats verwendet wird.

5. Entfernen Sie Taschen- und Schlüsselattribute aus der `.pem`-Datei, um sicherzustellen, dass die erste Zeile mit der folgenden Zeichenfolge beginnt.

```
---BEGIN ENCRYPTED PRIVATE KEY---
```

So laden Sie ein öffentliches Zertifikat und einen privaten Schlüssel in den DMS-Zertifikatmanager hoch und testen die Verbindung mit Amazon MSK

1. Führen Sie den folgenden Befehl aus, um den Upload in den DMS-Zertifikatmanager auszuführen.

```
aws dms import-certificate --certificate-identifier signed-cert --certificate-pem  
file://path to signed cert  
aws dms import-certificate --certificate-identifier private-key --certificate-pem  
file://path to private key
```

2. Erstellen Sie einen Amazon-MSK-Zielendpunkt und testen Sie die Verbindung, um sicherzustellen, dass die TLS-Authentifizierung funktioniert.

```
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

Sie können SSL-Authentifizierung verwenden, um eine Verbindung mit einem selbstverwalteten Kafka-Cluster zu sichern. In Einzelfällen können Sie eine private Zertifizierungsstelle (CA) in Ihrem On-Premises-Kafka-Cluster verwenden. Laden Sie in diesem Fall Ihre CA-Kette, das öffentliche Zertifikat und den privaten Schlüssel in den DMS-Zertifikatmanager hoch. Verwenden Sie dann den entsprechenden Amazon-Ressourcennamen (ARN) in Ihren Endpunkteinstellungen, wenn Sie Ihren On-Premises-Kafka-Zielendpunkt erstellen.

So bereiten Sie einen privaten Schlüssel und ein signiertes Zertifikat für einen selbstverwalteten Kafka-Cluster vor

1. Erzeugen Sie ein Schlüsselpaar wie im folgenden Beispiel dargestellt.

```
keytool -genkey -keystore kafka.server.keystore.jks -validity 300 -storepass your-keystore-password
-keypass your-key-phrase -dname "CN=your-cn-name"
-alias alias-of-key-pair -storetype pkcs12 -keyalg RSA
```

2. Generieren Sie eine Anfrage zum Signieren des Zertifikats (Certificate Sign Request, 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. Verwenden Sie die CA in Ihrem Cluster-Truststore zum Signieren der CSR. Wenn Sie keine CA haben, können Sie Ihre eigene private CA erstellen.

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days validate-days
```

4. Importieren Sie `ca-cert` in den Truststore und Keystore des Servers. Wenn Sie keinen Truststore haben, führen Sie folgenden Befehl aus, um den Truststore zu erstellen und `ca-cert` in diesen zu importieren.

```
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. Signieren Sie das Zertifikat.

```
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. Importieren Sie das signierte Zertifikat in den 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. Verwenden Sie den folgenden Befehl, um den Schlüssel `on-premise-rsa` von `kafka.server.keystore.jks` in `keystore.p12` zu importieren.

```
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. Verwenden Sie den folgenden Befehl, um `keystore.p12` in das Format `.pem` zu exportieren.

```
openssl pkcs12 -in keystore.p12 -out encrypted-private-server-key.pem -nocerts
```

9. Laden Sie `encrypted-private-server-key.pem`, `signed-certificate.pem` und `ca-cert` in den DMS-Zertifikatmanager hoch.
10. Erstellen Sie einen Endpunkt mithilfe der zurückgegebenen ARNs.

```
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
```

Verwenden von SASL-SSL-Authentifizierung für die Verbindung mit Amazon MSK

Bei der Methode Simple Authentication and Security Layer (SASL) werden ein Benutzername und ein Passwort verwendet, um die Identität eines Clients zu überprüfen, und es wird eine verschlüsselte Verbindung zwischen Server und Client hergestellt.

Um SASL zu verwenden, erstellen Sie zunächst einen sicheren Benutzernamen und ein sicheres Passwort, wenn Sie Ihren Amazon-MSK-Cluster einrichten. Eine Beschreibung, wie sich ein sicherer Benutzername und ein sicheres Passwort für einen Amazon-MSK-Cluster einrichten lassen, finden Sie unter [Einrichtung der SASL/SCRAM-Authentifizierung für einen Amazon-MSK-Cluster](#) im Entwicklerhandbuch für Amazon Managed Streaming für Apache Kafka.

Wenn Sie dann Ihren Kafka-Zielendpunkt erstellen, legen Sie die Sicherheitsprotokoll-Endpunkteinstellung (`SecurityProtocol`) mithilfe der Option `sasl-ssl` fest. Legen Sie auch die Optionen `SaslUsername` und `SaslPassword` fest. Stellen Sie sicher, dass diese mit dem sicheren Benutzernamen und Passwort übereinstimmen, die Sie bei der erstmaligen Einrichtung Ihres Amazon-MSK-Clusters erstellt haben, wie im folgenden JSON-Beispiel gezeigt.

```
"KafkaSettings": {  
  "SecurityProtocol": "sasl-ssl",  
  "SaslUsername": "Amazon MSK cluster secure user name",  
  "SaslPassword": "Amazon MSK cluster secure password"  
}
```

Note

- Derzeit AWS DMS unterstützt nur öffentliche CA-gestützte SASL-SSL. DMS unterstützt SASL-SSL nicht für die Verwendung mit selbstverwaltetem Kafka, das von einer privaten CA unterstützt wird.
- Für die SASL-SSL-Authentifizierung AWS DMS unterstützt standardmäßig den SCRAM-SHA-512-Mechanismus. - AWS DMS Versionen 3.5.0 und höher unterstützen auch den Plain-Mechanismus. Setzen Sie den Parameter `SaslMechanism` des API-Datentyps `KafkaSettings` auf `PLAIN`, um den Plain-Mechanismus zu unterstützen.

Verwenden eines Vorher-Abbilds zum Anzeigen von Originalwerten von CDC-Zeilen für Apache Kafka als Ziel

Beim Schreiben von CDC-Aktualisierungen auf ein Data-Streaming-Ziel wie Kafka können Sie die ursprünglichen Werte einer Quelldatenbankzeile anzeigen, bevor sie durch eine Aktualisierung geändert werden. Damit dies möglich ist, AWS DMS füllt ein Vorher-Image von Aktualisierungsereignissen basierend auf Daten aus, die von der Quelldatenbank-Engine bereitgestellt werden.

Verschiedene Quelldatenbank-Engines liefern unterschiedliche Mengen an Informationen für ein Vorher-Abbild:

- Oracle stellt für Spalten nur dann Aktualisierungen bereit, wenn sie sich ändern.
- PostgreSQL stellt nur Daten für Spalten bereit, die Teil des Primärschlüssels sind (geändert oder nicht). Wenn die logische Replikation verwendet wird und `REPLICA IDENTITY FULL` für die Quelltable festgelegt ist, können Sie vollständige Vorher-Nachher-Informationen zu der Zeile abrufen, die in die WALs geschrieben wurde und hier verfügbar ist.
- MySQL stellt generell Daten für alle Spalten (geändert oder nicht) bereit.

Verwenden Sie entweder die `BeforeImageSettings`-Aufgabeneinstellung oder den `add-before-image-columns`-Parameter, um die Erstellung von Vorher-Abbildern zum Hinzufügen von Originalwerten aus der Quelldatenbank zur AWS DMS -Ausgabe zu aktivieren. Dieser Parameter wendet eine Spalten-Transformationsregel an.

`BeforeImageSettings` fügt jeder Aktualisierungsoperation ein neues JSON-Attribut mit Werten hinzu, die aus dem Quelldatenbanksystem erfasst werden, wie nachfolgend gezeigt.

```
"BeforeImageSettings": {
  "EnableBeforeImage": boolean,
  "FieldName": string,
  "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

Note

Wenden Sie `BeforeImageSettings` auf Volllast plus CDC-Aufgaben (die vorhandene Daten migrieren und laufende Änderungen replizieren) oder nur auf CDC-Aufgaben (die nur Datenänderungen replizieren) an. Wenden Sie `BeforeImageSettings` nicht auf Nur-Volllast-Aufgaben an.

Für `BeforeImageSettings`-Optionen gilt Folgendes:

- Legen Sie die `EnableBeforeImage`-Option vor dem Imaging auf `true` fest. Der Standardwert ist `false`.
- Verwenden Sie die `FieldName`-Option, um dem neuen JSON-Attribut einen Namen zuzuweisen. Wann `EnableBeforeImage` `true` ist, ist `FieldName` erforderlich und darf nicht leer sein.
- Die `ColumnFilter`-Option gibt eine Spalte an, die vor dem Imaging hinzugefügt werden soll. Wenn Sie nur Spalten hinzufügen möchten, die Teil der Primärschlüssel der Tabelle sind, verwenden Sie den Standardwert `pk-only`. Wenn Sie nur Spalten hinzufügen möchten, die nicht vom LOB-Typ sind, verwenden Sie `non-lob`. Wenn Sie eine Spalte hinzufügen möchten, die einen Vorher-Abbild-Wert hat, verwenden Sie `all`.

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
```

```
"FieldName": "before-image",
"ColumnFilter": "pk-only"
}
```

Verwenden einer Vorher-Abbild-Transformationsregel

Alternativ zu den Aufgabeneinstellungen können Sie den `add-before-image-columns`-Parameter verwenden, der eine Spalten-Transformationsregel anwendet. Mit diesem Parameter können Sie das Vorher-Imaging während des CDC-Vorgangs auf Data-Streaming-Zielen wie Kafka aktivieren.

Wenn Sie `add-before-image-columns` in einer Transformationsregel verwenden, können Sie eine feinere Steuerung der Ergebnisse für das Vorher-Abbild anwenden. Mit Transformationsregeln können Sie einen Objekt-Locator verwenden, der Ihnen die Kontrolle über die für die Regel ausgewählten Tabellen gibt. Außerdem können Sie Transformationsregeln miteinander verketteten, wodurch verschiedene Regeln auf verschiedene Tabellen angewendet werden können. Anschließend können Sie die erzeugten Spalten mithilfe anderer Regeln bearbeiten.

Note

Verwenden Sie den `add-before-image-columns`-Parameter nicht zusammen mit der `BeforeImageSettings`-Aufgabeneinstellung innerhalb derselben Aufgabe. Verwenden Sie stattdessen entweder den Parameter oder die Einstellung, aber nicht beide, für eine einzelne Aufgabe.

Ein `transformation`-Regeltyp mit dem `add-before-image-columns`-Parameter für eine Spalte muss einen `before-image-def`-Abschnitt bereitstellen. Es folgt ein Beispiel.

```
{
  "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,
  }
}
```

Der Wert von `column-prefix` wird einem Spaltennamen vorangestellt, und der Standardwert von `column-prefix` ist `BI_`. Der Wert von `column-suffix` wird an den Spaltennamen angehängt, und der Standardwert ist leer. Setzen Sie nicht `column-prefix` und `column-suffix` auf leere Zeichenfolgen.

Wählen Sie einen Wert für `column-filter`. Wenn Sie nur Spalten hinzufügen möchten, die Teil der Primärschlüssel der Tabelle sind, wählen Sie `pk-only`. Wählen Sie `non-lob`, um nur Spalten hinzuzufügen, die nicht vom LOB-Typ sind. Oder lassen Sie eine Spalte hinzufügen, die einen Vorher-Abbild-Wert hat.

Beispiel für eine Vorher-Abbild-Transformationsregel

Die Transformationsregel im folgenden Beispiel fügt eine neue Spalte mit dem Namen `BI_emp_no` auf dem Ziel hinzu. Eine Anweisung wie `UPDATE employees SET emp_no = 3 WHERE emp_no = 1;` füllt daher das `BI_emp_no` Feld mit 1. Wenn Sie CDC-Aktualisierungen für Amazon-S3-Ziele schreiben, ermöglicht die `BI_emp_no`-Spalte, zu erkennen, welche ursprüngliche Zeile aktualisiert wurde.

```
{
  "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"
    }
}
]
```

Weitere Informationen zur Verwendung der `add-before-image-columns`-Regelaktion finden Sie unter [Transformationsregeln und Aktionen](#).

Einschränkungen bei der Verwendung von Apache Kafka als Ziel für AWS Database Migration Service

Bei der Verwendung von Apache Kafka als Ziel gelten die folgenden Einschränkungen:

- AWS DMS Kafka-Zielendpunkte unterstützen keine IAM-Zugriffskontrolle für Amazon Managed Streaming for Apache Kafka (Amazon MSK).
- Der vollständige LOB-Modus wird nicht unterstützt.
- Geben Sie eine Kafka-Konfigurationsdatei für Ihren Cluster mit Eigenschaften an, die es ermöglichen, automatisch neue Themen AWS DMS zu erstellen. Schließen Sie die Einstellung `auto.create.topics.enable = true` ein. Wenn Sie Amazon MSK verwenden, können Sie beim Erstellen Ihres Kafka-Clusters die Standardkonfiguration angeben und dann die Einstellung `auto.create.topics.enable` in `true` ändern. Weitere Informationen zu den Standardkonfigurationseinstellungen finden Sie unter [Die Standardkonfiguration von Amazon MSK](#) im Entwicklerhandbuch für Amazon Managed Streaming für Apache Kafka. Wenn Sie einen vorhandenen Kafka-Cluster ändern müssen, der mit Amazon MSK erstellt wurde, führen Sie den AWS CLI Befehl aus, `aws kafka create-configuration` um Ihre Kafka-Konfiguration zu aktualisieren, wie im folgenden Beispiel:

```
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"
```

```
}
```

Hier ist `//~/kafka_configuration` die Konfigurationsdatei, die Sie mit den erforderlichen Eigenschaftseinstellungen erstellt haben.

Wenn Sie Ihre eigene Kafka-Instance verwenden, die auf Amazon EC2 installiert ist, ändern Sie die Konfiguration des Kafka-Clusters mit der `-auto.create.topics.enable = true` Einstellung, damit AWS DMS automatisch neue Themen erstellen kann, indem Sie die Optionen verwenden, die mit Ihrer Instance bereitgestellt werden.

- AWS DMS veröffentlicht jede Aktualisierung für einen einzelnen Datensatz in der Quelldatenbank als einen Datensatz (Nachricht) in einem bestimmten Kafka-Thema, unabhängig von Transaktionen.
- AWS DMS unterstützt die folgenden zwei Formen für Partitionsschlüssel:
 - `SchemaName.TableName`: Eine Kombination des Schema- und Tabellennamens.
 - `${AttributeName}`: Der Wert in einem der Felder in der JSON-Datei oder der Primärschlüssel der Tabelle in der Quelldatenbank.
- `BatchApply` wird für einen Kafka-Endpunkt nicht unterstützt. Bei Verwendung von `Batch Apply` (z. B. der Zielmetadaten-Aufgabeneinstellung `BatchApplyEnabled`) für ein Kafka-Ziel kann es zu einem Datenverlust kommen.
- AWS DMS unterstützt nicht die Migration von Werten des `BigInt` Datentyps mit mehr als 16 Ziffern. Um diese Einschränkung zu umgehen, können Sie die folgende Transformationsregel verwenden, um die `BigInt`-Spalte in eine Zeichenfolge zu konvertieren. Informationen zu Transformationsregeln finden Sie unter [Transformationsregeln und Aktionen](#).

```
{
  "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
  }
}
```

```
}  
}
```

Verwenden der Objektzuweisung zum Migrieren von Daten zu einem Kafka-Thema

AWS DMS verwendet Tabellenzuordnungsregeln, um Daten aus der Quelle dem Ziel-Kafka-Thema zuzuordnen. Um Daten einem Zielthema zuzuweisen, verwenden Sie eine Art von Tabellenzuweisungsregel, die als Objektzuweisung bezeichnet wird. Mit der Objektzuweisung legen Sie fest, wie Datensätze in der Quelle den in einem Kafka-Thema veröffentlichten Datensätzen zugewiesen werden.

Kafka-Themen verfügen bis auf einen Partitionsschlüssel über keine voreingestellte Struktur.

Note

Sie müssen die Objektzuweisung nicht verwenden. Sie können die reguläre Tabellenzuweisung für verschiedene Transformationen verwenden. Der Partitionsschlüsseltyp folgt jedoch diesen Standardverhaltensweisen:

- Der Primärschlüssel wird als Partitionsschlüssel für Volllast verwendet.
- Wenn keine Aufgabeneinstellungen für parallele Anwendung verwendet werden, wird `schema.table` als Partitionsschlüssel für CDC verwendet.
- Wenn Aufgabeneinstellungen für parallele Anwendung verwendet werden, wird der Primärschlüssel als Partitionsschlüssel für CDC verwendet.

Um eine Objektzuweisungsregel zu erstellen, legen Sie `rule-type` als `object-mapping` fest. Diese Regel gibt an, welchen Objektzuweisungstyp Sie verwenden möchten.

Die Struktur für die Regel lautet wie folgt.

```
{  
  "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 unterstützt derzeit `map-record-to-record` und `map-record-to-document` als einzige gültige Werte für den `-rule-action` Parameter. Diese Einstellungen wirken sich auf Werte aus, die nicht in der `exclude-columns`-Attributliste ausgeschlossen sind. Die `map-record-to-document` Werte `map-record-to-record` und geben an, wie diese Datensätze standardmäßig AWS DMS verarbeitet. Diese Werte wirken sich in keiner Weise auf die Attributzuweisungen aus.

Verwenden Sie `map-record-to-record` beim Migrieren aus einer relationalen Datenbank zu einem Kafka-Thema. Dieser Regeltyp verwendet den `taskResourceId.schemaName.tableName`-Wert aus der relationalen Datenbank als Partitionsschlüssel in dem Kafka-Thema und erstellt ein Attribut für jede Spalte in der Quelldatenbank.

Beachten Sie bei Verwendung von `map-record-to-record` Folgendes:

- Diese Einstellung wirkt sich nur auf Spalten aus, die durch die `exclude-columns`-Liste ausgeschlossen wurden.
- Für jede solche Spalte AWS DMS erstellt ein entsprechendes Attribut im Zielthema.
- AWS DMS erstellt dieses entsprechende Attribut unabhängig davon, ob die Quellspalte in einer Attributzuordnung verwendet wird.

Eine Möglichkeit, `map-record-to-record` zu verstehen, besteht darin, sich die praktische Anwendung zu veranschaulichen. In diesem Beispiel wird davon ausgegangen, dass Sie mit einer Tabellenzeile einer relationalen Datenbank beginnen, die die folgende Struktur aufweist und die folgenden Daten enthält.

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		

Um diese Informationen von einem Schema mit dem Namen Test zu einem Kafka-Thema zu migrieren, erstellen Sie Regeln, um die Daten dem Zielthema zuzuweisen. Die folgende Regel veranschaulicht die Zuweisung.

```
{
  "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"
      }
    }
  ]
}
```

Nachfolgend wird das resultierende Datensatzformat bei Verwendung unserer Beispieldaten in dem Kafka-Zielthema für ein bestimmtes Kafka-Thema und einen bestimmten Partitionsschlüssel (in diesem Fall `taskResourceId.schemaName.tableName`) illustriert.

```
{
```

```
"FirstName": "Randy",
"LastName": "Marsh",
"StoreId": "5",
"HomeAddress": "221B Baker Street",
"HomePhone": "1234567890",
"WorkAddress": "31 Spooner Street, Quahog",
"WorkPhone": "9876543210",
"DateOfBirth": "02/29/1988"
}
```

Themen

- [Umstrukturieren von Daten mit Attributzuweisung](#)
- [Replikation für mehrere Themen mithilfe der Objektzuweisung](#)
- [Nachrichtenformat für Apache Kafka](#)

Umstrukturieren von Daten mit Attributzuweisung

Sie können die Daten umstrukturieren, während Sie sie mittels einer Attributzuweisung zu einem Kafka-Thema migrieren. So möchten Sie zum Beispiel vielleicht mehrere Felder in der Quelle in einem einzigen Feld im Ziel vereinen. Die folgenden Attributzuordnung veranschaulicht, wie die Daten umstrukturiert werden.

```
{
  "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}"
                    }
                }
            }
        ]
    }
}

```

Um einen konstanten Wert für `partition-key` festzulegen, geben Sie einen `partition-key`-Wert an. So könnten Sie auf diese Weise beispielsweise erzwingen, dass alle Daten in einer einzigen Partition gespeichert werden. Die folgende Darstellung veranschaulicht dieses Konzept.

```
{
  "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

Der `partition-key`-Wert für einen Steuerungsdatensatz, der für eine spezifische Tabelle bestimmt ist, lautet `TaskId.SchemaName.TableName`. Der `partition-key`-Wert für einen Steuerungsdatensatz, der für eine spezifische Aufgabe bestimmt ist, ist die `TaskId` des betreffenden Datensatzes. Wenn Sie einen `partition-key`-Wert in der Objektzuweisung angeben, hat dies keine Auswirkungen auf den `partition-key` für einen Steuerungsdatensatz.

Replikation für mehrere Themen mithilfe der Objektzuweisung

Standardmäßig migrieren AWS DMS Aufgaben alle Quelldaten zu einem der folgenden Kafka-Themen:

- Wie im Feld `Thema` des AWS DMS Zielendpunkts angegeben.
- Wie von `kafka-default-topic` angegeben, wenn das Feld `Thema` des Zielendpunkts nicht gefüllt ist und die Kafka-Einstellung `auto.create.topics.enable` auf `true` gesetzt ist.

Mit den AWS DMS Engine-Versionen 3.4.6 und höher können Sie das `-kafka-target-topic` Attribut verwenden, um jede migrierte Quelltablelle einem separaten Thema zuzuordnen. Mit den folgenden Objektzuweisungsregeln werden beispielsweise die Quelltabellen `Customer` und `Address` zu den Kafka-Themen `customer_topic` bzw. `address_topic` migriert. Gleichzeitig AWS DMS migriert alle anderen Quelltabellen, einschließlich der `Bills` Tabelle im `Test` Schema, zu dem im Zielendpunkt angegebenen Thema.

```
{
  "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"
  }
}
```

Mithilfe der Kafka-Replikation für mehrere Themen können Sie Quelltabellen mit einer einzigen Replikationsaufgabe gruppieren und zu separaten Kafka-Themen migrieren.

Nachrichtenformat für Apache Kafka

Die JSON-Ausgabe ist einfach eine Liste von Schlüssel-Wert-Paaren.

RecordType

Der Datensatztyp kann entweder für Daten oder zur Steuerung bestimmt sein. Datensätze für Datenrepräsentieren die tatsächlichen Zeilen in der Quelle. Steuerungsdatensätze sind für wichtige Ereignisse im Stream bestimmt, z. B. einen Neustart der Aufgabe.

Operation

Mögliche Operationen für Datensätze sind `load`, `insert`, `update` oder `delete`.

Mögliche Operationen für Steuerungsdatensätze sind `create-table`, `rename-table`, `drop-table`, `change-columns`, `add-column`, `drop-column`, `rename-column` oder `column-type-change`.

SchemaName

Das Quellschema für den Datensatz. Dieses Feld kann für einen Steuerungsdatensatz leer sein.

TableName

Die Quelltable für den Datensatz. Dieses Feld kann für einen Steuerungsdatensatz leer sein.

Zeitstempel

Der Zeitstempel für den Zeitpunkt, an dem die JSON-Nachricht erstellt wurde. Das Feld ist mit dem ISO-8601-Format formatiert.

Das folgende Beispiel für eine JSON-Nachricht veranschaulicht eine Datentyp-Nachricht mit allen zusätzlichen Metadaten.

```
{
  "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"
  }
}
```

Das folgende Beispiel für eine JSON-Nachricht veranschaulicht eine Steuerungstyp-Nachricht.

```
{
  "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":"sctest",
    "table-name":"sctest_t1"
}
}
```

Verwenden eines Clusters in Amazon OpenSearch Service als Ziel für AWS Database Migration Service

Mit AWS DMS können Sie Daten zu Amazon OpenSearch Service (OpenSearch Service) migrieren. OpenSearch Service ist ein verwalteter Service zum einfachen Bereitstellen, Betreiben und Skalieren von OpenSearch-Service-Clustern.

In OpenSearch Service arbeiten Sie mit Indizes und Dokumenten. Ein Index ist eine Sammlung von Dokumenten; ein Dokument ist ein JSON-Objekt, das Skalarwerte, Arrays und andere Objekte enthält. OpenSearch stellt eine JSON-basierte Abfragesprache bereit, sodass Sie Daten in einem Index abfragen und die entsprechenden Dokumente abrufen können.

Wenn AWS DMS Indizes für einen Zielendpunkt für OpenSearch Service erstellt, wird ein Index für jede Tabelle vom Quellendpunkt erstellt. Die Kosten für das Erstellen eines OpenSearch-Service-Index hängen von mehreren Faktoren ab. Dazu gehören die Anzahl der erstellten Indizes, die Gesamtmenge von Daten in diesen Indizes und die geringe Menge von Metadaten, die OpenSearch für jedes Dokument speichert.

Konfigurieren Sie Ihren OpenSearch-Service-Cluster mit Rechen- und Speicherressourcen, die für den Umfang der Migration geeignet sind. Wir empfehlen, dass Sie abhängig von der Replikationsaufgabe, die Sie verwenden möchten, die folgenden Faktoren berücksichtigen:

- Erwägen Sie für einen vollständigen Datenladevorgang die Gesamtmenge der Daten, die Sie migrieren möchten, sowie auch die Übertragungsgeschwindigkeit.
- Berücksichtigen Sie für die Replikation fortlaufender Änderungen die Häufigkeit der Aktualisierungen und Ihre Anforderungen für eine End-to-End-Latenz.

Konfigurieren Sie außerdem die Index-Einstellungen auf Ihrem OpenSearch-Cluster und achten Sie dabei genau auf die Dokumentenanzahl.

Aufgabeneinstellungen für vollständige Multithread-Ladevorgänge

Um die Übertragungsgeschwindigkeit zu erhöhen, unterstützt AWS DMS Multi-Thread-Volllastvorgänge in einen Ziel-Cluster in OpenSearch Service. AWS DMS unterstützt Multi-Threading u. a. mithilfe der folgenden Aufgabeneinstellungen:

- `MaxFullLoadSubTasks` – Geben Sie diese Option an, um die maximale Anzahl von Quelltabellen festzulegen, die parallel geladen werden sollen. DMS lädt jede Tabelle mithilfe einer dedizierten Unteraufgabe in den entsprechende OpenSearch-Service-Zielindex. Der Standardwert beträgt 8; der Maximalwert beträgt 49.
- `ParallelLoadThreads` – Verwenden Sie diese Option, um die Anzahl der Threads anzugeben, die AWS DMS zum Laden der einzelnen Tabellen in den OpenSearch-Service-Zielindex verwendet. Der Höchstwert für ein OpenSearch-Service-Ziel ist 32. Sie können eine Erhöhung dieses Höchstwerts anfordern.

Note

Wenn Sie am Standardwert für `ParallelLoadThreads` (0) keine Änderung vornehmen, überträgt AWS DMS jeweils immer nur einen Datensatz. Durch diesen Ansatz wird der

OpenSearch-Service-Cluster übermäßig belastet. Überprüfen Sie, dass für diese Option ein Wert 1 oder mehr angegeben wurde.

- `ParallelLoadBufferSize` – Verwenden Sie diese Option, um die maximale Anzahl der Datensätze anzugeben, die in dem Puffer gespeichert werden sollen, den die parallelen Lade-Threads zum Laden von Daten in das OpenSearch-Service-Ziel verwenden. Der Standardwert lautet 50. Die maximale Wert ist 1.000. Verwenden Sie diese Einstellung mit `ParallelLoadThreads`; `ParallelLoadBufferSize` ist nur gültig, wenn es mehr als einen Thread gibt.

Weitere Informationen dazu, wie DMS einen OpenSearch-Service-Cluster mithilfe von Multi-Threading lädt, finden Sie im AWS-Blogbeitrag [Scale Amazon OpenSearch Service for AWS Database Migration Service migrations](#).

Aufgabeneinstellungen für Multithreaded CDC-Ladevorgänge

Sie können die Leistung der Erfassung von Datenänderungen (Change data Capture, CDC) für einen Ziel-Cluster in OpenSearch Service mithilfe von Aufgabeneinstellungen verbessern, die das Verhalten des API-Aufrufs `PutRecords` ändern. Dazu können Sie die Anzahl der gleichzeitigen Threads, der Warteschlangen pro Thread und die Anzahl der Datensätze angeben, die in einem Puffer unter Verwendung von `ParallelApply*`-Aufgabeneinstellungen gespeichert werden sollen. Beispiel: Sie möchten eine CDC-Last ausführen und 32 Threads parallel anwenden. Außerdem möchten Sie auf 64 Warteschlangen pro Thread zugreifen, wobei 50 Datensätze pro Puffer gespeichert sind.

Note

Unterstützung für die Verwendung von `ParallelApply*`-Aufgabeneinstellungen während CDC-Vorgängen zu Zielendpunkten in Amazon OpenSearch Service ist in AWS DMS Version 3.4.0 und höher verfügbar.

Zur Steigerung der CDC-Leistung werden folgende Aufgabeneinstellungen von AWS DMS unterstützt:

- `ParallelApplyThreads` – Gibt die Anzahl der gleichzeitigen Threads an, die AWS DMS während einer CDC-Last verwendet, um Datensätze an einen OpenSearch-Service-Zielendpunkt zu übertragen. Der Standardwert ist Null (0) und der maximale Wert ist 32.

- `ParallelApplyBufferSize` – Gibt die maximale Anzahl von Datensätzen an, die in jeder Pufferwarteschlange für gleichzeitige Threads gespeichert werden sollen, um sie während einer CDC-Last an einen OpenSearch-Service-Zielendpunkt zu übertragen. Der Standardwert ist 100 und der Höchstwert 1 000. Verwenden Sie diese Option, wenn `ParallelApplyThreads` mehrere Threads angibt.
- `ParallelApplyQueuesPerThread` – Gibt die Anzahl der Warteschlangen an, auf die jeder Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und während CDC eine Stapellast für einen OpenSearch-Service-Endpunkt zu generieren.

Wenn Sie `ParallelApply*`-Aufgabeneinstellungen verwenden, ist der `primary-key` der Tabelle der `partition-key-type`-Standardwert, nicht `schema-name.table-name`.

Migrieren aus einer Tabelle in einer relationalen Datenbank zu einem OpenSearch-Service-Index

AWS DMS unterstützt die Migration von Daten in die skalaren OpenSearch-Service-Datentypen. Bei der Migration von Daten aus einer relationalen Datenbank wie Oracle oder MySQL zu OpenSearch Service möchten Sie möglicherweise die Art und Weise umstrukturieren, wie diese Daten gespeichert werden.

AWS DMS unterstützt die folgenden skalaren OpenSearch-Service-Datentypen:

- Boolesch
- Datum
- Gleitkommazahl
- Int
- Zeichenfolge

AWS DMS wandelt Daten vom Type "Datum" in den Typ "Zeichenfolge" um. Sie können zum Auslegen dieser Datumsangaben eine benutzerdefinierte Zuweisung angeben.

AWS DMS unterstützt keine Migration von LOB-Datentypen.

Voraussetzungen für die Verwendung von Amazon OpenSearch Service als Ziel für AWS Database Migration Service

Bevor Sie eine OpenSearch-Service-Datenbank als Ziel für AWS DMS einsetzen können, müssen Sie eine AWS Identity and Access Management (IAM)-Rolle erstellen. Diese Rolle sollte AWS DMS den Zugriff auf die OpenSearch-Service-Indizes am Zielendpunkt gewähren. In der folgenden IAM-Richtlinie sind die Mindestzugriffsberechtigungen dargestellt.

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

Die Rolle, die Sie für die Migration zu OpenSearch Service verwenden, muss die folgenden Berechtigungen aufweisen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpDelete",
        "es:ESHttpGet",
        "es:ESHttpHead",
        "es:ESHttpPost",
        "es:ESHttpPut"
      ],
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

```
]
}
```

Ersetzen Sie im vorherigen Beispiel *region* durch die ID der AWS-Region, *account-id* durch Ihre AWS-Konto-ID und *domain-name* durch den Namen Ihrer Domain von Amazon OpenSearch Service. Ein Beispiel ist `arn:aws:es:us-west-2:123456789012:domain/my-es-domain`.

Endpunkteinstellungen bei Verwendung von OpenSearch Service als Ziel für AWS DMS

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer OpenSearch-Service-Zieldatenbank verwenden. Sie legen die Einstellungen fest, wenn Sie den Zielendpunkt mithilfe der AWS DMS-Konsole erstellen oder indem Sie den Befehl `create-endpoint` in der [AWS CLI](#) aufrufen. Verwenden Sie die JSON-Syntax `--elasticsearch-settings '{"EndpointSetting": "value", ...}'`.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit OpenSearch Service als Ziel verwenden können.

Attributname	Zulässige Werte	Standardwert und Beschreibung
<code>FullLoadErrorPercentage</code>	Eine positive Ganzzahl größer als 0, aber nicht größer als 100.	10 – Bei einem vollständigen Ladevorgang bestimmt dieses Attribut den Schwellenwert von Fehlern, die zulässig sind, bevor die Aufgabe fehlschlägt. Nehmen wir beispielsweise an, dass 1.500 Zeilen am Quellendpunkt vorhanden sind und dieser Parameter auf 10 eingestellt ist. Die Aufgabe schlägt dann fehl, wenn AWS DMS mehr als 150 Fehler (10 Prozent der Zeilenanzahl) beim Schreiben zum Zielendpunkt antrifft.
<code>ErrorRetryDuration</code>	Eine positive Ganzzahl größer als 0.	300 – Wenn am Zielendpunkt ein Fehler auftritt, wiederholt AWS DMS den Vorgang für die angegebene Anzahl von Sekunden. Andernfalls schlägt die Aufgabe fehl.

Einschränkungen bei Verwendung von Amazon OpenSearch Service als Ziel für AWS Database Migration Service

Die folgenden Einschränkungen gelten bei Verwendung von Amazon OpenSearch Service als Ziel:

- OpenSearch Service nutzt die dynamische Zuordnung (einen automatischen Schätzwert), um die zu verwendenden Datentypen für migrierte Daten zu bestimmen.
- OpenSearch Service speichert jedes Dokument mit einer eindeutigen ID. Es folgt ein Beispiel für eine solche ID.

```
"_id": "D359F8B537F1888BC71FE20B3D79EAE6674BE7ACA9B645B0279C7015F6FF19FD"
```

Jede Dokument-ID ist 64 Bytes lang. Dies kann daher als Speicheranforderung vorausgesetzt werden. Wenn Sie zum Beispiel 100 000 Zeilen von einer AWS DMS-Quelle migrieren, erfordert der resultierende OpenSearch-Service-Index Speicher für zusätzliche 6 400 000 Bytes.

- OpenSearch Service lässt keine Aktualisierungen der Primärschlüsselattribute zu. Diese Einschränkung ist wichtig, wenn Sie die laufende Replikation mit Change Data Capture (CDC) verwenden, da sie zu unerwünschten Daten im Ziel führen kann. In CDC-Modus werden Primärschlüssel SHA256-Werten zugeordnet, die 32 Bytes lang sind. Diese werden in visuell lesbare 64-Byte-Zeichenfolgen konvertiert und als Dokument-IDs in OpenSearch Service verwendet.
- Wenn AWS DMS auf Elemente trifft, die nicht migriert werden können, werden Fehlermeldungen in Amazon CloudWatch Logs geschrieben. Dieses Verhalten unterscheidet sich von anderen AWS DMS-Ziel-Endpunkte, bei denen Fehler in eine Ausnahmetabelle geschrieben werden.
- AWS DMS unterstützt keine Verbindung mit einem Amazon-ES-Cluster, für den die differenzierte Zugriffskontrolle mit Hauptbenutzer und Passwort aktiviert ist.
- AWS DMS unterstützt OpenSearch Service Serverless nicht.
- OpenSearch Service unterstützt das Schreiben von Daten in bereits vorhandene Indizes nicht.

Zieldatentypen für Amazon OpenSearch Service

Wenn AWS DMS Daten aus heterogenen Datenbanken migriert, ordnet der Service Datentypen aus der Quelldatenbank Zwischendatentypen zu, die als AWS DMS-Datentypen bezeichnet werden. Der Service ordnet dann die Zwischendatentypen den Zieldatentypen zu. In der folgenden Tabelle sind die einzelnen AWS DMS-Datentypen mit dem jeweiligen Datentyp aufgeführt, zu dem die Zuordnung in OpenSearch Service erfolgt.

AWS DMS-Datentyp	OpenSearch-Service-Datentyp
Boolesch	boolesch
Datum	Zeichenfolge
Zeit	date
Zeitstempel	date
INT4	Ganzzahl
Real4	float
UINT4	Ganzzahl

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

Verwenden von Amazon DocumentDB als Ziel für den AWS Database Migration Service

Informationen zu den Versionen von Amazon DocumentDB (mit MongoDB-Kompatibilität), die AWS DMS unterstützt, finden Sie unter [Ziele für AWS DMS](#). Sie können mithilfe von AWS DMS Daten zu Amazon DocumentDB (mit MongoDB-Kompatibilität) migrieren – von jeder der Quelldaten-Engines aus, die AWS DMS unterstützt. Die Quell-Engine kann sich in einem von AWS verwalteten Service wie beispielsweise Amazon RDS, Aurora oder Amazon S3 befinden. Die Engine kann sich außerdem in einer selbstverwalteten Datenbank befinden, wie beispielsweise MongoDB, das auf Amazon EC2 oder On-Premises ausgeführt wird.

Sie können AWS DMS verwenden, um Quelldaten zu Datenbanken, Sammlungen oder Dokumenten in Amazon DocumentDB zu replizieren.

Note

Wenn der Quellendpunkt MongoDB oder Amazon DocumentDB ist, führen Sie die Migration im Dokumentmodus aus.

MongoDB speichert Daten in einem binären JSON-Format (BSON). AWS DMS unterstützt alle BSON-Datentypen, die von Amazon DocumentDB unterstützt werden. Eine Liste dieser Datentypen finden Sie unter [Supported MongoDB APIs, operations, and data types](#) im Entwicklerhandbuch für Amazon DocumentDB.

Wenn es sich beim Quellendpunkt um eine relationale Datenbank handelt, ordnet AWS DMS Amazon DocumentDB Datenbankobjekte wie folgt zu:

- Eine relationale Datenbank oder ein Datenbankschema wird einer Datenbank in Amazon DocumentDB zugeordnet.
- Tabellen innerhalb einer relationalen Datenbank werden Sammlungen in Amazon DocumentDB zugeordnet.
- Datensätze in einer relationalen Tabelle werden Dokumenten in Amazon DocumentDB zugeordnet. Jedes Dokument wird aus den Daten des Quelldatensatzes erstellt.

Wenn der Quellendpunkt Amazon S3 ist, entsprechen die resultierenden Amazon-DocumentDB-Objekte den AWS DMS-Zuordnungsregeln für Amazon S3. Betrachten Sie beispielsweise die folgende URI:

```
s3://mybucket/hr/employee
```

In diesem Fall ordnet AWS DMS die Objekte in mybucket Amazon DocumentDB wie folgt zu:

- Der oberste URI-Teil (hr) wird einer Datenbank in Amazon DocumentDB zugeordnet.
- Der nächste URI-Teil (employee) wird einer Sammlung in Amazon DocumentDB zugeordnet.
- Jedes Objekt in employee wird einem Dokument in Amazon DocumentDB zugeordnet.

Weitere Informationen zu den Zuordnungsregeln für Amazon S3 finden Sie unter [Verwenden von Amazon S3 als Quelle für AWS DMS](#).

Amazon-DocumentDB-Endpunkteinstellungen

In den AWS DMS-Versionen 3.5.0 und höher können Sie die Leistung der Erfassung von Datenänderungen (Change Data Capture, CDC) für Amazon-DocumentDB-Endpunkte verbessern, indem Sie die Aufgabeneinstellungen für parallele Threads und Massenoperationen optimieren. Dazu können Sie die Anzahl der gleichzeitigen Threads, der Warteschlangen pro Thread und die Anzahl der Datensätze angeben, die in einem Puffer unter Verwendung von `ParallelApply*`

Aufgabeneinstellungen gespeichert werden sollen. Beispiel: Sie möchten eine CDC-Last durchführen und 128 Threads parallel anwenden. Außerdem möchten Sie auf 64 Warteschlangen pro Thread zugreifen, wobei 50 Datensätze pro Puffer gespeichert sind.

Zur Steigerung der CDC-Leistung werden folgende Aufgabeneinstellungen von AWS DMS unterstützt:

- `ParallelApplyThreads` – Gibt die Anzahl der gleichzeitigen Threads an, die AWS DMS während einer CDC-Last verwendet, um Datensätze an einen Amazon-DocumentDB-Zielendpunkt zu übertragen. Der Standardwert ist Null (0) und der maximale Wert ist 32.
- `ParallelApplyBufferSize` – Gibt die maximale Anzahl von Datensätzen an, die in jeder Pufferwarteschlange für gleichzeitige Threads gespeichert werden sollen, um sie während einer CDC-Last an einen Amazon-DocumentDB-Zielendpunkt zu übertragen. Der Standardwert ist 100 und der Höchstwert 1 000. Verwenden Sie diese Option, wenn `ParallelApplyThreads` mehrere Threads angibt.
- `ParallelApplyQueuesPerThread` – Gibt die Anzahl der Warteschlangen an, auf die jeder Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und während CDC eine Stapellast für einen Amazon-DocumentDB-Endpunkt zu generieren. Der Standardwert ist 1. Der Höchstwert ist 512.

Weitere Informationen zum Arbeiten mit Amazon DocumentDB als Ziel für AWS DMS finden Sie in den folgenden Abschnitten:

Themen

- [Zuordnen von Daten von einer Quelle zu einem Amazon-DocumentDB-Ziel](#)
- [Herstellen einer Verbindung mit elastischen Amazon-DocumentDB-Clustern als Ziel](#)
- [Laufende Replikation mit Amazon DocumentDB als Ziel](#)
- [Einschränkungen bei Verwendung von Amazon DocumentDB als Ziel](#)
- [Verwenden von Endpunkteinstellungen mit Amazon DocumentDB als Ziel](#)
- [Zieldatentypen für Amazon DocumentDB](#)

Note

Eine ausführliche Anleitung zum Migrationsprozess finden Sie unter [Migrating from MongoDB to Amazon DocumentDB](#) im Schritt-für-Schritt-Migrationshandbuch für AWS Database Migration Service.

Zuordnen von Daten von einer Quelle zu einem Amazon-DocumentDB-Ziel

AWS DMS liest Datensätze aus dem Quellendpunkt und erstellt JSON-Dokumente basierend auf den gelesenen Daten. Für jedes JSON-Dokument muss AWS DMS ein `_id`-Feld bestimmen, das als eindeutige Kennung fungiert. Anschließend wird das JSON-Dokument in eine Amazon-DocumentDB-Sammlung geschrieben, wobei das Feld `_id` als Primärschlüssel verwendet wird.

Quelldaten, die eine einzelne Spalte darstellen

Wenn die Quelldaten aus einer einzelnen Spalte bestehen, müssen die Daten vom Typ Zeichenfolge sein. (Abhängig von der Quell-Engine kann der tatsächliche Datentyp VARCHAR, NVARCHAR, TEXT, LOB, CLOB o. Ä. sein.) AWS DMS geht davon aus, dass es sich bei den Daten um ein gültiges JSON-Dokument handelt, und repliziert die Daten unverändert nach Amazon DocumentDB.

Wenn das resultierende JSON-Dokument ein Feld mit dem Namen `_id` enthält, wird dieses Feld als eindeutige `_id` in Amazon DocumentDB verwendet.

Wenn das JSON-Dokument kein `_id`-Feld enthält, generiert Amazon DocumentDB automatisch einen `_id`-Wert.

Quelldaten, die mehrere Spalten darstellen

Wenn die Quelldaten aus mehreren Spalten bestehen, erstellt AWS DMS aus all diesen Spalten ein JSON-Dokument. Um das `_id`-Feld für das Dokument zu bestimmen, geht AWS DMS wie folgt vor:

- Wenn der Name einer der Spalten `_id` lautet, dann werden die Daten in dieser Spalte als Ziel-`_id` verwendet.
- Wenn keine `_id`-Spalte vorhanden ist, aber die Quelldaten über einen Primärschlüssel oder einen eindeutigen Index verfügen, dann verwendet AWS DMS diesen Schlüssel oder Indexwert als `_id`-Wert. Die Daten aus dem Primärschlüssel oder eindeutigen Index erscheinen auch als explizite Felder im JSON-Dokument.
- Wenn keine `_id`-Spalte vorhanden ist und es keinen Primärschlüssel oder eindeutigen Index gibt, generiert Amazon DocumentDB automatisch einen `_id`-Wert.

Zwingen eines Datentyps an den Zielendpunkt

AWS DMS kann Datenstrukturen beim Schreiben in einen Amazon-DocumentDB-Zielendpunkt ändern. Sie können diese Änderungen anfordern, indem Sie Spalten und Tabellen am Quellendpunkt umbenennen oder Transformationsregeln bereitstellen, die bei der Ausführung einer Aufgabe angewendet werden.

Verwenden eines verschachtelten JSON-Dokuments (json_prefix)

Um einen Datentyp zu erzwingen, können Sie dem Namen der Quellspalte das Präfix `json_` (d. h. `json_columnName`) entweder manuell oder mittels einer Transformation voranstellen. In diesem Fall wird die Spalte nicht als Zeichenfolgenfeld, sondern als verschachteltes JSON-Dokument innerhalb des Zieldokuments erstellt.

Angenommen, Sie möchten beispielsweise das folgende Dokument von einem MongoDB-Quellenendpunkt migrieren.

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": "{\"Home\": {\"Address\": \"Boston\", \"Phone\": \"1111111\"}, \"Work\": { \"Address\": \"Boston\", \"Phone\": \"2222222222\"}}"
```

Wenn Sie keinen der Quelldatentypen erzwingen, wird das eingebettete `ContactDetails`-Dokument als Zeichenfolge migriert.

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": "{\"Home\": {\"Address\": \"Boston\", \"Phone\": \"1111111\"}, \"Work\": { \"Address\": \"Boston\", \"Phone\": \"2222222222\"}}"
```

Sie können jedoch eine Transformationsregel hinzufügen, um `ContactDetails` zu einem JSON-Objekt umzuwandeln. Angenommen, der ursprüngliche Name der Quellspalte ist `ContactDetails`. Um den Datentyp als verschachteltes JSON zu erzwingen, muss die Spalte am Quellendpunkt in „`json_ContactDetails`“ umbenannt werden, indem der Quelle entweder manuell oder mithilfe von

Transformationsregeln das Präfix „*json_*“ hinzugefügt wird. Sie können beispielsweise die folgende Transformationsregel verwenden:

```
{
  "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 repliziert das Feld ContactDetails wie folgt als verschachteltes JSON.

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": {
    "Home": {
      "Address": "Boston",
      "Phone": "1111111111"
    },
    "Work": {
      "Address": "Boston",
      "Phone": "2222222222"
    }
  }
}
```

Verwenden eines JSON-Arrays (array_prefix)

Um einen Datentyp zu erzwingen, können Sie einem Spaltennamen das Präfix `array_` (d. h. `array_columnName`) entweder manuell oder mittels einer Transformation voranstellen. In diesem Fall betrachtet AWS DMS die Spalte als JSON-Array und erstellt sie als solches im Zieldokument.

Angenommen, Sie möchten das folgende Dokument von einem MongoDB-Quellenendpunkt migrieren.

```
{
  "_id" : "1",
  "FirstName": "John",
  "LastName": "Doe",

  "ContactAddresses": ["Boston", "New York"],

  "ContactPhoneNumbers": ["1111111111", "2222222222"]
}
```

Wenn Sie keinen der Quelldatentypen erzwingen, wird das eingebettete `ContactDetails`-Dokument als Zeichenfolge migriert.

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",

  "ContactAddresses": "[\"Boston\", \"New York\"]",

  "ContactPhoneNumbers": "[\"1111111111\", \"2222222222\"]"
}
```

Sie können jedoch Transformationsregeln hinzufügen, um `ContactAddress` und `ContactPhoneNumbers` zu JSON-Arrays umzuwandeln, wie in der folgenden Tabelle veranschaulicht.

Ursprünglicher Name der Quellspalte	Umbenannte Quellspalte
<code>ContactAddress</code>	<code>array_ContactAddress</code>

Ursprünglicher Name der Quellspalte	Umbenannte Quellspalte
ContactPhoneNumbers	array_ContactPhoneNumbers

AWS DMS repliziert ContactAddress und ContactPhoneNumbers wie folgt.

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactAddresses": [
    "Boston",
    "New York"
  ],
  "ContactPhoneNumbers": [
    "1111111111",
    "2222222222"
  ]
}
```

Herstellen einer Verbindung mit Amazon DocumentDB über TLS

Standardmäßig akzeptiert ein neu erstellter Amazon-DocumentDB-Cluster sichere Verbindungen nur mit Transport Layer Security (TLS). Wenn TLS aktiviert ist, erfordert jede Verbindung mit Amazon DocumentDB einen öffentlichen Schlüssel.

Sie können den öffentlichen Schlüssel für Amazon DocumentDB abrufen, indem Sie die Datei `rds-combined-ca-bundle.pem` aus einem von AWS gehosteten Amazon-S3-Bucket herunterladen. Weitere Informationen zum Herunterladen dieser Datei finden Sie unter [Encrypting connections using TLS](#) im Entwicklerhandbuch für Amazon DocumentDB.

Nachdem Sie diese PEM-Datei heruntergeladen haben, können Sie den darin enthaltenen öffentlichen Schlüssel wie im Folgenden beschrieben in AWS DMS importieren.

AWS Management Console

So importieren Sie die Datei mit dem öffentlichen Schlüssel (.pem)

1. Öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms>.
2. Wählen Sie im Navigationsbereich Certificates aus.

3. Wählen Sie **Import certificate** (Zertifikat importieren) aus und gehen Sie folgendermaßen vor:
 - Geben Sie für **Certificate Identifier** (Zertifikat-ID) einen eindeutigen Namen für das Zertifikat ein, z. B. `docdb-cert`.
 - Für **Import file** (Datei importieren) navigieren Sie zum Speicherort der `.pem`-Datei.

Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie **Add new CA certificate** (Neues CA-Zertifikat hinzufügen) aus.

AWS CLI

Verwenden Sie den Befehl `aws dms import-certificate`, wie im folgenden Beispiel gezeigt.

```
aws dms import-certificate \  
  --certificate-identifier docdb-cert \  
  --certificate-pem file://./rds-combined-ca-bundle.pem
```

Wenn Sie einen AWS DMS-Zielendpunkt erstellen, stellen Sie die Zertifikat-ID bereit (z. B. `docdb-cert`). Legen Sie außerdem den SSL-Modus-Parameter auf `verify-full` fest.

Herstellen einer Verbindung mit elastischen Amazon-DocumentDB-Clustern als Ziel

In den AWS DMS-Versionen 3.4.7 und höher können Sie einen Amazon-DocumentDB-Zielendpunkt als elastischen Cluster erstellen. Wenn Sie Ihren Zielendpunkt als elastischen Cluster erstellen, müssen Sie ein neues SSL-Zertifikat an Ihren Endpunkt im elastischen Amazon-DocumentDB-Cluster anfügen, da Ihr vorhandenes SSL-Zertifikat nicht funktioniert.

So fügen Sie ein neues SSL-Zertifikat an Ihren Endpunkt im elastischen Amazon-DocumentDB-Cluster an

1. Öffnen Sie in einem Browser <https://www.amazontrust.com/repository/SFSRootCAG2.pem> und speichern Sie den Inhalt in einer `.pem`-Datei mit eindeutigem Dateinamen, z. B. `SFSRootCAG2.pem`. Dies ist die Zertifikatsdatei, die Sie in den nachfolgenden Schritten importieren müssen.
2. Erstellen Sie den Endpunkt im elastischen Cluster und legen Sie die folgenden Optionen fest:
 - a. Wählen Sie unter **Endpunktkonfiguration** die Option **Neues CA-Zertifikat hinzufügen** aus.
 - b. Geben Sie für **Zertifikat-ID** **SFSRootCAG2.pem** ein.

- c. Wählen Sie unter Zertifikatdatei importieren die Option Datei auswählen und navigieren Sie zur Datei SFSRootCAG2.pem, die Sie zuvor heruntergeladen haben.
- d. Wählen Sie die heruntergeladene Datei SFSRootCAG2.pem aus und öffnen Sie sie.
- e. Wählen Sie Import certificate (Zertifikat importieren).
- f. Wählen Sie in der Drop-down-Liste Zertifikat auswählen die Option SFSRootCAG2.pem aus.

Das neue SSL-Zertifikat aus der heruntergeladenen Datei SFSRootCAG2.pem ist jetzt an Ihren Endpunkt im elastischen Amazon-DocumentDB-Cluster angehängt.

Laufende Replikation mit Amazon DocumentDB als Ziel

Wenn die laufende Replikation (Erfassung von Datenänderungen, CDC) für Amazon DocumentDB als Ziel aktiviert ist, bieten die AWS DMS-Versionen 3.5.0 und höher eine zwanzigfache Leistungsverbesserung gegenüber früheren Versionen. In früheren Versionen verarbeitete AWS DMS bis zu 250 Datensätze pro Sekunde, wohingegen AWS DMS jetzt etwa 5 000 Datensätze/Sekunde verarbeitet. AWS DMS stellt außerdem sicher, dass Dokumente in Amazon DocumentDB mit der Quelle synchronisiert bleiben. Wenn ein Quelldatensatz erstellt oder aktualisiert wird, muss AWS DMS zunächst wie folgt ermitteln, welcher Amazon-DocumentDB-Datensatz betroffen ist:

- Wenn der Quelldatensatz über eine Spalte mit dem Namen `_id` verfügt, bestimmt der Wert dieser Spalte den entsprechenden `_id`-Wert in der Amazon-DocumentDB-Sammlung.
- Wenn keine `_id`-Spalte vorhanden ist, aber die Quelldaten über einen Primärschlüssel oder einen eindeutigen Index verfügen, verwendet AWS DMS diesen Schlüssel oder Indexwert als `_id` für die Amazon-DocumentDB-Sammlung.
- Wenn der Quelldatensatz keine `_id`-Spalte, keinen Primärschlüssel und keinen eindeutigen Index hat, ordnet AWS DMS alle Quellspalten den entsprechenden Feldern in der Amazon-DocumentDB-Sammlung zu.

Wenn ein neuer Quelldatensatz erstellt wird, schreibt AWS DMS ein entsprechendes Dokument in Amazon DocumentDB. Wenn ein vorhandener Quelldatensatz aktualisiert wird, aktualisiert AWS DMS die entsprechenden Felder im Zieldokument in Amazon DocumentDB. Alle Felder, die zwar im Zieldokument, aber nicht im Quelldatensatz vorhanden sind, bleiben unberührt.

Wenn ein Quelldatensatz gelöscht wird, löscht AWS DMS das entsprechende Dokument aus Amazon DocumentDB.

Strukturelle Änderungen (DDL) an der Quelle

Bei der laufenden Replikation werden alle Änderungen an Quelldatenstrukturen (z. B. Tabellen, Spalten usw.) an ihre Entsprechungen in Amazon DocumentDB weitergegeben. In relationalen Datenbanken werden diese Änderungen durch DDL-Anweisungen (Data Definition Language) eingeleitet. In der folgenden Tabelle können Sie sehen, wie AWS DMS diese Änderungen an Amazon DocumentDB weitergibt.

DDL an Quelle	Auswirkung auf das Amazon-DocumentDB-Ziel
CREATE TABLE	Erstellt eine leere Sammlung.
Anweisung, die eine Tabelle umbenennt (RENAME TABLE, ALTER TABLE...RENAME und Ähnliches)	Benennt die Sammlung um.
TRUNCATE TABLE	Entfernt alle Dokumente aus der Sammlung, aber nur, wenn <code>HandleSourceTableTruncated true</code> ist. Weitere Informationen finden Sie unter Aufgabeneinstellungen für den Umgang mit der DDL-Änderungsverarbeitung .
DROP TABLE	Löscht die Sammlung, aber nur, wenn <code>HandleSourceTableDropped true</code> ist. Weitere Informationen finden Sie unter Aufgabeneinstellungen für den Umgang mit der DDL-Änderungsverarbeitung .
Anweisung, die einer Tabelle eine Spalte hinzufügt (ALTER TABLE...ADD und Ähnliches)	Die DDL-Anweisung wird ignoriert und eine Warnung ausgegeben. Wenn der erste INSERT an der Quelle ausgeführt wird, wird das neue Feld dem Zieldokument hinzugefügt.
ALTER TABLE...RENAME COLUMN	Die DDL-Anweisung wird ignoriert und eine Warnung ausgegeben. Wenn der erste INSERT an der Quelle ausgeführt wird, wird das neu benannte Feld dem Zieldokument hinzugefügt.

DDL an Quelle	Auswirkung auf das Amazon-DocumentDB-Ziel
ALTER TABLE...DROP COLUMN	Die DDL-Anweisung wird ignoriert und eine Warnung ausgegeben.
Anweisung, die den Datentyp der Spalte ändert (ALTER COLUMN...MODIFY und Ähnliches)	Die DDL-Anweisung wird ignoriert und eine Warnung ausgegeben. Wenn der erste INSERT an der Quelle mit dem neuen Datentyp ausgeführt wird, wird das Zieldokument mit einem Feld dieses neuen Datentyps erstellt.

Einschränkungen bei Verwendung von Amazon DocumentDB als Ziel

Die folgenden Einschränkungen gelten bei Verwendung von Amazon DocumentDB als Ziel für AWS DMS:

- In Amazon DocumentDB dürfen Namen für Sammlungen nicht das Dollarzeichen (\$) enthalten. Darüber hinaus dürfen Datenbanknamen keine Unicode-Zeichen enthalten.
- AWS DMS unterstützt nicht das Zusammenführen mehrerer Quelltabellen zu einer einzigen Amazon-DocumentDB-Sammlung.
- Wenn AWS DMS Änderungen aus einer Quelltable verarbeitet, die keinen Primärschlüssel hat, werden alle LOB-Spalten in dieser Tabelle ignoriert.
- Wenn die Option Change table (Änderungstabelle) aktiviert ist und AWS DMS auf eine Quellspalte mit dem Namen "_id" trifft, dann erscheint diese Spalte als "__id" (zwei Unterstriche) in der Änderungstabelle.
- Wenn Sie Oracle als Quellendpunkt wählen, muss für die Oracle-Quelle die vollständige ergänzende Protokollierung aktiviert sein. Andernfalls werden, wenn es an der Quelle Spalten gibt, die nicht geändert wurden, die Daten als Nullwerte in Amazon DocumentDB geladen.
- Die Einstellung `TargetTablePrepMode: TRUNCATE_BEFORE_LOAD` für die Replikationsaufgabe wird nicht für die Verwendung mit einem DocumentDB-Zielendpunkt unterstützt.

Verwenden von Endpunkteinstellungen mit Amazon DocumentDB als Ziel

Sie können Endpunkteinstellungen, ähnlich wie zusätzliche Verbindungsattribute, zum Konfigurieren Ihrer Amazon-DocumentDB-Zieldatenbank verwenden. Sie legen die Einstellungen fest, wenn Sie den Zielendpunkt mithilfe der AWS DMS-Konsole erstellen oder indem Sie den Befehl `create-`

endpoint in der [AWS CLI](#) aufrufen. Verwenden Sie die JSON-Syntax `--doc-db-settings '{"EndpointSetting": "value", ...}'`.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Amazon DocumentDB als Ziel verwenden können.

Attributname	Zulässige Werte	Standardwert und Beschreibung
replicateShardCollections	boolesch true false	<p>Wann <code>true</code> festgelegt ist, hat diese Endpunkteinstellung die folgenden Auswirkungen und beinhaltet die folgenden Einschränkungen:</p> <ul style="list-style-type: none"> • AWS DMS darf Daten in Ziel-Shard-Sammlungen replizieren. Diese Einstellung gilt nur, wenn es sich beim DocumentDB-Zielendpunkt um einen elastischen Cluster handelt. • Sie müssen <code>TargetTablePrepMode</code> auf <code>DO_NOTHING</code> setzen. • AWS DMS setzt <code>useUpdateLookup</code> während der Migration automatisch auf <code>false</code>.

Ziel datentypen für Amazon DocumentDB

In der folgenden Tabelle finden Sie die Amazon-DocumentDB-Ziel datentypen, die bei Verwendung von AWS DMS unterstützt werden, sowie deren Standardzuweisung von AWS-DMS-Datentypen. Weitere Informationen zu AWS-DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

AWS-DMS-Datentyp	Amazon-DocumentDB-Datentyp
BOOLEAN	Boolesch
BYTES	Binäre Daten
DATUM	Datum
TIME	String (UTF8)

AWS-DMS-Datentyp	Amazon-DocumentDB-Datentyp
DATETIME	Datum
INT1	32-Bit-Ganzzahl
INT2	32-Bit-Ganzzahl
INT4	32-Bit-Ganzzahl
INT8	64-Bit-Ganzzahl
NUMERIC	String (UTF8)
REAL4	Double
REAL8	Double
STRING	Wenn die Daten als JSON erkannt werden, migriert AWS DMS sie als Dokument nach Amazon DocumentDB. Andernfalls werden die Daten dem String (UTF8) zugewiesen.
UINT1	32-Bit-Ganzzahl
UINT2	32-Bit-Ganzzahl
UINT4	64-Bit-Ganzzahl
UINT8	String (UTF8)
WSTRING	Wenn die Daten als JSON erkannt werden, migriert AWS DMS sie als Dokument nach Amazon DocumentDB. Andernfalls werden die Daten dem String (UTF8) zugewiesen.
BLOB	Binary
CLOB	Wenn die Daten als JSON erkannt werden, migriert AWS DMS sie als Dokument nach Amazon DocumentDB. Andernfalls werden die Daten dem String (UTF8) zugewiesen.

AWS-DMS-Datentyp	Amazon-DocumentDB-Datentyp
NCLOB	Wenn die Daten als JSON erkannt werden, migriert AWS DMS sie als Dokument nach Amazon DocumentDB. Andernfalls werden die Daten dem String (UTF8) zugewiesen.

Verwenden von Amazon Neptune als Ziel für AWS Database Migration Service

Amazon Neptune ist ein schneller, zuverlässiger, vollständig verwalteter Graph-Datenbankservice, mit dem es ganz einfach ist, Anwendungen zu erstellen und auszuführen, die mit stark verbundenen Datensätzen arbeiten. Den Kern von Neptune bildet eine speziell entwickelte, hochleistungsfähige Graphdatenbank-Engine. Diese Engine ist für die Speicherung von Milliarden von Beziehungen und die Abfrage des Graphen mit einer Latenzzeit von Millisekunden optimiert. Neptune unterstützt die gängigen Graph-Abfragesprachen Apache TinkerPop Gremlin und W3C SPARQL. Weitere Informationen zu Amazon Neptune finden Sie unter [Was ist Amazon Neptune?](#) im Amazon-Neptune-Benutzerhandbuch.

Ohne eine Graphdatenbank wie Neptune modellieren Sie wahrscheinlich hoch verbundene Daten in einer relationalen Datenbank. Da die Daten über potenziell dynamische Verbindungen verfügen, müssen Anwendungen, die solche Datenquellen verwenden, verbundene Datenabfragen in SQL modellieren. Dieser Ansatz erfordert, dass Sie einen zusätzlichen Layer schreiben, um Graph-Abfragen in SQL zu konvertieren. Außerdem haben relationale Datenbanken eine Schemastarrheit. Änderungen am Schema an modelländernden Verbindungen erfordern Ausfallzeiten und zusätzliche Wartung der Abfragekonvertierung, um das neue Schema zu unterstützen. Die Abfrageleistung ist auch eine weitere große Einschränkung, die beim Entwerfen Ihrer Anwendungen berücksichtigt werden muss.

Graph-Datenbanken können solche Situationen erheblich vereinfachen. Frei von einem Schema, einem Rich Graph Query Layer (Gremlin oder SPARQL) und Indizes, die für Graph-Abfragen optimiert sind, erhöhen die Flexibilität und Leistung. Die Graphdatenbank von Amazon Neptune verfügt auch über Enterprise-Features wie Verschlüsselung im Ruhezustand, eine sichere Autorisierungsebene, Standardsicherungen, Multi-AZ-Unterstützung, Unterstützung für Lesereplikate und mehr.

Mit AWS DMS können Sie relationale Daten migrieren, die einen stark verbundenen Graphen von einem DMS-Quellendpunkt für jede unterstützte SQL-Datenbank auf einen Neptune-Zielendpunkt modellieren.

Weitere Informationen finden Sie im Folgenden.

Themen

- [Übersicht über die Migration zu Amazon Neptune als Ziel](#)
- [Angaben von Endpunkteinstellungen für Amazon Neptune als Ziel](#)
- [Erstellen einer IAM-Servicerolle für den Zugriff auf Amazon Neptune als Ziel](#)
- [Angaben von Graph-Zuordnungsregeln mit Gremlin und R2RML für Amazon Neptune als Ziel](#)
- [Datentypen für die Migration von Gremlin und R2RML zu Amazon Neptune als Ziel](#)
- [Einschränkungen bei der Verwendung von Amazon Neptune als Ziel](#)

Übersicht über die Migration zu Amazon Neptune als Ziel

Bevor Sie mit der Migration zu einem Neptune-Ziel beginnen, erstellen Sie die folgenden Ressourcen in Ihrem AWS-Konto:

- Einen Neptune-Cluster für den Zielendpunkt.
- Eine relationale SQL-Datenbank, die von AWS DMS unterstützt wird, für den Quellendpunkt.
- Einen Amazon-S3-Bucket für den Zielendpunkt. Erstellen Sie diesen S3-Bucket in derselben AWS-Region wie Ihren Neptune-Cluster. AWS DMS verwendet diesen S3-Bucket als Zwischendateispeicher für die Zieldaten, die in einem Masseneingabevorgang in die Neptune-Datenbank geladen werden. Weitere Informationen zum Erstellen eines S3-Buckets finden Sie unter [Erstellen von Buckets](#) im Benutzerhandbuch für Amazon Simple Storage Service.
- Einen Virtual Private Cloud (VPC)-Endpunkt für S3 in derselben VPC wie der Neptune-Cluster.
- Eine AWS Identity and Access Management(IAM)-Rolle, die eine IAM-Richtlinie enthält. In dieser Richtlinie sollten die Berechtigungen `ListObject`, `GetObject`, `PutObject` und `DeleteObject` für den S3-Bucket für den Zielendpunkt angegeben werden. Diese Rolle wird sowohl von AWS DMS als auch von Neptune mit IAM-Zugriff auf den Ziel-S3-Bucket und die Neptune-Datenbank übernommen. Weitere Informationen finden Sie unter [Erstellen einer IAM-Servicerolle für den Zugriff auf Amazon Neptune als Ziel](#).

Nachdem Sie über diese Ressourcen verfügen, sind die Einrichtung und das Starten einer Migration zu einem Neptune-Ziel ähnlich wie bei einer Volllastmigration unter Verwendung der Konsole oder der DMS-API. Eine Migration zu einem Neptune-Ziel erfordert jedoch einige spezifische Schritte.

So migrieren Sie eine relationale AWS DMS-Datenbank zu Neptune

1. Erstellen Sie eine Replikations-Instance wie unter [Erstellen einer Replikations-Instance](#) beschrieben.
2. Erstellen und testen Sie eine relationale SQL-Datenbank, die von AWS DMS für den Quellendpunkt unterstützt wird.
3. Erstellen und testen Sie den Zielendpunkt für Ihre Neptune-Datenbank.

Um den Zielendpunkt mit der Neptune-Datenbank zu verbinden, geben Sie den Servernamen für den Neptune-Cluster-Endpunkt oder den Neptune-Writer-Instance-Endpunkt an. Geben Sie außerdem den S3-Bucket-Ordner für AWS DMS an, um die Zwischendateien für das Masseladen in die Neptune-Datenbank zu speichern.

Während der Migration speichert AWS DMS alle migrierten Zieldaten in diesem S3-Bucket-Ordner bis zu einer von Ihnen angegebenen maximalen Dateigröße. Wenn dieser Dateispeicher diese maximale Größe erreicht, lädt AWS DMS in einem Masseladevorgang die gespeicherten S3-Daten in die Zieldatenbank. Der Ordner wird gelöscht, um die Speicherung zusätzlicher Zieldaten für das nachfolgende Laden in die Zieldatenbank zu ermöglichen. Weitere Informationen zum Festlegen dieser Einstellungen finden Sie unter [Angeben von Endpunkteinstellungen für Amazon Neptune als Ziel](#).

4. Erstellen Sie eine Volllast-Replikationsaufgabe mit den Ressourcen, die in den Schritten 1 bis 3 erstellt wurden, und gehen Sie wie folgt vor:
 - a. Verwenden Sie die Aufgaben-Tabellenzuordnung wie gewohnt, um bestimmte Quellschemas, Tabellen und Ansichten zu identifizieren, die mit entsprechenden Auswahl- und Transformationsregeln aus Ihrer relationalen Datenbank migriert werden sollen. Weitere Informationen finden Sie unter [Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen](#).
 - b. Geben Sie Zielzuordnungen an, indem Sie eine der folgenden Optionen auswählen, um Zuordnungsregeln aus Quelltabellen und Ansichten für den Neptune-Zieldatenbank-Graphen anzugeben:

- Gremlin JSON – Informationen zur Verwendung von Gremlin JSON zum Laden einer Neptune-Datenbank finden Sie unter [Gremlin-Format zum Laden von Daten](#) im Amazon-Neptune-Benutzerhandbuch.
 - SPARQL RDB to Resource Description Framework Mapping Language (R2RML) – Weitere Informationen zur Verwendung von SPARQL R2RML finden Sie in der W3C-Spezifikation [R2RML: RDB to RDF Mapping Language](#).
- c. Führen Sie eine der folgenden Aktionen aus:
- Geben Sie mithilfe der AWS DMS-Konsole Graph-Zuordnungsoptionen mithilfe der Graph-Zuordnungsregeln auf der Seite Datenbankmigrationsaufgabe erstellen.
 - Geben Sie mithilfe der AWS DMS-API diese Optionen über den TaskData-Anforderungsparameter des CreateReplicationTask-API-Aufrufs an.

Weitere Informationen und Beispiele für die Verwendung von Gremlin JSON und SPARQL R2RML zum Angeben von Graph-Zuordnungsregeln finden Sie unter [Angeben von Graph-Zuordnungsregeln mit Gremlin und R2RML für Amazon Neptune als Ziel](#).

5. Starten Sie die Replikation für Ihre Migrationsaufgabe.

Angeben von Endpunkteinstellungen für Amazon Neptune als Ziel

Um einen Zielendpunkt zu erstellen oder zu ändern, können Sie die Konsole oder die CreateEndpoint- oder ModifyEndpoint-API-Vorgänge verwenden.

Geben Sie für ein Neptune-Ziel in der AWS DMS-Konsole Endpunktspezifische Einstellungen auf der Konsolenseite Endpunkt erstellen oder Endpunkt ändern an. Geben Sie für CreateEndpoint und ModifyEndpoint Anforderungsparameter für die NeptuneSettings-Option an. Das folgende Beispiel zeigt, wie dies über die Befehlszeilenschnittstelle (CLI) möglich ist.

```
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}'
```

Hier gibt die CLI-Option `--server-name` den Servernamen für den Neptune-Cluster-Writer-Endpoint an. Sie können auch den Servernamen für einen Neptune-Writer-Instance-Endpoint angeben.

Die `--neptune-settings`-Option fordert Parameter wie folgt an:

- `ServiceAccessRoleArn` – (Erforderlich) Der Amazon-Ressourcenname (ARN) der Servicerolle, die Sie für den Neptune-Zielendpunkt erstellt haben. Weitere Informationen finden Sie unter [Erstellen einer IAM-Servicerolle für den Zugriff auf Amazon Neptune als Ziel](#).
- `S3BucketName` – (Erforderlich) Der Name des S3-Buckets, in dem DMS migrierte Diagrammdaten vorübergehend in CSV-Dateien speichern kann, bevor sie in einem Masseladevorgang in die Neptune-Zieldatenbank geladen werden. DMS ordnet die SQL-Quelldaten Diagrammdaten zu, bevor sie in diesen CSV-Dateien gespeichert werden.
- `S3BucketFolder` – (Erforderlich) Ein Ordnerpfad, in dem DMS migrierte Diagrammdaten in dem S3-Bucket speichern soll, der durch `S3BucketName` angegeben wird.
- `ErrorRetryDuration` – (Optional) Die Anzahl der Millisekunden, die DMS auf einen erneuten Masseladevorgang migrierter Diagrammdaten in die Neptune-Zieldatenbank warten soll, bevor ein Fehler ausgelöst wird. Der Standardwert ist 250.
- `MaxFileSize` – (Optional) Die maximale Größe der migrierten Diagrammdaten in KB, die in einer CSV-Datei gespeichert werden, bevor DMS eine Massenladung der Daten in die Neptune-Zieldatenbank vornimmt. Der Standardwert ist 1.048.576 KB (1 GB) Wenn dies erfolgreich ist, löscht DMS den Bucket und ist bereit, den nächsten Stapel der migrierten Diagrammdaten zu speichern.
- `MaxRetryCount` – (Optional) Gibt an, wie oft DMS eine Massenladung migrierter Diagrammdaten in die Neptune-Zieldatenbank erneut versucht, bevor ein Fehler ausgelöst wird. Der Standardwert für ist 5.
- `IAMAuthEnabled` – (Optional) Wenn Sie die IAM-Autorisierung für diesen Endpunkt aktivieren möchten, legen Sie für diesen Parameter `true` fest und fügen Sie das entsprechende IAM-Richtliniendokument zu Ihrer Servicerolle hinzu, die in `ServiceAccessRoleArn` angegeben ist. Der Standardwert ist `false`.

Erstellen einer IAM-Servicerolle für den Zugriff auf Amazon Neptune als Ziel

Um auf Neptune als Ziel zuzugreifen, erstellen Sie mithilfe von IAM eine Dienstrolle. Fügen Sie dieser Rolle abhängig von Ihrer Neptune-Endpunktkonfiguration einige oder alle folgenden IAM-Richtlinien- und Vertrauendokumente an. Wenn Sie den Neptune-Endpunkt erstellen, geben Sie den ARN dieser Servicerolle an. Auf diese Weise können AWS DMS und Amazon Neptune Berechtigungen für den Zugriff auf Neptune und den zugehörigen Amazon-S3-Bucket übernehmen.

Wenn Sie in Ihrer Neptune-Endpunktkonfiguration den Parameter `IAMAuthEnabled` in `NeptuneSettings` auf `true` festlegen, fügen Sie Ihrer Servicerolle eine IAM-Richtlinie wie die folgende hinzu. Wenn Sie `IAMAuthEnabled` auf `false` festlegen, können Sie diese Richtlinie ignorieren.

```
// 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/*"
    }
  ]
}
```

Die vorangehende IAM-Richtlinie ermöglicht vollen Zugriff auf den von Resource angegebenen Neptune-Ziel-Cluster.

Fügen Sie Ihrer Dienstrolle eine IAM-Richtlinie wie die folgende an. Diese Richtlinie ermöglicht es DMS, migrierte Diagrammdaten vorübergehend im S3-Bucket zu speichern, den Sie für den Masseneinladevorgang in die Neptune-Zieldatenbank erstellt haben.

```
//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/"
    ]
  }
]
}

```

Die vorangehende IAM-Richtlinie ermöglicht es Ihrem Konto, den Inhalt des S3-Buckets (`arn:aws:s3:::my-bucket`) abzufragen, der für Ihr Neptune-Ziel erstellt wurde. Außerdem kann Ihr Konto vollständig mit dem Inhalt aller Bucket-Dateien und Ordner arbeiten (`arn:aws:s3:::my-bucket/`).

Bearbeiten Sie die Vertrauensbeziehung und fügen Sie Ihrer Servicerolle die folgende IAM-Rolle an, damit AWS DMS und der Amazon-Neptune-Datenbankservice die Rolle übernehmen können.

```

{
  "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"
  }
]
```

Weitere Informationen zum Angeben dieser Servicerolle für den Neptune-Zielendpunkt finden Sie unter [Angeben von Endpunkteinstellungen für Amazon Neptune als Ziel](#).

Angeben von Graph-Zuordnungsregeln mit Gremlin und R2RML für Amazon Neptune als Ziel

Die von Ihnen erstellten Graph-Zuordnungsregeln geben an, wie Daten, die aus einer relationalen SQL-Datenbankquelle extrahiert werden, in ein Neptune-Datenbank-Cluster-Ziel geladen werden. Das Format dieser Zuweisungsregeln unterscheidet sich je nachdem, ob die Regeln zum Laden von Eigenschaftsdiagrammdaten mit Apache TinkerPop Gremlin oder RDF(Resource Description Framework)-Daten mit R2RML dienen. Im Folgenden finden Sie Informationen zu diesen Formaten und wo Sie mehr darüber erfahren können.

Sie können diese Zuordnungsregeln angeben, wenn Sie die Migrationsaufgabe mithilfe der Konsole oder der DMS-API erstellen.

Geben Sie mithilfe der Graph-Zuordnungsregeln auf der Seite Datenbankmigration erstellen diese Zuordnungsregeln an. In Graph-Zuordnungsregeln können Sie die Zuordnungsregeln direkt mit dem bereitgestellten Editor eingeben und bearbeiten. Sie können auch nach einer Datei suchen, die die Zuordnungsregeln im entsprechenden Graph-Zuordnungsformat enthält.

Geben Sie mithilfe der API diese Optionen über den `TaskData`-Anforderungsparameter des `CreateReplicationTask`-API-Aufrufs an. Legen Sie `TaskData` auf den Pfad einer Datei fest, die die Zuordnungsregeln im entsprechenden Graph-Zuordnungsformat enthält.

Graph-Zuordnungsregeln zum Generieren von Eigenschafts-Graph-Daten mit Gremlin

Verwenden Sie Gremlin zum Generieren der Eigenschafts-Graph-Daten an, geben Sie ein JSON-Objekt mit einer Zuordnungsregel für jede Graph-Entität ein, die aus Quelldaten generiert werden soll. Das Format dieses JSON-Objekts ist speziell für einen Masseladevorgang von Amazon Neptune definiert. Die folgende Vorlage zeigt, wie die einzelnen Regeln in diesem Objekt aussehen:

```
{
  "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)"
        }
    ]
}
]
}
]
}
}

```

Das Vorhandensein eines Vertex-Labels bedeutet, dass der Knoten hier erstellt wird. Seine Abwesenheit bedeutet, dass der Knoten von einer anderen Quelle erstellt wird, und diese Definition fügt nur Knoteneigenschaften hinzu. Geben Sie so viele Knoten- und Grenzdefinitionen an, wie erforderlich, um die Zuordnungen für die gesamte relationale Datenbankquelle anzugeben.

Es folgt eine Beispielregel für eine employee-Tabelle.

```

{
  "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"
        }
      ]
    }
  ]
}
]
}
}

```

Hier ordnen die Knoten- und Grenzdefinitionen eine Meldebeziehung von einem employee-Knoten mit Mitarbeiter-ID (EmpID) und einem employee-Knoten mit einer Manager-ID (managerId) zu.

Weitere Informationen zum Erstellen von Graph-Zuordnungsregeln mit Gremlin JSON finden Sie unter [Gremlin-Format zum Laden von Daten](#) im Amazon-Neptune-Benutzerhandbuch.

Graph-Zuordnungsregeln zum Generieren von RDF/SPARQL-Daten

Wenn Sie RDF-Daten laden, die mit SPARQL abgefragt werden sollen, schreiben Sie die Graph-Zuordnungsregeln in R2RML. R2RML ist eine Standard-W3C-Sprache für die Zuordnung relationaler Daten zu RDF. In einer R2RML-Datei gibt eine Dreifach-Zuordnung (z. B. `<#TriplesMap1>` folgend) eine Regel an, um jede Zeile einer logischen Tabelle in null oder mehr RDF-Triples zu übersetzen. Eine Betreffzuordnung (z. B. eine beliebige `rr:subjectMap` folgend) gibt eine Regel zum Generieren der Betreffende der RDF-Triples an, die von einer Triples-Zuordnung generiert werden. Eine -Prädikat-Objekt-Zuordnung (z. B. eine beliebige `rr:predicateObjectMap` folgend) ist eine Funktion, die ein oder mehrere Prädikat-Objekt-Paare für jede logische Tabellenzeile einer logischen Tabelle erstellt.

Ein einfaches Beispiel für eine nodes-Tabelle folgt.

```
@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" ];
  ]
```

Im vorherigen Beispiel definiert die Zuordnung Graph-Knoten, die aus einer Tabelle mit Mitarbeitern zugeordnet sind.

Ein weiteres einfaches Beispiel für eine Student-Tabelle folgt.

```
@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" ]
  ].
```

Im vorherigen Beispiel definiert die Zuordnung Graph-Knoten, die Freund-eines-Freundes-Beziehungen zwischen Personen in einer Student-Tabelle zuordnen.

Weitere Informationen zum Erstellen von Graph-Zuordnungsregeln mit SPARQL R2RML finden Sie in der W3C-Spezifikation [R2RML: RDB to RDF Mapping Language](#).

Datentypen für die Migration von Gremlin und R2RML zu Amazon Neptune als Ziel

AWS DMS führt eine Datentypzuordnung von Ihrem SQL-Quellendpunkt zu Ihrem Neptune-Ziel auf zwei Arten durch. Welche Art Sie verwenden, hängt vom Graph-Zuordnungsformat ab, das Sie zum Laden der Neptune-Datenbank verwenden:

- Apache TinkerPop Gremlin, mit einer JSON-Darstellung der Migrationsdaten.
- SPARQL von W3C, unter Verwendung einer R2RML-Darstellung der Migrationsdaten.

Weitere Informationen zu diesen beiden Graph-Zuordnungsformaten finden Sie unter [Angeben von Graph-Zuordnungsregeln mit Gremlin und R2RML für Amazon Neptune als Ziel](#).

Nachfolgend finden Sie Beschreibungen der Datentypzuordnungen für jedes Format.

Datentypzuordnungen von SQL-Quelle auf Gremlin-Ziel

Die folgende Tabelle zeigt die Datentypzuordnungen von einer SQL-Quelle auf ein Gremlin-formatiertes Ziel.

AWS DMS ordnet einen beliebigen nicht aufgelisteten SQL-Quelldatentyp einem Gremlin-String zu.

SQL-Quelldatentypen	Gremlin-Zieldatentypen
NUMERIC (und Varianten)	Double
DECIMAL	
TINYINT	Byte
SMALLINT	Short
INT, INTEGER	Int
BIGINT	Long
FLOAT	Float
DOUBLE PRECISION	
REAL	Double
BIT	Boolean
BOOLEAN	
DATE	Date
TIME	
TIMESTAMP	
CHARACTER (und Varianten)	String

Weitere Informationen zu den Gremlin-Datentypen zum Laden von Neptune finden Sie unter [Gremlin-Datentypen](#) im Neptune-Benutzerhandbuch.

SQL-Quell-zu-R2RML-Zieldatentyp-Zuordnungen (RDF)

Die folgende Tabelle zeigt die Datentypzuordnungen von einer SQL-Quelle zu einem R2RML-formatierten Ziel.

Bei allen aufgelisteten RDF-Datentypen wird zwischen Groß- und Kleinschreibung unterschieden, mit Ausnahme des RDF-Literal. AWS DMS ordnet jeden nicht aufgelisteten SQL-Quelldatentyp einem RDF-Literal zu.

Ein RDF-Literal ist eine von einer Vielzahl von wörtlichen lexikalischen Formen und Datentypen. Weitere Informationen finden Sie unter [RDF-Literale](#) in der W3C-Spezifikation Resource Description Framework (RDF): Konzepte und abstrakte Syntax.

SQL-Quelldatentypen	R2RML-(RDF-)Zieldatentypen
BINARY (und Varianten)	xsd:hexBinary
NUMERIC (und Varianten)	xsd:decimal
DECIMAL	
TINYINT	xsd:integer
SMALLINT	
INT, INTEGER	
BIGINT	
FLOAT	xsd:double
DOUBLE PRECISION	
REAL	
BIT	xsd:boolean
BOOLEAN	
DATE	xsd:date
TIME	xsd:time
TIMESTAMP	xsd:dateTime
CHARACTER (und Varianten)	RDF-Literal

Weitere Informationen zu den RDF-Datentypen zum Laden von Neptune und zu ihren Zuordnungen zu SQL-Quelldatentypen finden Sie unter [Datentypkonvertierungen](#) in der W3C-Spezifikation R2RML: RDB to RDF Mapping Language.

Einschränkungen bei der Verwendung von Amazon Neptune als Ziel

Bei der Verwendung von Neptune als Ziel gelten die folgenden Einschränkungen:

- AWS DMS unterstützt derzeit nur Volllastaufgaben für die Migration zu einem Neptune-Ziel. Die CDC-Migration (CDC = Change Data Capture) zu einem Neptune-Ziel wird nicht unterstützt.
- Stellen Sie sicher, dass in Ihrer Neptune-Zieldatenbank alle Daten manuell gelöscht werden, bevor Sie die Migrationsaufgabe starten, wie in den folgenden Beispielen zu sehen ist.

Um alle Daten (Eckpunkte und Edges) innerhalb des Graphen zu löschen, führen Sie den folgenden Gremlin-Befehl aus.

```
gremlin> g.V().drop().iterate()
```

Um Eckpunkte mit dem Label 'customer' zu löschen, führen Sie den folgenden Gremlin-Befehl aus.

```
gremlin> g.V().hasLabel('customer').drop()
```

Note

Es kann einige Zeit dauern, bis ein großes Dataset gelöscht wird. Möglicherweise möchten Sie `drop()` mit einem Limit iterieren, z. B. `limit(1000)`.

Um Edges mit dem Label 'rated' zu löschen, führen Sie den folgenden Gremlin-Befehl aus.

```
gremlin> g.E().hasLabel('rated').drop()
```

Note

Es kann einige Zeit dauern, bis ein großes Dataset gelöscht wird. Möglicherweise möchten Sie `drop()` mit einem Limit iterieren, z. B. `limit(1000)`.

- Der DMS-API-Vorgang `DescribeTableStatistics` kann aufgrund der Art der Neptune-Graph-Datenstrukturen ungenaue Ergebnisse zu einer bestimmten Tabelle zurückgeben.

Während der Migration scannt AWS DMS jede Quelltable und verwendet die Graph-Zuordnung, um die Quelldaten in einen Neptune-Graphen zu konvertieren. Die konvertierten Daten werden zuerst im S3-Bucket-Ordner gespeichert, der für den Zielendpunkt angegeben ist. Wenn die Quelle gescannt wird und diese zwischengeschalteten S3-Daten erfolgreich generiert werden, geht `DescribeTableStatistics` davon aus, dass die Daten erfolgreich in die Neptune-Zieldatenbank geladen wurden. Aber das ist nicht immer wahr. Um zu überprüfen, ob die Daten für eine bestimmte Tabelle korrekt geladen wurden, vergleichen Sie die `count()`-Rückgabewerte an beiden Enden der Migration für diese Tabelle.

Im folgenden Beispiel hat AWS DMS eine `customer`-Tabelle aus der Quelldatenbank geladen, der das Label `'customer'` im Neptune-Zieldatenbank-Graphen zugewiesen ist. Sie können sicherstellen, dass dieses Label in die Zieldatenbank geschrieben wird. Vergleichen Sie dazu die Anzahl der in der Quelldatenbank verfügbaren `customer`-Zeilen mit der Anzahl der mit `'customer'` beschrifteten Zeilen, die nach Abschluss der Aufgabe in die Neptune-Zieldatenbank geladen wurden.

Führen Sie die folgenden Schritte aus, um die Anzahl der Kundenzeilen aus der Quelldatenbank mithilfe von SQL abzurufen.

```
select count(*) from customer;
```

Um die Anzahl der beschrifteten `'customer'`-Zeilen zu erhalten, die mit Gremlin in den Zieldatenbank-Graph geladen wurden, führen Sie Folgendes aus.

```
gremlin> g.V().hasLabel('customer').count()
```

- Wenn derzeit eine einzelne Tabelle nicht geladen werden kann, schlägt die gesamte Aufgabe fehl. Im Gegensatz zu einem relationalen Datenbankziel sind Daten in Neptune stark verbunden, was es in vielen Fällen unmöglich macht, eine Aufgabe fortzusetzen. Wenn eine Aufgabe aufgrund dieser Art von Datenladefehler nicht erfolgreich fortgesetzt werden kann, erstellen Sie eine neue Aufgabe, um die Tabelle zu laden, die nicht geladen werden konnte. Bevor Sie diese neue Aufgabe ausführen, löschen Sie die teilweise geladene Tabelle manuell aus dem Neptune-Ziel.

Note

Sie können eine Aufgabe fortsetzen, bei der die Migration zu einem Neptune-Ziel fehlschlägt, wenn der Fehler behoben werden kann (z. B. ein Netzwerk-Transitfehler).

- AWS DMS unterstützt die meisten Standards für R2RML. Bestimmte R2RML-Standards, einschließlich inverser Ausdrücke, Verknüpfungen und Ansichten, werden jedoch von AWS DMS nicht unterstützt. Eine Problemumgehung für eine R2RML-Ansicht besteht darin, eine entsprechende benutzerdefinierte SQL-Ansicht in der Quelldatenbank zu erstellen. Verwenden Sie in der Migrationsaufgabe die Tabellenzuordnung, um die Ansicht als Eingabe auszuwählen. Ordnen Sie dann die Ansicht einer Tabelle zu, die dann von R2RML verwendet wird, um Diagrammdateien zu generieren.
- Wenn Sie Quelldaten mit nicht unterstützten SQL-Datentypen migrieren, sind die sich daraus ergebenden Zieldaten möglicherweise nicht ganz genau. Weitere Informationen finden Sie unter [Datentypen für die Migration von Gremlin und R2RML zu Amazon Neptune als Ziel](#).
- AWS DMS unterstützt die Migration von LOB-Daten zu einem Neptune-Ziel nicht.

Verwenden von Redis als Ziel für AWS Database Migration Service

Redis ist ein Open-Source-In-Memory-Datenstrukturspeicher, der als Datenbank, Cache und Message Broker verwendet wird. Durch die Verwaltung von Daten im Speicher können Lese- oder Schreibvorgänge unter Umständen in weniger als einer Millisekunde ausgeführt werden und es sind Hunderte Millionen von Vorgängen pro Sekunde möglich. Als In-Memory-Datenspeicher unterstützt Redis die anspruchsvollsten Anwendungen, die Antwortzeiten unter einer Millisekunde erfordern.

Mit AWS DMS können Sie Daten von jeder unterstützten Quelldatenbank mit minimalen Ausfallzeiten zu einem Redis-Zieldatenspeicher migrieren. Weitere Informationen zu Redis finden Sie in der [Redis-Dokumentation](#).

Zusätzlich zu On-Premises-Redis unterstützt AWS Database Migration Service Folgendes:

- [Amazon ElastiCache für Redis](#) als Zieldatenspeicher. ElastiCache für Redis kann mit Ihren Redis-Clients verwendet werden. Es nutzt das offene Redis-Datenformat zum Speichern Ihrer Daten.
- [Amazon MemoryDB für Redis](#) als Zieldatenspeicher. MemoryDB ist mit Redis kompatibel und ermöglicht es Ihnen, Anwendungen mit allen heute verwendeten Redis-Datenstrukturen, -APIs und -Befehlen zu erstellen.

Weitere Informationen zum Arbeiten mit Redis als Ziel für AWS DMS finden Sie in den folgenden Abschnitten:

Themen

- [Voraussetzungen für die Verwendung eines Redis-Clusters als Ziel für AWS DMS](#)
- [Einschränkungen bei der Verwendung von Redis als Ziel für AWS Database Migration Service](#)
- [Migrieren von Daten aus einer relationalen oder nicht relationalen Datenbank zu einem Redis-Ziel](#)
- [Angaben von Endpunkteinstellungen für Redis als Ziel](#)

Voraussetzungen für die Verwendung eines Redis-Clusters als Ziel für AWS DMS

DMS unterstützt ein Redis-On-Premises-Ziel in einer eigenständigen Konfiguration oder als Redis-Cluster, in dem Daten automatisch mittels Sharding auf mehrere Knoten verteilt werden. Sharding ist ein Prozess, bei dem Daten in kleinere Blöcke, sogenannte Shards, aufgeteilt werden, die auf mehrere Server oder Knoten verteilt werden. Ein Shard ist eine Datenpartition, die eine Teilmenge des gesamten Datensatzes enthält und einen Teil der gesamten Workload abdeckt.

Da es sich bei Redis um einen NoSQL-Schlüssel-Wert-Datenspeicher handelt, müssen Sie für die Schlüsselbenennung die Redis-Konvention `schema-name.table-name.primary-key` verwenden, wenn es sich bei Ihrer Quelle um eine relationale Datenbank handelt. In Redis dürfen Schlüssel und Werte nicht das Sonderzeichen `%` enthalten. Andernfalls überspringt DMS den Datensatz.

Note

Wenn Sie ElastiCache für Redis als Ziel verwenden, unterstützt DMS nur Konfigurationen mit aktiviertem Cluster-Modus. Weitere Informationen zur Verwendung von ElastiCache für Redis Version 6.x oder höher zum Erstellen eines Zieldatenspeichers mit aktiviertem Cluster-Modus finden Sie unter [Erste Schritte](#) im Benutzerhandbuch für Amazon ElastiCache für Redis.

Bevor Sie mit einer Datenbankmigration beginnen, starten Sie Ihren Redis-Cluster unter Berücksichtigung der folgenden Kriterien.

- Ihr Cluster enthält einen oder mehrere Shards.
- Wenn Sie ein ElastiCache-für-Redis-Ziel verwenden, stellen Sie sicher, dass Ihr Cluster keine rollenbasierte IAM-Zugriffskontrolle verwendet. Verwenden Sie zur Authentifizierung von Benutzern stattdessen Redis Auth.

- Aktivieren Sie Multi-AZ (Availability Zones).
- Stellen Sie sicher, dass auf dem Cluster genug Speicher für die zu migrierenden Daten aus der Datenbank verfügbar ist.
- Stellen Sie sicher, dass Ihr Redis-Ziel-Cluster vor Beginn der ersten Migrationsaufgabe keine Daten enthält.

Sie sollten Ihre Sicherheitsanforderungen für die Datenmigration ermitteln, bevor Sie Ihre Cluster-Konfiguration erstellen. DMS unterstützt die Migration zu Zielreplikationsgruppen unabhängig von ihrer Verschlüsselungskonfiguration. Sie können die Verschlüsselung jedoch nur aktivieren oder deaktivieren, wenn Sie Ihre Cluster-Konfiguration erstellen.

Einschränkungen bei der Verwendung von Redis als Ziel für AWS Database Migration Service

Bei der Verwendung von Redis als Ziel gelten die folgenden Einschränkungen:

- Da es sich bei Redis um einen NoSQL-Schlüssel-Wert-Datenspeicher handelt, müssen Sie für die Schlüsselbenennung die Redis-Konvention `schema-name.table-name.primary-key` verwenden, wenn es sich bei Ihrer Quelle um eine relationale Datenbank handelt.
- In Redis dürfen Schlüssel und Werte nicht das Sonderzeichen `%` enthalten. Andernfalls überspringt DMS den Datensatz.
- DMS migriert keine Zeilen, die Sonderzeichen enthalten.
- DMS migriert keine Felder, die Sonderzeichen im Feldnamen enthalten.
- Der vollständige LOB-Modus wird nicht unterstützt.
- Bei Verwendung von ElastiCache für Redis wird eine private Zertifizierungsstelle (CA) nicht unterstützt.

Migrieren von Daten aus einer relationalen oder nicht relationalen Datenbank zu einem Redis-Ziel

Sie können Daten aus jedem SQL- oder NoSQL-Quelldatenspeicher direkt zu einem Redis-Ziel migrieren. Die Einrichtung und der Start einer Migration zu einem Redis-Ziel erfolgen ähnlich wie eine Volllastmigration und eine CDC-Migration (Change Data Capture) über die DMS-Konsole oder -API. Gehen Sie für eine Datenbankmigration zu einem Redis-Ziel wie folgt vor:

- Erstellen Sie eine Replikations-Instance, die alle Migrationsprozesse durchführt. Weitere Informationen finden Sie unter [Erstellen einer Replikations-Instance](#).
- Geben Sie einen Quellendpunkt an. Weitere Informationen finden Sie unter [Erstellen der Quell- und Zielendpunkte](#).
- Suchen Sie nach dem DNS-Namen und der Portnummer Ihres Clusters.
- Laden Sie ein Zertifikatpaket herunter, das Sie zur Überprüfung von SSL-Verbindungen verwenden können.
- Geben Sie einen Zielendpunkt an, wie unten beschrieben.
- Erstellen Sie eine oder mehrere Aufgaben, um festzulegen, welche Tabellen und Replikationsprozesse verwendet werden sollen. Weitere Informationen finden Sie unter [Erstellen einer Aufgabe](#).
- Migrieren Sie Daten aus Ihrer Quelldatenbank zu Ihrem Ziel-Cluster.

Sie haben zwei Möglichkeiten, mit einer Datenbankmigration zu beginnen:

1. Sie können die AWS DMS-Konsole auswählen und jeden Schritt dort ausführen.
2. Sie können die AWS Command Line Interface (AWS CLI) verwenden. Weitere Informationen zur Verwendung der CLI mit AWS DMS finden Sie unter [AWS CLI für AWS DMS](#).

So finden Sie den DNS-Namen und die Portnummer Ihres Clusters

- Verwenden Sie den folgenden AWS CLI-Befehl um für `replication-group-id` den Namen Ihrer Replikationsgruppe anzugeben.

```
aws elasticache describe-replication-groups --replication-group-id myreplgroup
```

Hier zeigt die Ausgabe den DNS-Namen im Attribut `Address` und die Portnummer im Attribut `Port` des Primärknotens im Cluster.

```
...
"ReadEndpoint": {
  "Port": 6379,
  "Address": "myreplgroup-
111.1abc1d.1111.uuu1.cache.example.com"
```

```
}
...
```

Wenn Sie MemoryDB für Redis als Ziel verwenden, stellen Sie mithilfe des folgenden AWS CLI-Befehls eine Endpunktadresse für Ihren Redis-Cluster bereit.

```
aws memorydb describe-clusters --clusterid clusterid
```

Laden Sie ein Zertifikatpaket herunter, mit dem SSL-Verbindungen überprüft werden können.

- Geben Sie in der Befehlszeile den folgenden `wget`-Befehl ein: `Wget` ist ein kostenloses GNU-Befehlszeilen-Dienstprogramm, das zum Herunterladen von Dateien aus dem Internet verwendet wird.

```
wget https://s3.aws-api-domain/rds-downloads/rds-combined-ca-bundle.pem
```

Hier vervollständigt `aws-api-domain` die Amazon-S3-Domain in Ihrer AWS-Region, die für den Zugriff auf den angegebenen S3-Bucket und die bereitgestellte Datei „`rds-combined-ca-bundle.pem`“ erforderlich ist.

So erstellen Sie einen Zielendpunkt über die AWS DMS-Konsole

Dieser Endpunkt ist für Ihr Redis-Ziel bestimmt, das bereits ausgeführt wird.

- Wählen Sie im Navigationsbereich Endpunkte und anschließend Endpunkt erstellen aus. In der folgenden Tabelle sind die Einstellungen beschrieben.

Option	Vorgehensweise
Endpunkttyp	Wählen Sie den Ziel-Endpunkttyp aus.

Option	Vorgehensweise
Endpoint identifizier	Geben Sie den Namen Ihres Endpunkts ein. Beziehen Sie beispielsweise den Endpunkttyp in den Namen ein (z. B. my-redis-target).
Ziel-Engine	Wählen Sie Redis als Typ der Datenbank-Engine aus, zu der dieser Endpunkt eine Verbindung herstellen soll.
Cluster name	Geben Sie den DNS-Namen Ihres Redis-Clusters ein.
Port	Geben Sie die Portnummer Ihres Redis-Clusters ein.
SSL-Sicherheitsprotokoll	<p>Wählen Sie entweder Nur-Text oder SSL-Verschlüsselung aus.</p> <p>Nur-Text – Diese Option bietet keine Transport Layer Security (TLS)-Verschlüsselung für den Datenverkehr zwischen Endpunkt und Datenbank.</p> <p>SSL-Verschlüsselung – Geben Sie bei Auswahl dieser Option einen ARN für ein SSL-Zertifikat der Zertifizierungsstelle (CA) ein, um das Serverzertifikat zu überprüfen und eine verschlüsselte Verbindung herzustellen.</p> <p>Für On-Premises-Redis unterstützt DMS sowohl öffentliche als auch private Zertifizierungsstellen (CA). Für ElastiCache für Redis unterstützt DMS nur eine öffentliche Zertifizierungsstelle.</p>

Option	Vorgehensweise
Authentifizierungstyp	<p>Wählen Sie die Art der Authentifizierung aus, die bei einer Verbindung zu Redis durchgeführt werden soll. Mögliche Optionen sind: Keine, Authentifizierungsrolle und Authentifizierungs-Token.</p> <p>Geben Sie bei Auswahl von „Authentifizierungsrolle“ einen Benutzernamen für die Authentifizierung und ein Passwort für die Authentifizierung an.</p> <p>Geben Sie bei Auswahl von „Authentifizierungs-Token“ nur ein Passwort für die Authentifizierung an.</p>
Replication instance	<p>[Optional] Nur wenn Sie Ihre Verbindung testen möchten, wählen Sie hier den Namen der Replikations-Instance, die Sie zuvor auf der Seite Replikations-Instance erstellen eingegeben haben.</p>

Wenn Sie alle Informationen für Ihren Endpunkt bereitgestellt haben, erstellt AWS DMS Ihren Redis-Zielendpunkt, der bei der Datenbankmigration verwendet werden soll.

Informationen zum Erstellen einer Migrationsaufgabe und zum Starten Ihrer Datenbankmigration finden Sie unter [Erstellen einer Aufgabe](#).

Angeben von Endpunkteinstellungen für Redis als Ziel

Um einen Zielendpunkt zu erstellen oder zu ändern, können Sie die Konsole oder die `CreateEndpoint`- oder `ModifyEndpoint`-API-Vorgänge verwenden.

Geben Sie für ein Redis-Ziel in der AWS DMS-Konsole Endpunktspezifische Einstellungen auf der Konsolenseite `Endpunkt erstellen` oder `Endpunkt ändern` an.

Wenn Sie die API-Operationen `CreateEndpoint` und `ModifyEndpoint` verwenden, geben Sie Anforderungsparameter für die Option `RedisSettings` an. Das folgende Beispiel zeigt, wie dies über die AWS CLI geschehen kann.

```
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"
    }
  }
}
```

Die Parameter für `--redis-settings` lauten wie folgt:

- `ServerName` – (Erforderlich), Typ `string`, gibt den Redis-Cluster an, zu dem Daten migriert werden, und befindet sich in derselben VPC.
- `Port` – (Erforderlich), Typ `number`, der Portwert, der für den Zugriff auf den Endpunkt verwendet wird.
- `SslSecurityProtocol` – (Optional), gültige Werte sind `plaintext` und `ssl-encryption`. Der Standardwert ist `ssl-encryption`.

Die `plaintext`-Option bietet keine Transport Layer Security (TLS)-Verschlüsselung für den Datenverkehr zwischen Endpunkt und Datenbank.

Verwenden Sie `ssl-encryption`, um eine verschlüsselte Verbindung herzustellen. `ssl-encryption` benötigt keinen ARN der SSL-Zertifizierungsstelle (CA), um das Zertifikat eines

Servers zu verifizieren, über die Einstellung `SslCaCertificateArn` kann jedoch optional einer angegeben werden. Wenn kein ARN für eine Zertifizierungsstelle angegeben wird, verwendet DMS die Amazon-Stammzertifizierungsstelle.

Bei Verwendung eines On-Premises-Redis-Ziels können Sie `SslCaCertificateArn` verwenden, um eine öffentliche oder private Zertifizierungsstelle (CA) in DMS zu importieren, und diesen ARN für die Serverauthentifizierung angeben. Bei Verwendung von ElastiCache für Redis als Ziel wird eine private Zertifizierungsstelle (CA) nicht unterstützt.

- `AuthType` – (Erforderlich), gibt an, welche Art von Authentifizierung bei der Verbindung zu Redis durchgeführt werden soll. Gültige Werte sind: `none`, `auth-token` und `auth-role`.

Bei Verwendung der Option `auth-token` muss ein *Passwort für die Authentifizierung* angegeben werden, während bei Verwendung der Option `auth-role` ein *Benutzername für die Authentifizierung* und ein *Passwort für die Authentifizierung* erforderlich sind.

Verwenden von Babelfish als Ziel für AWS Database Migration Service

Mit AWS Database Migration Service können Sie Daten von einer Quelldatenbank in Microsoft SQL Server zu einem Babelfish-Ziel migrieren.

Babelfish for Aurora PostgreSQL erweitert Ihre Amazon Aurora PostgreSQL kompatible Edition-Datenbank um die Möglichkeit, Datenbankverbindungen von Microsoft SQL Server-Clients zu akzeptieren. Dadurch können Anwendungen, die ursprünglich für SQL Server entwickelt wurden, direkt mit Aurora PostgreSQL arbeiten – mit nur wenigen Codeänderungen im Vergleich zu einer herkömmlichen Migration und ohne Änderung der Datenbanktreiber.

Informationen zu den Versionen von Babelfish, die AWS DMS als Ziel unterstützt, finden Sie unter [Ziele für AWS DMS](#). Frühere Versionen von Babelfish in Aurora PostgreSQL erfordern ein Upgrade, damit der Babelfish-Endpunkt verwendet werden kann.

Note

Der Aurora-PostgreSQL-Zielendpunkt ist die bevorzugte Methode für die Migration von Daten zu Babelfish. Weitere Informationen finden Sie unter [Verwenden von Babelfish für Aurora PostgreSQL als Ziel](#).

Informationen zur Verwendung von Babelfish als Datenbank-Endpunkt finden Sie unter [Babelfish for Aurora PostgreSQL](#) im Benutzerhandbuch für Amazon Aurora.

Voraussetzungen für die Verwendung von Babelfish als Ziel für AWS DMS

Sie müssen Ihre Tabellen vor der Migration von Daten erstellen, um sicherzustellen, dass AWS DMS die richtigen Datentypen und Tabellenmetadaten verwendet. Wenn Sie die Tabellen nicht auf dem Ziel erstellen, bevor Sie die Migration ausführen, erstellt AWS DMS die Tabellen möglicherweise mit falschen Datentypen und Berechtigungen. Beispielsweise erstellt AWS DMS eine Zeitstempelspalte stattdessen als binary(8) und bietet nicht die erwartete timestamp/rowversion-Funktionalität.

So können Sie Ihre Tabellen vor der Migration vorbereiten und erstellen

1. Führen Sie Create-Table-DDL-Anweisungen aus, die alle eindeutigen Einschränkungen, Primärschlüssel oder Standardeinschränkungen enthalten.

Fügen Sie keine Einschränkungen hinsichtlich Fremdschlüsseln und keine DDL-Anweisungen für Objekte wie Ansichten, gespeicherte Prozeduren, Funktionen oder Auslöser hinzu. Sie können sie nach der Migration Ihrer Quelldatenbank anwenden.

2. Identifizieren Sie alle Identitätsspalten, berechneten Spalten oder Spalten mit den Datentypen rowversion oder timestamp für Ihre Tabellen. Erstellen Sie anschließend die erforderlichen Transformationsregeln, um bekannte Probleme bei der Ausführung der Migrationsaufgabe zu beheben. Weitere Informationen finden Sie unter [Transformationsregeln und Aktionen](#).
3. Identifizieren Sie Spalten mit Datentypen, die Babelfish nicht unterstützt. Ändern Sie dann die betroffenen Spalten in der Zieltabelle, sodass sie unterstützte Datentypen verwenden, oder erstellen Sie eine Transformationsregel, durch die sie während der Migrationsaufgabe entfernt werden. Weitere Informationen finden Sie unter [Transformationsregeln und Aktionen](#).

In der folgenden Tabelle sind Quelldatentypen, die von Babelfish nicht unterstützt werden, und der entsprechende empfohlene Zieldatentyp aufgeführt.

Quelldatentyp	Empfohlener Babelfish-Datentyp
HEIRARCHYID	NVARCHAR(250)
GEOMETRY	VARCHAR(MAX)
GEOGRAPHY	VARCHAR(MAX)

So legen Sie Aurora-Kapazitätseinheiten (Aurora Capacity Units, ACUs) für Ihre Quelldatenbank von Aurora PostgreSQL Serverless V2 fest

Sie können die Leistung Ihrer AWS DMS-Migrationsaufgabe vor der Ausführung verbessern, indem Sie den ACU-Mindestwert festlegen.

- Legen Sie im Fenster mit den Kapazitätseinstellungen für Serverless v2 die Option Minimum-ACUs auf 2 oder einen für Ihren Aurora-DB-Cluster angemessenen Wert fest.

Weitere Informationen zum Festlegen von Aurora-Kapazitätseinheiten finden Sie unter [Auswählen des Aurora-Serverless-v2-Kapazitätsbereichs für einen Aurora-Cluster](#) im Benutzerhandbuch für Amazon Aurora.

Nach der Ausführung Ihrer AWS DMS-Migrationsaufgabe können Sie den Mindestwert der ACUs auf einen für Ihre Quelldatenbank von Aurora PostgreSQL Serverless V2 angemessenen Wert zurücksetzen.

Sicherheitsanforderungen bei Verwendung von Babelfish als Ziel für AWS Database Migration Service

Im folgenden Abschnitt werden die Sicherheitsanforderungen bei Verwendung von AWS DMS mit einem Babelfish-Ziel beschrieben:

- Der Name des Administrator-Benutzers (Admin-Benutzer), mit dem die Datenbank erstellt wurde.
- PSQL-Login und -Benutzer mit den ausreichenden SELECT-, INSERT-, UPDATE-, DELETE- und REFERENCES-Berechtigungen.

Benutzerberechtigungen für die Verwendung von Babelfish als Ziel für AWS DMS

Important

Aus Sicherheitsgründen muss das für die Datenmigration verwendete Benutzerkonto ein registrierter Benutzer in einer Babelfish-Zieldatenbank sein.

Ihr Babelfish-Zielendpunkt erfordert Mindestbenutzerberechtigungen, um eine AWS DMS-Migration ausführen zu können.

So erstellen Sie einen Login und einen Transact-SQL (T-SQL)-Benutzer mit geringen Berechtigungen

1. Erstellen Sie einen Login und ein Passwort für die Verbindung mit dem Server.

```
CREATE LOGIN dms_user WITH PASSWORD = 'password';  
GO
```

2. Erstellen Sie die virtuelle Datenbank für Ihren Babelfish-Cluster.

```
CREATE DATABASE my_database;  
GO
```

3. Erstellen Sie den T-SQL-Benutzer für Ihre Zieldatenbank.

```
USE my_database  
GO  
CREATE USER dms_user FOR LOGIN dms_user;  
GO
```

4. Erteilen Sie für jede Tabelle in Ihrer Babelfish-Datenbank GRANT-Berechtigungen für die Tabellen.

```
GRANT SELECT, DELETE, INSERT, REFERENCES, UPDATE ON [dbo].[Categories] TO dms_user;
```

Einschränkungen bei Verwendung von Babelfish als Ziel für AWS Database Migration Service

Die folgenden Einschränkungen gelten, wenn Sie eine Babelfish-Datenbank als Ziel für AWS DMS verwenden:

- Es wird nur der Tabellenvorbereitungsmodus „Nichts unternehmen“ unterstützt.
- Der Datentyp ROWVERSION erfordert eine Tabellenzuordnungsregel, die den Spaltennamen während der Migrationsaufgabe aus der Tabelle entfernt.
- Der Datentyp sql_variant wird nicht unterstützt.
- Der vollständige LOB-Modus wird unterstützt. Für die Verwendung von SQL Server als Quellendpunkt muss die Einstellung ForceFullLob=True für das SQL-Server-Endpunkt-Verbindungsattribut festgelegt werden, damit LOBs zum Zielendpunkt migriert werden können.

- Für Einstellungen von Replikationsaufgaben gelten die folgenden Einschränkungen:

```
{
  "FullLoadSettings": {
    "TargetTablePrepMode": "DO_NOTHING",
    "CreatePkAfterFullLoad": false,
  }.
}
```

- Die Datentypen TIME(7), DATETIME2(7) und DATETIMEOFFSET(7) in Babelfish begrenzen den Präzisionswert für den Sekundenanteil der Zeit auf 6 Ziffern. Ziehen Sie die Verwendung des Genauigkeitswerts 6 für Ihre Zieltabelle in Betracht, wenn Sie diese Datentypen verwenden. Bei den Babelfish-Versionen 2.2.0 und höher ist die siebte Stelle der Präzision bei Verwendung von TIME(7) und DATETIME2(7) immer Null.
- Im Modus DO_NOTHING prüft DMS, ob die Tabelle bereits vorhanden ist. Wenn die Tabelle nicht im Zielschema vorhanden ist, erstellt DMS die Tabelle auf Grundlage der Quelltabellendefinition und ordnet alle benutzerdefinierten Datentypen ihrem Basisdatentyp zu.
- Eine AWS DMS-Migrationsaufgabe zu einem Babelfish-Ziel unterstützt keine Tabellen mit Spalten mit den Datentypen ROWVERSION oder TIMESTAMP. Sie können eine Tabellenzuordnungsregel verwenden, die den Spaltennamen während der Übertragung aus der Tabelle entfernt. Im folgenden Beispiel für eine Transformationsregel wird die Tabelle mit dem Namen Actor in Ihrer Quelle so transformiert, dass alle Spalten, die mit den Zeichen col beginnen, aus der Tabelle Actor in Ihrem Ziel entfernt werden.

```
{
  "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%"
}
}]
}

```

- Für Tabellen mit Identitäts- oder berechneten Spalten, bei denen die Zieltabellen Namen mit gemischter Groß- und Kleinschreibung wie Kategorien verwenden, müssen Sie eine Transformationsregelaktion erstellen, die die Tabellennamen für Ihre DMS-Aufgabe in Kleinbuchstaben umwandelt. Das folgende Beispiel zeigt, wie Sie die Transformationsregelaktion in Kleinbuchstaben umwandeln mithilfe der AWS DMS-Konsole erstellen. Weitere Informationen finden Sie unter [Transformationsregeln und Aktionen](#).

▼ 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 ▼

- Vor Babelfish Version 2.2.0 beschränkte DMS die Anzahl der Spalten, die Sie auf einen Babelfish-Zielendpunkt replizieren konnten, auf zwanzig (20) Spalten. Mit Babelfish 2.2.0 wurde das Limit auf 100 Spalten erhöht. Mit den Babelfish-Versionen 2.4.0 und höher wird die Anzahl der Spalten, die Sie replizieren können, erneut erhöht. Sie können das folgende Codebeispiel für Ihre SQL-Server-Datenbank ausführen, um zu ermitteln, welche Tabellen zu lang sind.

```

USE myDB;
GO
DECLARE @Babelfish_version_string_limit INT = 8000; -- Use 380 for Babelfish versions
  before 2.2.0
WITH bfendpoint
AS (
SELECT
  [TABLE_SCHEMA]
    , [TABLE_NAME]
    , COUNT( [COLUMN_NAME] ) AS NumberColumns
    , ( SUM( LEN( [COLUMN_NAME] ) + 3 )
  + SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
    + LEN( TABLE_SCHEMA ) + 3
  + 12 -- INSERT INTO string
  + 12) AS InsertIntoCommandLength -- values string
    , CASE WHEN ( SUM( LEN( [COLUMN_NAME] ) + 3 )
  + SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
    + LEN( TABLE_SCHEMA ) + 3
  + 12 -- INSERT INTO string
  + 12) -- values string
  >= @Babelfish_version_string_limit
  THEN 1
  ELSE 0
  END AS IsTooLong
FROM [INFORMATION_SCHEMA].[COLUMNS]
GROUP BY [TABLE_SCHEMA], [TABLE_NAME]
)
SELECT *
FROM bfendpoint
WHERE IsTooLong = 1
ORDER BY TABLE_SCHEMA, InsertIntoCommandLength DESC, TABLE_NAME
;

```

Zieldatentypen für Babelfish

In der folgenden Tabelle sind die bei Verwendung von AWS DMS unterstützten Babelfish-Zieldatentypen sowie die Standardzuordnung von AWS DMS-Datentypen aufgeführt.

Weitere Informationen zu AWS DMS-Datentypen finden Sie unter [Datentypen für den AWS Database Migration Service](#).

AWS DMS-Datentyp	Babelfish-Datentyp
BOOLEAN	TINYINT
BYTES	VARBINARY (Länge)
DATUM	DATUM
TIME	TIME
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	FLOAT
STRING	<p>Wenn die Spalte eine Datums- oder Zeitspalte ist, gehen Sie wie folgt vor:</p> <ul style="list-style-type: none"> • Verwenden Sie für SQL Server 2008 und höher DATETIME2. • Verwenden Sie für frühere Versionen, wenn die Skalierung 3 oder weniger ist, DATETIME. Verwenden Sie in allen anderen Fällen VARCHAR (37).

AWS DMS-Datentyp	Babelfish-Datentyp
	Wenn die Spalte keine Datums- oder Zeitspalte ist, verwenden Sie VARCHAR (Länge).
UINT1	TINYINT
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR(Länge)
BLOB	<p>VARBINARY(max)</p> <p>Wenn Sie diesen Datentyp mit DMS verwenden möchten, müssen Sie die Verwendung von BLOBs für eine bestimmte Aufgabe aktivieren. DMS unterstützt BLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>
CLOB	<p>VARCHAR(max)</p> <p>Wenn Sie diesen Datentyp mit DMS verwenden möchten, müssen Sie die Verwendung von CLOBs für eine bestimmte Aufgabe aktivieren.</p>
NCLOB	<p>NVARCHAR(max)</p> <p>Wenn Sie diesen Datentyp mit DMS verwenden möchten, müssen Sie die Verwendung von NCLOBs für eine bestimmte Aufgabe aktivieren. Bei CDC unterstützt DMS NCLOB-Datentypen nur in Tabellen, die einen Primärschlüssel enthalten.</p>

Verwenden von Amazon Timestream als Ziel für AWS Database Migration Service

Sie können AWS Database Migration Service verwenden, um Daten von Ihrer Quelldatenbank zu einem Amazon-Timestream-Zielendpunkt zu migrieren, wobei Full-Load- und CDC-Datenmigrationen unterstützt werden.

Amazon Timestream ist ein schneller, skalierbarer Serverless-Zeitreihendatenbank-Service, der für die Erfassung großer Datenmengen entwickelt wurde. Zeitreihendaten sind Folgen von Datenpunkten, die über ein Zeitintervall erfasst wurden. Sie werden zur Messung von Ereignissen verwendet, die sich im Laufe der Zeit ändern. Es wird verwendet, um Metriken aus IoT-Anwendungen, DevOps Anwendungen und Analyseanwendungen zu sammeln, zu speichern und zu analysieren. Sobald Sie Ihre Daten in Timestream haben, können Sie Trends und Muster in Ihren Daten nahezu in Echtzeit visualisieren und identifizieren. Informationen zu Amazon Timestream finden Sie unter [Was ist Amazon Timestream?](#) im Amazon-Timestream-Entwicklerhandbuch.

Themen

- [Voraussetzungen für die Verwendung von Amazon Timestream als Ziel für AWS Database Migration Service](#)
- [Aufgabeneinstellungen für vollständige Multithread-Ladevorgänge](#)
- [Aufgabeneinstellungen für Multithreaded CDC-Ladevorgänge](#)
- [Endpunkteinstellungen bei der Verwendung von Timestream als Ziel für AWS DMS](#)
- [Erstellen und Ändern eines Amazon-Timestream-Zielendpunkts](#)
- [Verwenden der Objektzuweisung zum Migrieren von Daten zu einem Timestream-Thema](#)
- [Einschränkungen bei der Verwendung von Amazon Timestream als Ziel für AWS Database Migration Service](#)

Voraussetzungen für die Verwendung von Amazon Timestream als Ziel für AWS Database Migration Service

Bevor Sie Amazon Timestream als Ziel für AWS DMS einrichten, stellen Sie sicher, dass Sie eine IAM-Rolle erstellt haben. Diese Rolle muss AWS DMS den Zugriff auf die Daten ermöglichen, die zu Amazon Timestream migriert werden. Die Mindestanzahl an Zugriffsberechtigungen für die Rolle, die Sie für die Migration zu Timestream verwenden, ist in der folgenden IAM-Richtlinie aufgeführt.

```
{
```

```

"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"
  }
]
}

```

Wenn Sie alle Tabellen migrieren möchten, verwenden Sie im obigen Beispiel * für *TABLE_NAME*.

Beachten Sie Folgendes zur Verwendung von Timestream als Ziel:

- Wenn Sie historische Daten mit Zeitstempeln aufnehmen möchten, die älter als ein Jahr sind, empfehlen wir, die Daten mit AWS DMS im .csv-Format in Amazon S3 zu schreiben. Verwenden Sie dann den Stapelladevorgang von Timestream, um die Daten in Timestream aufzunehmen.

Weitere Informationen dazu finden Sie unter [Verwenden des Stapelladens in Timestream](#) im [Amazon-Timestream-Entwicklerhandbuch](#).

- Für Volllast-Datenmigrationen von Daten, die weniger als ein Jahr alt sind, empfehlen wir, die Aufbewahrungsdauer der Timestream-Tabelle für den Speicher größer oder gleich dem ältesten Zeitstempel festzulegen. Sobald die Migration abgeschlossen ist, bearbeiten Sie die Speicherdauer der Tabelle zum gewünschten Wert. Gehen Sie beispielsweise wie folgt vor, um Daten zu migrieren, bei denen der älteste Zeitstempel 2 Monate alt ist:
 - Legen Sie die Speicherdauer der Timestream-Zieltabelle auf 2 Monate fest.
 - Starten Sie die Datenmigration mit AWS DMS.
 - Sobald die Datenmigration abgeschlossen ist, ändern Sie den Aufbewahrungszeitraum der Timestream-Zieltabelle auf den gewünschten Wert.

Wir empfehlen, die Speicherkosten vor der Migration anhand der Informationen auf den folgenden Seiten abzuschätzen:

- [Amazon-Timestream-Preise](#)
- [AWS-Preisrechner](#)
- Für CDC-Datenmigrationen empfehlen wir, den Aufbewahrungszeitraum für den Speicher der Zieltabelle so festzulegen, dass die aufgenommenen Daten innerhalb der Aufbewahrungsgrenzen des Speichers liegen. Weitere Informationen dazu finden Sie unter [Bewährte Verfahren für Schreibvorgänge](#) im [Amazon-Timestream-Entwicklerhandbuch](#).

Aufgabeneinstellungen für vollständige Multithread-Ladevorgänge

AWS DMS unterstützt eine Multithread-Migrationsaufgabe mit Volllast zu einem Timestream-Zielendpunkt mit folgenden Aufgabeneinstellungen, um die Geschwindigkeit der Datenübertragung zu erhöhen:

- `MaxFullLoadSubTasks` – Geben Sie diese Option an, um die maximale Anzahl von Quelltabellen festzulegen, die parallel geladen werden sollen. DMS lädt jede Tabelle in die entsprechende Amazon-Timestream-Zieltabelle mithilfe einer dedizierten Unteraufgabe. Der Standardwert beträgt 8; der Maximalwert beträgt 49.
- `ParallelLoadThreads` – Verwenden Sie diese Option, um die Anzahl der Threads anzugeben, die AWS DMS zum Laden der einzelnen Tabellen in ihre Amazon-Timestream-Zieltabelle verwendet. Der maximale Wert für ein Timestream-Ziel ist 32. Sie können eine Erhöhung dieses Höchstwerts anfordern.

- `ParallelLoadBufferSize` – Verwenden Sie diese Option, um die maximale Anzahl der Datensätze anzugeben, die in dem Puffer gespeichert werden sollen, den die Parallellade-Threads zum Laden von Daten in das Amazon-Timestream-Ziel verwenden. Der Standardwert lautet 50. Die maximale Wert ist 1.000. Verwenden Sie diese Einstellung mit `ParallelLoadThreads`; `ParallelLoadBufferSize` ist nur gültig, wenn es mehr als einen Thread gibt.
- `ParallelLoadQueuesPerThread` – Verwenden Sie diese Option, um die Anzahl der Warteschlangen anzugeben, auf die jeder gleichzeitige Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und eine Stapellast für das Ziel zu generieren. Der Standardwert ist 1. Für Amazon-Timestream-Ziele mit verschiedenen Nutzlastgrößen beträgt der gültige Bereich jedoch 5–512 Warteschlangen pro Thread.

Aufgabeneinstellungen für Multithreaded CDC-Ladevorgänge

Zur Steigerung der CDC-Leistung werden folgende Aufgabeneinstellungen von AWS DMS unterstützt:

- `ParallelApplyThreads` – Gibt die Anzahl der gleichzeitigen Threads an, die AWS DMS während eines CDC-Ladevorgangs verwendet, um Datensätze an einen Timestream-Zielendpunkt zu übertragen. Der Standardwert ist 0 und der maximale Wert ist 32.
- `ParallelApplyBufferSize` – Gibt die maximale Anzahl von Datensätzen an, die in jeder Pufferwarteschlange für gleichzeitige Threads gespeichert werden sollen, um sie während eines CDC-Ladevorgangs an einen Timestream-Zielendpunkt zu übertragen. Der Standardwert ist 100 und der Höchstwert 1 000. Verwenden Sie diese Option, wenn `ParallelApplyThreads` mehrere Threads angibt.
- `ParallelApplyQueuesPerThread` – Gibt die Anzahl der Warteschlangen an, auf die jeder Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und während des CDC eine Stapellast für einen Timestream-Endpunkt zu generieren. Der Standardwert ist 1 und der maximale Wert 512.

Endpunkteinstellungen bei der Verwendung von Timestream als Ziel für AWS DMS

Sie können Endpunkteinstellungen zur Konfiguration Ihrer Timestream-Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen beim Erstellen des Zielendpunkts über die AWS DMS-Konsole oder über den Befehl `create-endpoint` in der [AWS CLI](#) mit der JSON-Syntax `--timestream-settings '{"EndpointSetting": "value", ...}'` an.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Timestream als Ziel verwenden können.

Name	Beschreibung
MemoryDuration	<p>Legen Sie dieses Attribut so fest, dass die Aufbewahrungsfrist angegeben wird, mit der die migrierten Daten im Speicher von Timestream gespeichert werden sollen. Die Zeit wird in Stunden gemessen. Der Speicher von Timestream ist für einen hohen Aufnahmedurchsatz und schnellen Zugriff optimiert.</p> <p>Standardwert: 24 (Stunden)</p> <p>Gültige Werte: 1 bis 8.736 (1 Stunde bis 12 Monate, gemessen in Stunden)</p> <p>Beispiel: <code>--timestream-settings '{"MemoryDuration": 20}'</code></p>
DatabaseName	<p>Legen Sie dieses Attribut fest, um den Namen der Timestream-Zieldatenbank anzugeben.</p> <p>Typ: Zeichenfolge</p> <p>Beispiel: <code>--timestream-settings '{"DatabaseName": "db_name"}</code></p>
TableName	<p>Legen Sie dieses Attribut fest, um den Namen der Timestream-Zieltabelle anzugeben.</p> <p>Typ: Zeichenfolge</p> <p>Beispiel: <code>--timestream-settings '{"TableName": "table_name"}</code></p>
MagneticDuration	<p>Legen Sie dieses Attribut fest, um die magnetische Dauer, die auf die Timestream-Tabellen angewendet wird, in Tagen anzugeben. Dies ist die Aufbewahrungsgrenze für die erfassten Daten. Timestream löscht</p>

Name	Beschreibung
	<p>jeden Zeitstempel, der die Aufbewahrungsgrenze überschreitet. Weitere Informationen dazu finden Sie unter Speicher im Amazon-Timestream-Entwicklerhandbuch.</p> <p>Beispiel: <code>--timestream-settings '{"MagneticDuration": "3"}'</code></p>
CdcInsertsAndUpdates	<p>Legen Sie dieses Attribut auf <code>true</code> fest, um anzugeben, dass AWS DMS nur Einfügungen und Updates und keine Löschungen anwendet. Timestream erlaubt das Löschen von Datensätzen nicht. Wenn dieser Wert also <code>false</code> ist, setzt AWS DMS den entsprechende Datensatz in der Timestream-Datenbank auf Null, anstatt ihn zu löschen. Weitere Informationen finden Sie unter Einschränkungen.</p> <p>Standardwert: <code>false</code></p> <p>Beispiel: <code>--timestream-settings '{"CdcInsertsAndUpdates": "true"}'</code></p>
EnableMagneticStoreWrites	<p>Setzen Sie dieses Attribut auf <code>true</code> um Magnetspeicher-Schreibvorgänge zu aktivieren. Wenn dieser Wert <code>false</code> ist, schreibt AWS DMS keine Datensätze, deren Zeitstempel älter als die Speicherdauer der Zieltabelle ist, da Timestream standardmäßig keine Magnetspeicher-Schreibvorgänge zulässt. Weitere Informationen dazu finden Sie unter Bewährte Verfahren für Schreibvorgänge im Amazon-Timestream-Entwicklerhandbuch.</p> <p>Standardwert: <code>false</code></p> <p>Beispiel: <code>--timestream-settings '{"EnableMagneticStoreWrites": "true"}'</code></p>

Erstellen und Ändern eines Amazon-Timestream-Zielendpunkts

Nachdem Sie eine IAM-Rolle erstellt und die Mindestanzahl an Zugriffsberechtigungen festgelegt haben, können Sie einen Amazon-Timestream-Zielendpunkt mithilfe der AWS DMS-Konsole oder mithilfe des `create-endpoint`-Befehls in der [AWS CLI](#) mit der `--timestream-settings` `'{"EndpointSetting": "value", ...}'`-JSON-Syntax erstellen.

In den folgenden Beispielen wird erläutert, wie ein Timestream-Zielendpunkt mit der AWS CLI erstellt und geändert wird.

Befehl „Timestream-Zielendpunkt erstellen“

```
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,
}
```

Befehl „Timestream-Zielendpunkt ändern“

```
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,
}
```

Verwenden der Objektzuweisung zum Migrieren von Daten zu einem Timestream-Thema

AWS DMS verwendet Tabellenzuweisungsregeln, um Daten aus der Quelle dem Timestream-Zielthema zuzuweisen. Um Daten einem Zielthema zuzuweisen, verwenden Sie eine Art von Tabellenzuweisungsregel, die als Objektzuweisung bezeichnet wird. Mit der Objektzuweisung legen

Sie fest, wie Datensätze in der Quelle den in einem Timestream-Thema veröffentlichten Datensätzen zugewiesen werden.

Timestream-Themen verfügen bis auf einen Partitionsschlüssel über keine voreingestellte Struktur.

Note

Sie müssen die Objektzuweisung nicht verwenden. Sie können die reguläre Tabellenzuweisung für verschiedene Transformationen verwenden. Der Partitionsschlüsseltyp folgt jedoch diesen Standardverhaltensweisen:

- Der Primärschlüssel wird als Partitionsschlüssel für Volllast verwendet.
- Wenn keine Aufgabeneinstellungen für die parallele Anwendung verwendet werden, wird `schema.table` als Partitionsschlüssel für CDC verwendet.
- Wenn Aufgabeneinstellungen für parallele Anwendung verwendet werden, wird der Primärschlüssel als Partitionsschlüssel für CDC verwendet.

Um eine Objektzuweisungsregel zu erstellen, legen Sie `rule-type` als `object-mapping` fest. Diese Regel gibt an, welchen Objektzuweisungstyp Sie verwenden möchten. Die Struktur für die Regel lautet wie folgt.

```
{
  "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 unterstützt derzeit `map-record-to-record` und `map-record-to-document` als einzige gültige Werte für den `rule-action`-Parameter. Die Werte `map-record-to-record` und `map-record-to-document` geben an, wie AWS DMS standardmäßig Datensätze handhabt, die nicht als Teil der `exclude-columns`-Attributliste nicht ausgeschlossen werden. Diese Werte wirken sich in keiner Weise auf die Attributzuweisungen aus.

Verwenden Sie `map-record-to-record` beim Migrieren aus einer relationalen Datenbank zu einem Timestream-Thema. Dieser Regeltyp verwendet den `taskResourceId.schemaName.tableName`-Wert aus der relationalen Datenbank als Partitionsschlüssel in dem Timestream-Thema und erstellt ein Attribut für jede Spalte in der Quelldatenbank. Wenn Sie `map-record-to-record` für eine Spalte in der Quelltable verwenden, die nicht in der `exclude-columns`-Attributliste aufgeführt ist, erstellt AWS DMS ein entsprechendes Attribut im Zielthema. Dieses entsprechende Attribut wird unabhängig davon erstellt, ob diese Quellspalte in einer Attributzuweisung verwendet wird oder nicht.

Eine Möglichkeit, `map-record-to-record` zu verstehen, besteht darin, sich die praktische Anwendung zu veranschaulichen. In diesem Beispiel wird davon ausgegangen, dass Sie mit einer Tabellenzeile einer relationalen Datenbank beginnen, die die folgende Struktur aufweist und die folgenden Daten enthält.

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

Um diese Informationen von einem Schema mit dem Namen `Test` zu einem Timestream-Thema zu migrieren, erstellen Sie Regeln, um die Daten dem Zielthema zuzuweisen. Die folgende Regel veranschaulicht die Zuweisung.

```
{
  "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"
      }
    }
  ]
}
```

```
}
```

Nachfolgend wird das resultierende Datensatzformat bei Verwendung unserer Beispieldaten in dem Timestream-Zielthema für ein bestimmtes Timestream-Thema und einen bestimmten Partitionsschlüssel (in diesem Fall `taskResourceId.schemaName.tableName`) illustriert.

```
{
  "FirstName": "Randy",
  "LastName": "Marsh",
  "StoreId": "5",
  "HomeAddress": "221B Baker Street",
  "HomePhone": "1234567890",
  "WorkAddress": "31 Spooner Street, Quahog",
  "WorkPhone": "9876543210",
  "DateOfBirth": "02/29/1988"
}
```

Einschränkungen bei der Verwendung von Amazon Timestream als Ziel für AWS Database Migration Service

Bei der Verwendung von Amazon Timestream als Ziel gelten die folgenden Einschränkungen:

- Dimensionen und Zeitstempel: Timestream verwendet die Dimensionen und Zeitstempel in den Quelldaten wie einen zusammengesetzten Primärschlüssel und ermöglicht Ihnen auch nicht, diese Werte zu ändern. Dies bedeutet: Wenn Sie den Zeitstempel oder die Dimensionen für einen Datensatz in der Quelldatenbank ändern, versucht die Timestream-Datenbank, einen neuen Datensatz zu erstellen. Es ist also möglich, dass, wenn Sie die Dimension oder den Zeitstempel eines Datensatzes so ändern, dass sie denen eines anderen vorhandenen Datensatzes entsprechen, AWS DMS die Werte des anderen Datensatzes aktualisiert, anstatt einen neuen Datensatz zu erstellen oder den vorherigen entsprechenden Datensatz zu aktualisieren.
- DDL-Befehle: Die aktuelle Version von AWS DMS unterstützt nur CREATE TABLE- und DROP TABLE-DDL-Befehle.
- Einschränkungen bei Datensätzen: Timestream hat Einschränkungen für Datensätze wie Datensatzgröße und Maßgröße. Weitere Informationen dazu finden Sie unter [Kontingente](#) im [Amazon-Timestream-Entwicklerhandbuch](#).
- Löschen von Datensätzen und Nullwerten: Timestream unterstützt das Löschen von Datensätzen nicht. AWS DMS löscht die entsprechenden Felder in den Datensätzen in der Timestream-

Zieldatenbank, um die Migration von Datensätzen zu unterstützen, die aus der Quelle gelöscht wurden. AWS DMS ändert die Werte in den Feldern des entsprechenden Zieldatensatzes zu 0 für numerische Felder, Null für Textfelder und false für boolesche Felder.

- Timestream als Ziel unterstützt keine Quellen, die keine relationalen Datenbanken (RDBMS) sind.
- AWS DMS unterstützt Timestream als Ziel nur in den folgenden Regionen:
 - USA Ost (Nord-Virginia)
 - USA Ost (Ohio)
 - USA West (Oregon)
 - Europa (Irland)
 - Europa (Frankfurt)
 - Asien-Pazifik (Sydney)
 - Asien-Pazifik (Tokio)
- Timestream als Ziel unterstützt die Einstellung `TargetTablePrepMode` auf `TRUNCATE_BEFORE_LOAD` nicht. Wir empfehlen, `DROP_AND_CREATE` für diese Einstellung zu verwenden.

Verwendung von Amazon RDS für Db2 und IBM Db2 LUW als Ziel für AWS DMS

Mit AWS Database Migration Service (AWS DMS) können Sie Daten von einer Db2-LUW-Datenbank zu einer Amazon-RDS-für-Db2- oder einer On-Premises-Db2-Datenbank migrieren.

Hinweise zu Versionen von Db2 LUW, die AWS DMS als Ziel unterstützt, finden Sie unter [Ziele für AWS DMS](#).

Sie können Secure Sockets Layer (SSL) verwenden, um Verbindungen zwischen Ihrem Db2 LUW-Endpunkt und der Replikations-Instance zu verschlüsseln. Weitere Informationen zur Verwendung von SSL mit einem Db2-LUW-Endpunkt finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).

Einschränkungen bei der Verwendung von Db2 LUW als Ziel für AWS DMS

Die folgenden Einschränkungen gelten bei der Verwendung einer Db2-LUW-Datenbank als Ziel für AWS DMS. Zu Einschränkungen bei der Verwendung von Db2 LUW als Quelle siehe [Einschränkungen bei der Verwendung von Db2 LUW als Quelle für AWS DMS](#).

- AWS DMS unterstützt Db2 LUW nur dann als Ziel, wenn die Quelle entweder Db2 LUW oder Db2 für z/OS ist.
- Die Verwendung von Db2 LUW als Ziel unterstützt keine Replikationen mit vollständigem LOB-Modus.
- Die Verwendung von Db2 LUW als Ziel unterstützt den XML-Datentyp in der Vollladephase nicht. Dies ist eine Einschränkung des IBM dbload-Dienstprogramms. Weitere Informationen finden Sie unter [Das Hilfsprogramm dbload](#) in der Dokumentation von IBM Informix Server.
- AWS DMS kürzt BLOB-Felder mit Werten, die dem doppelten Anführungszeichen (") entsprechen. Dies ist eine Einschränkung des IBM dbload-Dienstprogramms.

Endpunkteinstellungen bei Verwendung von Db2 LUW als Ziel für AWS DMS

Sie können Endpunkteinstellungen zum Konfigurieren Ihrer Db2-LUW-Zieldatenbank verwenden, ähnlich wie Sie zusätzliche Verbindungsattribute verwenden. Sie geben die Einstellungen beim Erstellen des Zielendpunkts über die AWS DMS-Konsole oder über den Befehl `create-endpoint` in der [AWS CLI](#) mit der JSON-Syntax `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` an.

Die folgende Tabelle zeigt die Endpunkteinstellungen, die Sie mit Db2 LUW als Ziel verwenden können.

Name	Beschreibung
KeepCsvFiles	Wenn der Wert „true“ ist, speichert AWS DMS alle .csv-Dateien, die zur Datenreplikation verwendet wurden, auf dem Db2-LUW-Ziel. DMS verwendet diese Dateien zur Analyse und Problembehandlung..
LoadTimeout	Die Zeitspanne (in Millisekunden), bevor bei AWS DMS ein Timeout für die von DMS auf dem Db2-Ziel ausgeführten Operationen auftritt. Der Standardwert ist 1 200 (20 Minuten).
MaxFileSize	Gibt die maximale Größe (in KB) der .csv-Dateien zum Übertragen von Daten zu Db2 LUW an.

Name	Beschreibung
<code>WriteBufferSize</code>	Die Größe (in KB) des In-Memory-Datei-Schreibpuffers, der beim Generieren von .csv-Dateien auf der lokalen Festplatte in der DMS-Replikations-Instance verwendet wird. Der Standardwert ist 1 024 (1 MB).

Konfigurieren von VPC-Endpunkten als Quell- und Zielendpunkte in AWS DMS

AWS DMS unterstützt Endpunkte von Amazon Virtual Private Cloud (VPC) als Quellen und Ziele. AWS DMS kann eine Verbindung mit jeder AWS-Quell- oder Zieldatenbank mit Amazon-VPC-Endpunkten herstellen, sofern explizit definierte Routen zu diesen Quell- und Zieldatenbanken in ihrer AWS-DMS-VPC definiert sind.

Die Unterstützung von Amazon-VPC-Endpunkten durch AWS DMS ermöglicht eine einfachere Aufrechterhaltung der durchgängigen Netzwerksicherheit für alle Replikationsaufgaben ohne zusätzliche Netzwerkkonfiguration und -einrichtung. Durch die Verwendung von VPC-Endpunkten für alle Quell- und Zielendpunkte wird sichergestellt, dass Ihr gesamter Datenverkehr in Ihrer VPC und unter Ihrer Kontrolle bleibt. Für Upgrades auf AWS-DMS-Version 3.4.7 und höher müssen Sie AWS DMS so konfigurieren, dass VPC-Endpunkte oder öffentliche Routen zu all Ihren Quell- und Zielendpunkten verwendet werden, die mit den folgenden Amazon Web Services interagieren:

- Amazon S3
- Amazon Kinesis
- AWS Secrets Manager
- Amazon DynamoDB
- Amazon Redshift
- Amazon OpenSearch Service

Möglicherweise benötigen Sie VPC-Endpunkte zur Unterstützung von AWS DMS ab Version 3.4.7, wie im Folgenden beschrieben.

Wer ist bei einer Migration auf AWS-DMS-Version 3.4.7 und höher betroffen?

Sie sind betroffen, wenn Sie einen oder mehrere der zuvor aufgeführten AWS-DMS-Endpunkte verwenden und diese Endpunkte nicht öffentlich Routing-fähig sind oder ihnen noch keine VPC-Endpunkte zugeordnet wurden.

Wer ist bei einer Migration auf AWS-DMS-Version 3.4.7 und höher nicht betroffen?

In den folgenden Fällen sind Sie nicht betroffen:

- Sie verwenden keinen der zuvor aufgeführten AWS-DMS-Endpunkte.
- Sie verwenden einen der zuvor aufgeführten Endpunkte und diese sind öffentlich Routing-fähig.
- Sie verwenden einen der zuvor aufgeführten Endpunkte und ihnen wurden VPC-Endpunkte zugeordnet.

Vorbereitung einer Migration auf AWS-DMS-Version 3.4.7 oder höher

Führen Sie vor dem Upgrade von AWS DMS auf Version 3.4.7 oder höher einen der folgenden Schritte aus, um zu verhindern, dass AWS-DMS-Aufgaben bei Verwendung eines der zuvor beschriebenen Endpunkte fehlschlagen:

- Machen Sie die betroffenen AWS-DMS-Endpunkte öffentlich Routing-fähig. Fügen Sie beispielsweise einer beliebigen VPC, die bereits von Ihrer Replikations-Instance von AWS DMS verwendet wird, eine Internet Gateway (IGW)-Route hinzu, um all ihre Quell- und Zielendpunkte öffentlich Routing-fähig zu machen.
- Erstellen Sie VPC-Endpunkte für den Zugriff auf alle Quell- und Zielendpunkte, die von AWS DMS verwendet werden, wie im Folgenden beschrieben.

Stellen Sie sicher, dass alle vorhandenen VPC-Endpunkte, die Sie für Ihre AWS-DMS-Quell- und Zielendpunkte verwenden, eine Vertrauensrichtlinie verwenden, die dem XML-Richtliniendokument `dms-vpc-role` entspricht. Weitere Informationen zu diesem XML-Richtliniendokument finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Konfigurieren Sie andernfalls Ihre Replikations-Instances als VPC-Endpunkte, indem Sie der VPC, die diese enthält, einen VPC-Endpunkt hinzufügen. Wenn Sie Ihre Replikations-Instances ohne

öffentliche Endpunkte konfiguriert haben, werden sie öffentlich zugänglich, indem Sie der VPC, die Ihre Replikations-Instances enthält, einen öffentlich zugänglichen VPC-Endpunkt hinzufügen. Sie müssen nichts weiter tun, um Ihre Replikations-Instances speziell dem VPC-Endpunkt zuzuordnen.

Note

Verschiedene Services haben möglicherweise spezifische VPC-Endpunktkonfigurationen. Wenn Sie beispielsweise AWS Secrets Manager verwenden, müssen Sie die Routing-Tabelle in der Regel nicht anpassen. Überprüfen Sie immer die spezifischen Anforderungen für jeden Service.

Einen VPC-Endpunkt auf der VPC erstellen, die die Replikations-Instances enthält

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie in der Menüleiste der VPC-Konsole dieselbe AWS-Region aus, in der sich die Replikations-Instance von AWS DMS befindet.
3. Wählen Sie im VPC-Navigationsbereich Endpunkte aus.
4. Wählen Sie unter Endpunkte die Option Endpunkt erstellen aus.
5. Optional können Sie ein Name-Tag angeben. Zum Beispiel **my-endpoint-DynamoDB-01**.
6. Wählen Sie unter Services einen Service-Name mit dem Typ Gateway aus (nur für S3 oder DynamoDB).
7. Wählen Sie unter VPC dieselbe VPC, in der sich die Replikations-Instance von AWS DMS befindet, um den Endpunkt zu erstellen.
8. Wählen Sie unter Routing-Tabellen alle verfügbaren Werte für Routing-Tabellen-ID aus.
9. Wählen Sie unter Richtlinie die Option Vollzugriff aus, um die Zugriffskontrolle festzulegen. Wenn Sie ein Tool zur Richtlinienerstellung verwenden möchten, um Ihre eigene Zugriffskontrolle festzulegen, wählen Sie Benutzerdefiniert aus. Verwenden Sie in jedem Fall eine Vertrauensrichtlinie, die dem JSON-Richtliniendokument `dms-vpc-role` entspricht. Weitere Informationen zu diesem Richtliniendokument finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).
10. Vergewissern Sie sich unter Endpunkte, dass der Status des neu erstellten VPC-Endpunkts Verfügbar lautet.

Weitere Informationen zur Konfiguration von VPC-Endpunkten für eine Replikations-Instance von AWS DMS finden Sie unter [Netzwerkkonfigurationen für die Datenbankmigration](#). Weitere Informationen zum Erstellen von Schnittstellen-VPC-Endpunkten für den Zugriff auf AWS-Services im Allgemeinen finden Sie unter [Zugriff auf AWS-Service über einen Schnittstellen-VPC-Endpunkt](#) im Handbuch zu AWS PrivateLink. Informationen zur regionalen Verfügbarkeit von AWS DMS für VPC-Endpunkte finden Sie in der [Tabelle der AWS-Regionen](#).

Von AWS DMS unterstützte DDL-Anweisungen

Sie können DDL-Anweisungen (Data Definition Language) auf der Quelldatenbank während des Migrationsprozesses ausführen. Diese Anweisungen werden vom Replikationsserver in der Zieldatenbank repliziert.

Unterstützte DDL-Anweisungen:

- Create table
- Drop table
- Rename table
- Verkürzen einer Tabelle
- Add column
- Drop column
- Rename column
- Change column data type

DMS erfasst nicht alle unterstützten DDL-Anweisungen für einige Quell-Engine-Typen. Dazu behandelt DMS DDL-Anweisungen unterschiedlich, wenn sie auf bestimmte Ziel-Engines angewendet werden. Informationen darüber, welche DDL-Anweisungen für eine bestimmte Quelle unterstützt werden und wie sie auf ein Ziel angewendet werden, finden Sie im jeweiligen Dokumentationsthema für diesen Quell- und Zielendpunkt.

Mithilfe von Aufgabeneinstellungen können Sie konfigurieren, wie DMS das DDL-Verhalten bei Change Data Capture (CDC) behandelt. Weitere Informationen finden Sie unter [Aufgabeneinstellungen für den Umgang mit der DDL-Änderungsverarbeitung](#).

Arbeiten mit AWS DMS-Aufgaben

Über eine Aufgabe im AWS Database Migration Service (AWS DMS) wird die gesamte Arbeit abgewickelt. Sie geben an, welche Tabellen (oder Ansichten) und Schemas für die Migration verwendet werden sollen. Zudem legen Sie eine spezielle Verarbeitung, z. B. die Protokollierung von Anforderungen, die Steuerung von Tabellendaten und die Fehlerbehandlung, fest.

Eine Aufgabe kann aus drei Hauptphasen bestehen:

- Migration vorhandener Daten (vollständiges Laden)
- Die Anwendung der zwischengespeicherten Änderungen
- Laufende Replikation (Erfassung von Datenänderungen)

Weitere Informationen sowie einen Überblick über die Datenmigration mit Migrationsaufgaben in [Überblick über AWS DMS](#) finden Sie unter AWS DMS

Bei der Erstellung einer Migrationsaufgabe müssen Sie mehrere Dinge beachten:

- Bevor Sie eine Aufgabe erstellen können, müssen Sie einen Quellendpunkt, einen Zielendpunkt und eine Replikations-Instance erstellen.
- Sie können viele Aufgabeneinstellungen festlegen, um Ihre Migrationsaufgabe anzupassen. Sie können diese über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die AWS-API festlegen. Dazu gehören Einstellungen, die die Behandlung von Migrationsfehlern, die Fehlerprotokollierung und Informationen zu Steuerungstabellen betreffen. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).
- Nachdem Sie eine Aufgabe erstellt haben, können Sie sie sofort ausführen. Die Zieltabellen mit den erforderlichen Metadatendefinitionen werden automatisch erstellt und geladen, und Sie können die laufende Replikation angeben.
- Standardmäßig startet AWS DMS Ihre Aufgabe, sobald Sie sie erstellen. In manchen Situationen möchten Sie den Beginn einer Aufgabe jedoch möglicherweise verschieben. Wenn Sie zum Beispiel die AWS CLI verwenden, gibt es ggf. einen Prozess, der eine Aufgabe erstellt, und einen anderen Prozess, der die Aufgabe basierend auf einem auslösenden Ereignis startet. Bei Bedarf können Sie den Beginn Ihrer Aufgabe verschieben.

- Sie können Aufgaben über die Konsole, AWS CLI oder AWS DMS-API überwachen, beenden oder neu starten. Informationen zum Beenden einer Aufgabe mithilfe der AWS DMS-API finden Sie unter [StopReplicationTask](#) in der [API-Referenz zu AWS DMS](#).

Die folgenden Aktionen können Sie bei der Durchführung einer AWS DMS-Aufgabe ausführen.

Aufgabe	Relevante Dokumentation
<p>Erstellen einer Aufgabe</p> <p>Wenn Sie eine Aufgabe erstellen, geben Sie die Quelle, das Ziel und die Replikations-Instance zusammen mit den Migrationseinstellungen an.</p>	<p>Erstellen einer Aufgabe</p>
<p>Erstellen einer laufenden Replikationsaufgabe</p> <p>Sie können eine Aufgabe einrichten, um die fortlaufende Replikation zwischen Quelle und Ziel bereitzustellen.</p>	<p>Erstellen von Aufgaben für die laufende Replikation mit AWS DMS</p>
<p>Anwenden der Aufgabeneinstellungen</p> <p>Jede Aufgabe hat Einstellungen, die entsprechend den Anforderungen Ihrer Datenbankmigration konfiguriert werden können. Sie erstellen diese Einstellungen in einer JSON-Datei oder Sie können einige Einstellungen mithilfe der AWS DMS-Konsole angeben. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie</p>	<p>Angaben von Aufgabeneinstellungen für Aufgaben des AWS Database Migration Service</p>

Aufgabe	Relevante Dokumentation
unter Beispiel für Aufgabeneinstellungen .	
<p>Verwenden der Tabellenzuweisung</p> <p>Die Tabellenzuweisung legt zusätzliche Aufgabeneinstellungen für Tabellen mithilfe verschiedener Regeltypen fest. Mit diesen Regeln können Sie die Datenquelle, das Quellschema, Tabellen und Ansichten, Daten, alle Tabellen- und Daten Transformationen, die während der Aufgabe durchgeführt werden sollen, sowie Einstellungen für die Migration dieser Tabellen und Spalten von der Quelle zum Ziel angeben.</p>	<p>Auswahlregeln</p> <p>Auswahlregeln und Aktionen</p> <p>Transformationsregeln</p> <p>Transformationsregeln und Aktionen</p> <p>Regeln für Tabelleneinstellungen</p> <p>Regeln und Operationen für Tabellen- und Sammlungseinstellungen</p>

Aufgabe	Relevante Dokumentation
<p data-bbox="115 226 652 310">Ausführen von Aufgabenbewertungen vor der Migration</p> <p data-bbox="115 352 634 1150">Sie können Aufgabenbewertungen vor der Migration aktivieren und ausführen, um Fehler bei einer unterstützten Quell- und Zieldatenbank aufzuzeigen, die während einer Migration zu Problemen führen können. Dies kann Fehler wie nicht unterstützte Datentypen, nicht übereinstimmende Indizes und Primärschlüssel sowie andere widersprüchliche Aufgabeneinstellungen umfassen. Diese Bewertungen vor der Migration werden vor der Aufgabenausführung vorgenommen, um potenzielle Fehler zu identifizieren, bevor sie während einer Migration auftreten.</p>	<p data-bbox="695 226 1419 310">Aktivieren und Verwenden von Vormigrationsbewertungen für eine Aufgabe</p>
<p data-bbox="115 1199 354 1234">Datenvalidierung</p> <p data-bbox="115 1276 646 1507">Bei der Datenvalidierung handelt es sich um eine Aufgabeneinstellung, bei der AWS DMS die Daten in Ihrem Zieldatenspeicher mit den Daten in Ihrem Quelldatenspeicher vergleicht.</p>	<p data-bbox="695 1199 1149 1234">Datenvalidierung mit AWS DMS.</p>
<p data-bbox="115 1549 423 1585">Ändern einer Aufgabe</p> <p data-bbox="115 1627 639 1759">Wenn eine Aufgabe angehalten wird, können Sie die Einstellungen für die Aufgabe ändern.</p>	<p data-bbox="695 1549 1003 1585">Ändern einer Aufgabe</p>

Aufgabe	Relevante Dokumentation
<p data-bbox="110 226 500 262">Verschieben einer Aufgabe</p> <p data-bbox="110 306 597 485">Wenn eine Aufgabe angehalten wird, können Sie die Aufgabe auf eine andere Replikations-Instance verschieben.</p>	<p data-bbox="688 226 1078 262">Verschieben einer Aufgabe</p>
<p data-bbox="110 531 532 611">Erneutes Laden von Tabellen während einer Aufgabe</p> <p data-bbox="110 657 597 835">Sie können eine Tabelle während einer Aufgabe neu laden, wenn während einer Aufgabe ein Fehler auftritt.</p>	<p data-bbox="688 531 1451 567">Erneutes Laden von Tabellen während einer Aufgabe</p>
<p data-bbox="110 884 423 919">Anwenden von Filtern</p> <p data-bbox="110 963 646 1377">Sie können Quellfilter verwenden, um Anzahl und Art der Datensätze zu beschränken, die von Ihrer Quelle auf Ihr Ziel übertragen werden. So können Sie beispielsweise festlegen, dass nur die am Hauptsitz ansässigen Angestellten in die Zieldatenbank übertragen werden. Sie wenden Filter auf eine Datenspalte an.</p>	<p data-bbox="688 884 1078 919">Verwenden von Quellfiltern</p>
<p data-bbox="110 1425 505 1461">Überwachen einer Aufgabe</p> <p data-bbox="110 1505 607 1728">Es gibt mehrere Möglichkeiten, um Informationen über die Durchführung einer Aufgabe und die von der Aufgabe verwendeten Tabellen zu erhalten.</p>	<p data-bbox="688 1425 1240 1461">Überwachen von AWS-DMS-Aufgaben</p>

Aufgabe	Relevante Dokumentation
Verwalten von Aufgabenprotokollen Sie können Aufgabenprotokolle über die AWS DMS-API oder die AWS CLI ansehen und löschen.	Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen

Themen

- [Erstellen einer Aufgabe](#)
- [Erstellen von Aufgaben für die laufende Replikation mit AWS DMS](#)
- [Ändern einer Aufgabe](#)
- [Verschieben einer Aufgabe](#)
- [Erneutes Laden von Tabellen während einer Aufgabe](#)
- [Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen](#)
- [Verwenden von Quellfiltern](#)
- [Aktivieren und Verwenden von Vormigrationsbewertungen für eine Aufgabe](#)
- [Angaben zusätzlicher Daten für Aufgabeneinstellungen](#)

Erstellen einer Aufgabe

Gehen Sie wie folgt vor, um eine AWS DMS Migrationsaufgabe zu erstellen:

- Erstellen Sie einen Quellendpunkt, einen Zielendpunkt und eine Replikations-Instance, bevor Sie eine Migrationsaufgabe erstellen.
- Wählen Sie eine Migrationsmethode aus:
 - Migrieren von Daten in die Zieldatenbank – Dieser Prozess erstellt Dateien oder Tabellen in der Zieldatenbank und definiert automatisch die am Ziel erforderlichen Metadaten. Außerdem füllt er die Tabellen mit Daten aus der Quelle aus. Für eine bessere Effizienz werden die Daten parallel aus den Tabellen geladen. Dieser Vorgang ist die Option Bestehende Daten migrieren in der AWS Management Console und wird `Full Load` in der API aufgerufen.
 - Aufzeichnen von Änderungen während der Migration – Dieser Prozess erfasst Änderungen an der Quelldatenbank, die während der Migration der Daten von der Quelle zum Ziel auftreten. Wenn die Migration der ursprünglich angefragten Daten abgeschlossen ist, wendet der

CDC (Change Data Capture)-Prozess die erfassten Änderungen auf die Zieldatenbank an. Änderungen werden erfasst und als Einheiten einzelner durchgeführter Transaktionen angewendet. Sie können mehrere verschiedene Tabellen als Commit einer einzelnen Quelle aktualisieren. Dieser Ansatz garantiert Transaktionsintegrität in der Zieldatenbank. Bei diesem Prozess handelt es sich in der Konsole um die Option `Migrate existing data and replicate ongoing changes` (Vorhandene Daten migrieren und fortlaufende Änderungen replizieren) und in der API um `full-load-and-cdc`.

- Nur Replizieren von Datenänderungen in der Quelldatenbank – Bei diesem Prozess wird die Datei mit dem Wiederherstellungsprotokoll des Quell-DBMS (Datenbankmanagementsystem) gelesen und die Einträge für die einzelnen Transaktionen werden gruppiert. In einigen Fällen AWS DMS können Änderungen nicht innerhalb eines angemessenen Zeitraums auf das Ziel angewendet werden (z. B. wenn das Ziel nicht zugänglich ist). In diesen Fällen werden die AWS DMS Änderungen so lange wie nötig auf dem Replikationsserver zwischengespeichert. Er liest die Quell-DBMS-Protokolle nicht erneut, was viel Zeit in Anspruch nehmen kann. Dieser Prozess entspricht der Option `Nur Replizieren von Datenänderungen` in der AWS DMS -Konsole.
- Bestimmen, wie die Aufgabe große Binärobjekte (LOBs) auf der Quelldatenbank verarbeiten soll. Weitere Informationen finden Sie unter [Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS](#).
- Geben Sie die Migrationsaufgabeneinstellungen an. Dazu gehören die Einrichtung der Protokollierung, Angaben dazu, welche Daten in die Migrationssteuertabelle geschrieben werden, wie Fehler behandelt werden sowie andere Einstellungen. Weitere Informationen zu den Aufgabeneinstellungen finden Sie unter [Angaben von Aufgabeneinstellungen für Aufgaben des AWS Database Migration Service](#).
- Einrichten der Tabellenzuweisung, um Regeln zu definieren, anhand derer die zu migrierenden Daten ausgewählt und gefiltert werden. Weitere Informationen zur Tabellenzuweisung finden Sie unter [Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen](#). Bevor Sie Ihre Zuweisung angeben, sollten Sie den Abschnitt in der Dokumentation zur Zuweisung der Datentypen für Ihre Quell- und Zieldatenbank lesen.
- Aktivieren Sie Vormigrations-Aufgabenbewertungen und führen Sie diese aus, bevor Sie die Aufgabe ausführen. Weitere Informationen zu Vormigrationsbewertungen finden Sie unter [Aktivieren und Verwenden von Vormigrationsbewertungen für eine Aufgabe](#).
- Geben Sie alle erforderlichen zusätzlichen Daten für die Aufgabe an, um Ihre Daten zu migrieren. Weitere Informationen finden Sie unter [Angaben zusätzlicher Daten für Aufgabeneinstellungen](#).

Sie können festlegen, dass eine Aufgabe gestartet wird, sobald Sie die Informationen zu dieser Aufgabe auf der Seite Create task (Aufgabe erstellen) angegeben haben. Alternativ können Sie die Aufgabe auch später von der Dashboard-Seite aus starten.

Im folgenden Verfahren wird davon ausgegangen, dass Sie bereits Informationen und Endpunkte der Replikations-Instance angegeben haben. Weitere Informationen zum Einrichten von Endpunkten finden Sie unter [Erstellen der Quell- und Zielendpunkte](#).

So erstellen Sie eine Migrationsaufgabe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als AWS Identity and Access Management (IAM-) Benutzer angemeldet sind, stellen Sie sicher, dass Sie über die entsprechenden Zugriffsberechtigungen verfügen. AWS DMS Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich Datenbankmigrationsaufgaben und dann Aufgabe erstellen aus.
3. Geben Sie auf der Seite Datenbankmigrationsaufgabe erstellen im Abschnitt Aufgabenkonfiguration die Aufgabenoptionen an. In der folgenden Tabelle sind die Einstellungen beschrieben.

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

Option	Vorgehensweise
Aufgaben-ID	Geben Sie einen Namen für die Aufgabe ein.


Option	Vorgehensweise
Beschreibender Amazon-Resourcenname (ARN) – optional	Ein benutzerfreundlicher Name, um den AWS DMS Standard-ARN zu überschreiben. Sie können diesen Namen nicht mehr ändern, nachdem Sie die Aufgabe erstellt haben.
Replication instance	Zeigt die Replikations-Instance an, die verwendet werden soll.
Quelldatenbankendpunkt	Zeigt den Quellendpunkt an, der verwendet werden soll.
Zielfdatenbankendpunkt	Zeigt den Zielendpunkt an, der verwendet werden soll.
Migration type	Wählen Sie die Migrationsmethode aus, die Sie verwenden möchten. Sie können festlegen, dass nur die vorhandenen Daten in die Zielfdatenbank migriert werden oder zusätzlich zu den migrierten Daten kontinuierlich Änderungen an die Zielfdatenbank gesendet werden.

4. Geben Sie im Abschnitt Aufgabeneinstellungen Werte für die Bearbeitung Ihrer Aufgabe, den Zieltabellen-Vorbereitungsmodus, den Aufgabenstopp, LOB-Einstellungen, Validierung und Protokollierung an.

Option	Vorgehensweise
Bearbeitungsmodus	Wählen Sie aus, ob Sie den Assistenten oder den JSON-Editor verwenden möchten, um Ihre Aufgabeneinstellungen festzulegen. Wenn Sie „Assistent“ auswählen, werden die folgenden Optionen angezeigt.
CDC-Startmodus für Quelltransaktionen	Diese Einstellung wird nur angezeigt, wenn Sie im vorherigen Abschnitt Nur Replizieren von Datenänderungen für Migrationstyp ausgewählt haben. Benutzerdefinierten CDC-Startmodus deaktivieren – Wenn Sie diese Option auswählen, können Sie Ihre

Option	Vorgehensweise
	<p data-bbox="727 212 1442 338">Aufgabe entweder automatisch mithilfe der Option Automatisch bei Erstellung oder manuell über die Konsole starten.</p> <p data-bbox="727 386 1468 562">Benutzerdefinierten CDC-Startmodus aktivieren – Wenn Sie diese Option auswählen, können Sie eine benutzerdefinierte UTC-Startzeit für den Beginn der Verarbeitung von Änderungen angeben.</p>

Option	Vorgehensweise
Target table preparation mode	<p>Diese Einstellung wird nur angezeigt, wenn Sie im vorherigen Abschnitt Migrieren von bestehenden Daten oder Migrieren von bestehenden Daten und Replizieren fortlaufender Änderungen für Migrationstyp ausgewählt haben.</p> <p>Nichts tun — Geht im Modus „Nichts tun“ AWS DMS davon aus, dass die Zieltabellen auf dem Ziel vorab erstellt wurden. Wenn die Tabellen nicht leer sind, können während der Datenmigration Konflikte auftreten und es kann zu einem DMS-Aufgabenfehler kommen. Wenn die Zieltabelle nicht vorhanden ist, wird sie von DMS erstellt. Ihre Tabellenstruktur bleibt unverändert und alle vorhandenen Daten verbleiben in der Tabelle. Der Modus Do nothing (Keine Aktion) bietet sich für CDC-Aufgaben an, wenn die Zieltabellen von der Quelle vorab gefüllt wurden und die laufende Replikation angewendet wird, um Quelle und Ziel zu synchronisieren. Um Tabellen vorab zu erstellen, können Sie AWS Schema Conversion Tool (AWS SCT) verwenden. Weitere Informationen finden Sie unter Installation AWS SCT.</p> <p>Tabellen am Ziel ablegen – Im Modus Tabellen am Ziel ablegen löscht AWS DMS die Zieltabellen und erstellt sie vor dem Start der Migration neu. Dieser Ansatz stellt sicher, dass die Zieltabellen leer sind, wenn die Migration beginnt. AWS DMS erstellt nur die Objekte, die für eine effiziente Migration der Daten erforderlich sind: Tabellen, Primärschlüssel und in einigen Fällen eindeutige Indizes. AWS DMS erstellt keine Sekundärindizes, Einschränkungen für Nichtprimärschlüssel oder Standardwerte für Spaltendaten. Wenn Sie eine Full-Load-Aufgabe mit CDC oder eine reine CDC-Aufgabe ausführen, empfehlen wir, dass</p>

Option	Vorgehensweise
	<p>Sie die Migration an dieser Stelle pausieren. Erstellen Sie dann sekundäre Indizes, die das Filtern nach Update- und Löschanweisungen unterstützen.</p> <p>Möglicherweise müssen Sie einige Konfigurationseinstellungen in der Zieldatenbank vornehmen, wenn Sie den Modus Drop tables on target (Tabellen im Ziel löschen) verwenden. Beispielsweise AWS DMS kann für ein Oracle-Ziel aus Sicherheitsgründen kein Schema (Datenbankbenutzer) erstellt werden. In diesem Fall erstellen Sie den Schema-Benutzer vorab, damit er die Tabellen erstellen AWS DMS kann, wenn die Migration beginnt. Bei den meisten anderen Zieltypen werden das Schema und alle zugehörigen Tabellen mit den richtigen Konfigurationsparametern AWS DMS erstellt.</p> <p>Kürzen — Im Modus Kürzen werden alle AWS DMS Zieltabellen gekürzt, bevor die Migration beginnt. Wenn die Zieltabelle nicht vorhanden ist, wird sie von DMS erstellt. Ihre Tabellenstruktur bleibt unverändert, doch die Tabellen werden im Ziel gekürzt. Der Modus Truncate (Verkürzen) eignet sich für vollständige Ladevorgänge oder für den vollständigen Ladevorgang plus CDC-Migrationen, in denen das Zielschema erstellt worden ist, bevor die Migration gestartet wird. Mit AWS SCT können Sie Tabellen vorab erstellen . Weitere Informationen finden Sie unter Installation. AWS SCT</p> <div data-bbox="732 1591 1507 1866"><p> Note</p><p>Wenn Ihr Ziel MongoDB ist, werden im Modus Kürzen Tabellen im Ziel nicht gekürzt. Vielmehr wird die Sammlung gelöscht und alle Indizes gehen verloren. Vermeiden Sie</p></div>

Option	Vorgehensweise
	<p>die Verwendung des Modus Kürzen, wenn MongoDB Ihr Ziel ist.</p>
<p>Stop task after full load completes (Aufgabe nach Abschluss des vollständigen Ladens anhalten)</p>	<p>Diese Einstellung wird nur angezeigt, wenn Sie im vorherigen Abschnitt Migrieren von bestehenden Daten und Replizieren fortlaufender Änderungen für Migrationstyp ausgewählt haben.</p> <p>Nicht stoppen – Die Aufgabe nicht stoppen, zwischeng gespeicherte Änderungen jedoch sofort anwenden und Vorgang fortsetzen.</p> <p>Stoppen, bevor die zwischengespeicherten Änderungen angewendet werden – Die Aufgabe vor Anwendung der zwischengespeicherten Änderungen stoppen. Mit diesem Ansatz können Sie sekundäre Indizes hinzufügen, die die Übernahme von Änderungen beschleunigen können.</p> <p>Stoppen, nachdem die zwischengespeicherten Änderungen angewendet wurden – Die Aufgabe nach Anwendung der zwischengespeicherten Änderungen stoppen. Bei diesem Vorgehen können Sie Fremdschlüssel hinzufügen, wenn Sie die transaktionale Anwendung verwenden.</p>

Option	Vorgehensweise
Include LOB columns in replication (LOB-Spalten in Replikation einschließen)	<p>Keine LOB-Spalten berücksichtigen – LOB-Spalten werden von der Migration ausgeschlossen.</p> <p>Vollständiger LOB-Modus — Migrieren Sie komplette LOBs unabhängig von ihrer Größe. AWS DMS migriert LOBs stückweise in Chunks, die durch den Parameter LOB-Chunk-Größe gesteuert werden. Dieser Modus ist langsamer als der Modus "Limited LOB".</p> <p>Limitierter LOB-Modus – LOBs werden auf den Wert des Parameters Maximale LOB-Größe gekürzt. Dieser Modus ist schneller als der Modus "Full LOB".</p>
Maximale LOB-Größe (KB)	Im Limited LOB Mode (Eingeschränkter LOB-Modus) werden LOB-Spalten, die den Wert Max LOB Size (Maximale LOB-Größe) überschreiten, auf die angegebene Max LOB Size (Maximale LOB-Größe) gekürzt.
Enable validation (Validierung aktivieren)	Aktiviert die Datenvalidierung, um sicherzustellen, dass die Daten korrekt von der Quelle ans Ziel migriert werden. Weitere Informationen finden Sie unter Datenvalidierung mit AWS DMS .
CloudWatch Aktiviert Protokolle	Aktiviert die Protokollierung durch Amazon CloudWatch.

- Wählen Sie im Abschnitt Vormigrationsbewertung aus, ob eine Vormigrationsbewertung ausgeführt werden soll. Eine Vormigrationsbewertung warnt Sie vor potenziellen Migrationsproblemen, bevor Sie mit der Datenbankmigration beginnen. Weitere Informationen finden Sie unter [Aktivieren und Verwenden von Vormigrationsbewertungen](#).
- Geben Sie im Abschnitt Startup-Konfiguration der Migrationsaufgabe an, ob die Aufgabe automatisch nach der Erstellung gestartet werden soll.
- Geben Sie im Abschnitt Tags alle Tags an, die Sie zur Organisation Ihrer Aufgabe benötigen. Sie können Tags verwenden, um Ihre IAM-Rollen und -Richtlinien zu verwalten und Ihre DMS-Kosten nachzuverfolgen. Weitere Informationen finden Sie unter [Markieren von Ressourcen](#).

8. Wenn Sie mit den Aufgabeneinstellungen fertig sind, wählen Sie **Create task** (Aufgabe erstellen) aus.

Angeben von Aufgabeneinstellungen für Aufgaben des AWS Database Migration Service

Jede Aufgabe hat Einstellungen, die entsprechend den Anforderungen Ihrer Datenbankmigration konfiguriert werden können. Sie erstellen diese Einstellungen in einer JSON-Datei, oder bei einigen Einstellungen können Sie die Einstellungen mithilfe der AWS DMS Konsole angeben. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Es gibt mehrere Hauptarten von Aufgabeneinstellungen, wie nachfolgend aufgelistet:

Themen

- [Beispiel für Aufgabeneinstellungen](#)
- [Ziel-Metadaten-Aufgabeneinstellungen](#)
- [Aufgabeneinstellungen für vollständiges Laden](#)
- [Time-Travel-Aufgabeneinstellungen](#)
- [Einstellungen für das Protokollieren von Aufgaben](#)
- [Kontrolltabellen-Aufgabeneinstellungen](#)
- [Stream-Puffer-Aufgabeneinstellungen](#)
- [Einstellungen für die Optimierung der Verarbeitung von Änderungen](#)
- [Aufgabeneinstellungen zur Datenvalidierung](#)
- [Aufgabeneinstellungen für den Umgang mit der DDL-Änderungsverarbeitung](#)
- [Einstellungen der Zeichenersetzungsaufgabe](#)
- [Vorher-Abbild-Aufgabeneinstellungen](#)
- [Aufgabeneinstellungen zur Fehlerbehandlung](#)
- [Speichern der Aufgabeneinstellungen](#)

Task settings (Aufgabeneinstellungen)	Relevante Dokumentation
<p>Erstellen eines Aufgabenbewertungsberichts</p> <p>Sie können einen Aufgabenbewertungsbericht erstellen, der alle nicht unterstützten Datentypen aufzeigt, die während der Migration zu Problemen führen könnten. Sie können diesen Bericht vor der Ausführung Ihrer Aufgabe ausführen, um potenzielle Probleme aufzuzeigen.</p>	<p>Aktivieren und Verwenden von Vormigrationsbewertungen für eine Aufgabe</p>
<p>Erstellen einer Aufgabe</p> <p>Wenn Sie eine Aufgabe erstellen, geben Sie die Quelle, das Ziel und die Replikations-Instance zusammen mit den Migrationseinstellungen an.</p>	<p>Erstellen einer Aufgabe</p>
<p>Erstellen einer fortlaufenden Replikationsaufgabe</p> <p>Sie können eine Aufgabe einrichten, um die fortlaufende Replikation zwischen Quelle und Ziel bereitzustellen.</p>	<p>Erstellen von Aufgaben für die laufende Replikation mit AWS DMS</p>
<p>Anwenden von Aufgabeneinstellungen</p> <p>Jede Aufgabe hat Einstellungen, die entsprechend den Anforderungen Ihrer Datenbankmigration konfiguriert werden können. Sie erstellen</p>	<p>Angaben von Aufgabeneinstellungen für Aufgaben des AWS Database Migration Service</p>

Task settings (Aufgabeneinstellungen)	Relevante Dokumentation
diese Einstellungen in einer JSON-Datei, oder bei einigen Einstellungen können Sie die Einstellungen mithilfe der AWS DMS Konsole angeben.	
Datenvalidierung Verwenden Sie die Datenvalidierung, um die Daten in Ihrem Zieldaten Speicher mit den Daten aus Ihrem Quelldatenspeicher zu AWS DMS vergleichen.	Datenvalidierung mit AWS DMS
Ändern einer Aufgabe Wenn eine Aufgabe angehalten wird, können Sie die Einstellungen für die Aufgabe ändern.	Ändern einer Aufgabe
Erneutes Laden von Tabellen während einer Aufgabe Sie können eine Tabelle während einer Aufgabe neu laden, wenn während einer Aufgabe ein Fehler auftritt.	Erneutes Laden von Tabellen während einer Aufgabe
Verwenden der Tabellenzuweisung Die Tabellenzuweisung verwendet mehrere Arten von Aufgabeneinstellungen für die Angabe der Datenquelle, des Quellschemas, der Daten und aller Transformationen, die während der Aufgabe auftreten sollten.	Auswahlregeln Auswahlregeln und Aktionen Transformationsregeln Transformationsregeln und Aktionen

Task settings (Aufgabeneinstellungen)	Relevante Dokumentation
<p>Anwenden von Filtern</p> <p>Sie können Quellfilter verwenden, um Anzahl und Art der Datensätze zu beschränken, die von Ihrer Quelle auf Ihr Ziel übertragen werden. So können Sie beispielsweise festlegen, dass nur die am Hauptsitz ansässigen Angestellten in die Zieldatenbank übertragen werden. Sie wenden Filter auf eine Datenspalte an.</p>	<p>Verwenden von Quellfiltern</p>
<p>Überwachen einer Aufgabe</p> <p>Es gibt mehrere Möglichkeiten, um Informationen über die Durchführung einer Aufgabe und die von der Aufgabe verwendeten Tabellen zu erhalten.</p>	<p>Überwachen von AWS-DMS-Aufgaben</p>
<p>Verwalten von Aufgabenprotokollen</p> <p>Sie können Aufgabenprotokolle mithilfe der AWS DMS API oder anzeigen und löschen AWS CLI.</p>	<p>Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen</p>

Beispiel für Aufgabeneinstellungen

Sie können entweder den AWS Management Console oder verwenden AWS CLI , um eine Replikationsaufgabe zu erstellen. Wenn Sie die verwenden AWS CLI, legen Sie die Aufgabeneinstellungen fest, indem Sie eine JSON-Datei erstellen und dann den file://URI der JSON-Datei als [ReplicationTaskSettings](#)Parameter des [CreateReplicationTask](#)Vorgangs angeben.

Das folgende Beispiel zeigt, wie der AWS CLI zum Aufrufen der CreateReplicationTask Operation verwendet wird:

```
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
```

Im vorherigen Beispiel wurde eine Tabellenzuweisungsdatei namens `tablemappings.json` verwendet. Beispiele für Tabellenzuweisungen finden Sie unter [Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen](#).

Eine JSON-Datei für Aufgabeneinstellungen kann folgendermaßen aussehen.

```
{  
  "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
  }
}

```

Ziel-Metadaten-Aufgabeneinstellungen

Zu den Ziel-Metadateneinstellungen gehören folgende Einstellungen: Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

- **TargetSchema** – Der Name des Zieltabellenschemas. Wenn diese Metadatenoption leer ist, wird das Schema der Quelltable verwendet. AWS DMS fügt allen Tabellen automatisch das Eigentümerpräfix für die Zieldatenbank hinzu, wenn kein Quellschema definiert ist. Diese Option sollte für Ziel-Endpunkte vom Typ MySQL leer gelassen werden. Das Umbenennen eines Schemas in der Datenzuordnung hat Vorrang vor dieser Einstellung.
- **LOB-Einstellungen** – Einstellungen, die bestimmen, wie große Objekte (LOBs) verwaltet werden. Wenn Sie `SupportLobs=true` festlegen, müssen Sie eines der Folgenden auf `true` setzen:

- `FullLobMode` – Wenn Sie für diese Option `true` festlegen, müssen Sie einen Wert für die Option `LobChunkSize` eingeben. Geben Sie die zu verwendende Größe der LOB-Blöcke (in Kilobyte) ein, wenn die Daten in der Zieldatenbank repliziert werden. Die Option `FullLobMode` funktioniert am besten bei sehr großen LOB-Größen, verlangsamt jedoch den Ladevorgang. Der empfohlene Wert für `LobChunkSize` ist 64 Kilobyte. Wenn für `LobChunkSize` ein höherer Wert als 64 Kilobyte verwendet wird, kann dies zu Aufgabenfehlern führen.
- `InlineLobMaxSize`— Dieser Wert bestimmt, welche LOBs bei Volllast inline AWS DMS übertragen werden. Es ist effizienter, kleine LOBs zu übertragen als sie in einer Quelltable nachzuschlagen. AWS DMS überprüft bei einem vollen Ladevorgang alle LOBs und führt eine Inline-Übertragung für die LOBs durch, die kleiner als `InlineLobMaxSize` sind. AWS DMS überträgt alle LOBs, die größer als `InlineLobMaxSize` sind. Der Standardwert für `InlineLobMaxSize` ist 0 und der Bereich liegt zwischen 1 und 102 400 Kilobyte (100 MB). Legen Sie einen Wert für `InlineLobMaxSize` nur fest, wenn Sie wissen, dass die meisten LOBs kleiner sind als der unter `InlineLobMaxSize` festgelegte Wert.
- `LimitedSizeLobMode` – Wenn Sie für diese Option `true` festlegen, müssen Sie einen Wert für die Option `LobMaxSize` eingeben. Geben Sie die maximale Größe in Kilobyte für ein einzelnes LOB ein. Der empfohlene Höchstwert für `LobMaxSize` liegt bei 102 400 Kilobyte (100 MB).

Weitere Informationen zu den Kriterien für die Verwendung dieser Aufgaben-LOB-Einstellungen finden Sie unter [Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS](#). Sie können auch die Verwaltung von LOBs für einzelne Tabellen steuern. Weitere Informationen finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

- `LoadMaxFileSize` – Eine Option für CSV-basierte Zielendpunkte wie MySQL, PostgreSQL und Amazon Redshift, die die Verwendung von Dateien mit kommagetrennten Werten (CSV) zum Laden von Daten unterstützen. `LoadMaxFileSize` legt die maximale Größe von gespeicherten, entladenen Daten, wie z. B. CSV-Dateien, auf der Festplatte fest. Diese Option setzt das Zielendpunkt-Verbindungsattribut, `maxFileSize`, außer Kraft. Sie können Werte von 0 bis 100.000 KB angeben, wobei "0" anzeigt, dass diese Option das Verbindungsattribut nicht außer Kraft setzt.
- `BatchApplyEnabled` – Legt fest, ob jede Transaktion einzeln angewendet wird oder ob Änderungen in Stapeln übernommen werden. Der Standardwert ist `false`.

Wenn `BatchApplyEnabled` auf `true` gesetzt ist, erfordert DMS einen Primärschlüssel (PK) oder einen eindeutigen Schlüssel (Unique Key – UK) in der/den Quelltable(n). Ohne PK oder

UK in den Quelltabellen werden nur Stapeleinfügungen, jedoch keine Stapelaktualisierungen und Stapellöschungen angewendet.

Wenn `BatchApplyEnabled` auf `true` gesetzt ist, generiert AWS DMS eine Fehlermeldung, wenn eine Ziel-Tabelle über eine Eindeutigkeitseinschränkung und einen Primärschlüssel verfügt. Zieltabellen mit Eindeutigkeitseinschränkung und Primärschlüssel werden nicht unterstützt, wenn `BatchApplyEnabled` auf `true` gesetzt ist.

Wenn der Wert auf `true` gesetzt `BatchApplyEnabled` ist AWS DMS und ein Datenfehler in einer Tabelle mit der Standardrichtlinie zur Fehlerbehandlung auftritt, wechselt die AWS DMS Aufgabe vom Batchmodus in den one-by-one Modus für die übrigen Tabellen. Um dieses Verhalten zu ändern, können Sie die Aktion "SUSPEND_TABLE" für die folgenden Richtlinien in der Gruppeneigenschaft "ErrorBehavior" der JSON-Datei mit den Aufgabeneinstellungen festlegen:

- `DataErrorPolicy`
- `ApplyErrorDeletePolicy`
- `ApplyErrorInsertPolicy`
- `ApplyErrorUpdatePolicy`

Weitere Informationen zu dieser Gruppeneigenschaft "ErrorBehavior" finden Sie in der JSON-Beispieldatei mit Aufgabeneinstellungen in [Angeben von Aufgabeneinstellungen für Aufgaben des AWS Database Migration Service](#). Nachdem diese Richtlinien auf gesetzt wurden "SUSPEND_TABLE", unterbricht die AWS DMS Aufgabe Datenfehler in allen Tabellen, die sie auslösen, und setzt den Vorgang für alle Tabellen im Batchmodus fort.

Sie können den Parameter `BatchApplyEnabled` mit dem Parameter `BatchApplyPreserveTransaction` verwenden. Wenn `BatchApplyEnabled` auf `true` festgelegt ist, bestimmt der Parameter `BatchApplyPreserveTransaction` die Integrität der Transaktionen.

Wenn `BatchApplyPreserveTransaction` auf `true` festgelegt ist, wird die Transaktionsintegrität bewahrt und ein Stapel enthält garantiert alle Änderungen innerhalb einer Transaktion von der Quelle.

Wenn `BatchApplyPreserveTransaction` auf `false` festgelegt ist, kann es zu temporären Ausfällen bei der Transaktionsintegrität kommen, um die Leistung zu verbessern.

Der Parameter `BatchApplyPreserveTransaction` gilt ausschließlich für Oracle-Zielendpunkte und ist nur relevant, wenn der Parameter `BatchApplyEnabled` auf `true` festgelegt ist.

Wenn die Replikation LOB-Spalten enthält, können Sie `BatchApplyEnabled` nur im Limited LOB-Modus verwenden.

Weitere Informationen zur Verwendung dieser Einstellungen zum CDC-Laden (CDC = Change Data Capture) finden Sie unter [Einstellungen für die Optimierung der Verarbeitung von Änderungen](#).

- `MaxFullLoadSubTasks` – Gibt die maximale Anzahl von Tabellen an, die parallel geladen werden sollen. Der Standardwert beträgt 8; der Maximalwert beträgt 49.
- `ParallelLoadThreads`— Gibt die Anzahl der Threads an, die AWS DMS verwendet werden, um jede Tabelle in die Zieldatenbank zu laden. Für diesen Parameter gelten Höchstwerte für Nicht-RDBMS-Ziele. Der Höchstwert für ein DynamoDB-Ziel beträgt 200. Der Höchstwert für ein Amazon Kinesis Data Streams-, Apache Kafka- oder Amazon OpenSearch Service-Ziel ist 32. Sie können eine Erhöhung dieses Höchstwerts beantragen. `ParallelLoadThreads` gilt für Aufgaben mit vollständigem Ladevorgang. Weitere Informationen zu den Einstellungen, mit denen das parallele Laden einzelner Tabellen aktiviert wird, finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Diese Einstellung gilt für die folgenden Endpunkt-Engine-Typen:

- DynamoDB
- Amazon-Kinesis-Data-Streams
- Amazon MSK
- OpenSearch Amazon-Dienst
- Amazon-Redshift

AWS DMS unterstützt `ParallelLoadThreads` MySQL als zusätzliches Verbindungsattribut. `ParallelLoadThreadsgilt` nicht für MySQL als Aufgabeneinstellung.

- `ParallelLoadBufferSize` gibt die maximale Anzahl der Datensätze an, die in dem Puffer gespeichert werden können, den die parallelen Lade-Threads zum Laden von Daten in das Ziel verwenden. Der Standardwert lautet 50. Die maximale Wert ist 1.000. Diese Einstellung ist derzeit nur gültig, wenn DynamoDB, Kinesis, Apache Kafka oder OpenSearch das Ziel ist. Verwenden Sie diesen Parameters mit `ParallelLoadThreads`. `ParallelLoadBufferSize` ist nur gültig, wenn mehr als ein Thread vorhanden ist. Weitere Informationen zu den Einstellungen, mit denen

das parallele Laden einzelner Tabellen aktiviert wird, finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

- `ParallelLoadQueuesPerThread` – Gibt die Anzahl der Warteschlangen an, auf die jeder gleichzeitige Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und einen Stapelladevorgang für das Ziel zu generieren. Der Standardwert ist 1. Diese Einstellung ist derzeit nur gültig, wenn Kinesis oder Apache Kafka das Ziel ist.
- `ParallelApplyThreads`— Gibt die Anzahl der gleichzeitigen Threads an, die während eines CDC-Ladens AWS DMS verwendet werden, um Datensätze an einen Amazon DocumentDB-, Kinesis-, Amazon MSK- oder Amazon Redshift Redshift-Zielendpunkt OpenSearch zu übertragen. Der Standardwert ist Null (0).

Diese Einstellung gilt nur für reines CDC. Diese Einstellung gilt nicht für vollständiges Laden.


Diese Einstellung gilt für die folgenden Endpunkt-Engine-Typen:

- Amazon DocumentDB (mit MongoDB-Kompatibilität)
- Amazon-Kinesis-Data-Streams
- Amazon Managed Streaming für Apache Kafka
- OpenSearch Amazon-Dienst
- Amazon-Redshift
- `ParallelApplyBufferSize`— Gibt die maximale Anzahl von Datensätzen an, die in jeder Pufferwarteschlange gespeichert werden sollen, damit gleichzeitige Threads während eines CDC-Ladevorgangs an einen Amazon DocumentDB-, Kinesis-, Amazon MSK- oder Amazon Redshift-Zielendpunkt weitergeleitet werden. OpenSearch Der Standardwert lautet 100. Der Höchstwert ist 1000. Verwenden Sie diese Option, wenn `ParallelApplyThreads` mehrere Threads angibt.
- `ParallelApplyQueuesPerThread`— Gibt die Anzahl der Warteschlangen an, auf die jeder Thread zugreift, um Datensätze aus Warteschlangen zu entfernen und während der CDC einen Batchload für eine Amazon DocumentDB, Kinesis, Amazon MSK oder einen Endpunkt zu generieren. OpenSearch Der Standardwert lautet 1.

Aufgabeneinstellungen für vollständiges Laden

Zu den Einstellungen für vollständiges Laden gehören die folgenden Einstellungen. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

- Um anzuzeigen, wie das Laden der Zieldatenbank beim Start von Full Load zu behandeln ist, bestimmen Sie einen der folgenden Werte für die Option `TargetTablePrepMode`:
 - `DO_NOTHING` – Daten und Metadaten der vorhandenen Zieltabelle sind nicht betroffen.
 - `DROP_AND_CREATE` – Die vorhandene Tabelle wird gelöscht und stattdessen wird eine neue Tabelle erstellt.
 - `TRUNCATE_BEFORE_LOAD` – Daten werden gekürzt, ohne dass sich dies auf die Metadaten der Tabelle auswirkt.
- Um das Erstellen des Primärschlüssels oder des eindeutigen Index bis nach Abschluss des vollständigen Ladens zu verzögern, setzen Sie die Option `CreatePkAfterFullLoad` auf `true`.
- Für Aufgaben, bei denen Full-Load und CDC aktiviert sind, können Sie die folgenden Optionen für `Stop task after full load completes` festlegen:
 - `StopTaskCachedChangesApplied` – Legen Sie für diese Option `true` fest, um eine Aufgabe zu stoppen, nachdem das vollständige Laden abgeschlossen ist und zwischengespeicherte Änderungen angewendet wurden.
 - `StopTaskCachedChangesNotApplied` – Legen Sie für diese Option `true` fest, um eine Aufgabe zu stoppen, bevor zwischengespeicherte Änderungen angewendet werden.
- Um die maximale Anzahl von Tabellen anzugeben, die parallel geladen werden, legen Sie die Option `MaxFullLoadSubTasks` fest. Der Standardwert beträgt 8; der Maximalwert beträgt 49.
- Legen Sie die Option `ParallelLoadThreads` fest, um anzugeben, wie viele gleichzeitige Threads DMS während eines vollständigen Ladevorgangs verwenden soll, um Datensätze an einen Zielendpunkt zu übertragen. Der Standardwert ist Null (0).

 **Important**

`MaxFullLoadSubTasks` steuert die Anzahl der Tabellen oder Tabellensegmente, die parallel geladen werden sollen. `ParallelLoadThreads` steuert die Anzahl der Threads, die von einer Migrationsaufgabe verwendet werden, um die Ladevorgänge parallel auszuführen. Diese Einstellungen sind multiplikativ. Daher ist die Gesamtzahl der Threads, die während einer Vollast-Aufgabe verwendet werden, ungefähr das Ergebnis des Werts `ParallelLoadThreads` multipliziert mit dem Wert von `MaxFullLoadSubTasks` (`ParallelLoadThreads * MaxFullLoadSubtasks`).

Wenn Sie Aufgaben mit einer hohen Anzahl von Unteraufgaben für vollständiges Laden und einer hohen Anzahl von parallelen Lade-Threads erstellen, kann Ihre Aufgabe zu viel Speicher verbrauchen und fehlschlagen.

- Sie können festlegen, wie viele Sekunden auf das Schließen von Transaktionen AWS DMS gewartet werden soll, bevor ein Vollladevorgang gestartet wird. Wenn Transaktionen beim Start der Aufgabe geöffnet sind, legen Sie dazu die Option `TransactionConsistencyTimeout` fest. Der Standardwert ist 600 (10 Minuten). AWS DMS startet den vollen Ladevorgang, nachdem der Timeout-Wert erreicht ist, auch wenn offene Transaktionen vorhanden sind. Eine full-load-only Aufgabe wartet nicht 10 Minuten, sondern wird sofort gestartet.
- Um die maximale Anzahl von Datensätzen anzugeben, die zusammen übertragen werden können, legen Sie die Option `CommitRate` fest. Der Standardwert ist 10 000 und der maximale Wert 50 000.

Time-Travel-Aufgabeneinstellungen

Um Replikationsaufgaben zu protokollieren und zu debuggen, können Sie AWS DMS Time Travel verwenden. Bei diesem Vorgehen verwenden Sie Amazon S3, um Protokolle zu speichern und mit Ihren Verschlüsselungsschlüsseln zu verschlüsseln. Nur wenn Sie Zugriff auf Ihren Time-Travel-S3-Bucket haben, können Sie Ihre S3-Protokolle mithilfe von Datums- und Uhrzeitfiltern abrufen und die Protokolle dann nach Bedarf anzeigen, herunterladen und verschleiern. Auf diese Weise können Sie sicher „in die Vergangenheit reisen“, um Datenbankaktivitäten zu untersuchen. Time Travel funktioniert unabhängig von der CloudWatch Protokollierung. Weitere Informationen zur CloudWatch Protokollierung finden Sie unter [Einstellungen für das Protokollieren von Aufgaben](#).

Sie können Time Travel in allen AWS Regionen mit AWS DMS unterstützten Oracle-, Microsoft SQL Server- und PostgreSQL-Quellendpunkten und AWS DMS-unterstützten PostgreSQL- und MySQL-Zielendpunkten verwenden. Sie können Time Travel nur für Aufgaben für vollständiges Laden und CDC (Change Data Capture) und für reine CDC-Aufgaben aktivieren. Um Time Travel zu aktivieren oder vorhandene Time-Travel-Einstellungen zu ändern, stellen Sie sicher, dass Ihre Replikationsaufgabe gestoppt wurde.

Die Time-Travel-Einstellungen umfassen die folgenden `TTSettings`-Eigenschaften:

- `EnableTT` – Wenn diese Option auf `true` gesetzt ist, ist die Time-Travel-Protokollierung für die Aufgabe aktiviert. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

- **EncryptionMode** – Die Art der serverseitigen Verschlüsselung, die in Ihrem S3-Bucket zum Speichern Ihrer Daten und Protokolle verwendet wird. Sie können entweder "SSE_S3" (Standardwert) oder "SSE_KMS" angeben.

Sie können **EncryptionMode** von "SSE_KMS" auf "SSE_S3" umstellen, aber nicht umgekehrt.

Typ: Zeichenfolge

Erforderlich: Nein

- **ServerSideEncryptionKmsKeyId**— Wenn Sie "SSE_KMS" für angeben, geben Sie die ID für Ihren benutzerdefinierten **EncryptionMode** verwalteten Schlüssel an. AWS KMS Stellen Sie sicher, dass dem Schlüssel, den Sie verwenden, eine Richtlinie angehängt ist, die AWS Identity and Access Management (IAM-) Benutzerberechtigungen aktiviert und die Verwendung des Schlüssels ermöglicht.

Mit der Option "SSE_KMS" wird nur Ihr eigener benutzerdefiniert-verwalteter symmetrischer KMS-Schlüssel unterstützt.

Typ: Zeichenfolge

Erforderlich: nur wenn **EncryptionMode** auf "SSE_KMS" gesetzt wird

- **ServiceAccessRoleArn** – Der Amazon-Ressourcenname (ARN), den der Service zum Zugriff auf die IAM-Rolle verwendet. Legen Sie für den Rollennamen `dms-tt-s3-access-role` fest. Dies ist eine erforderliche Einstellung, die das Schreiben und Lesen von Objekten aus einem S3-Bucket ermöglicht AWS DMS .

Typ: Zeichenfolge

Erforderlich: wenn Time Travel aktiviert ist

Im Folgenden sehen Sie eine Beispielrichtlinie für diese Rolle.

```
{
  "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"
    ]
}
]
}

```

Im Folgenden sehen Sie eine Beispielvertrauensrichtlinie für diese Rolle.

```

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

```

- **BucketName** – Der Name des S3-Buckets, in dem Time-Travel-Protokolle gespeichert werden sollen. Stellen Sie sicher, dass Sie diesen S3-Bucket erstellen, bevor Sie Time-Travel-Protokolle aktivieren.

Typ: Zeichenfolge

Erforderlich: wenn Time Travel aktiviert ist

- **BucketFolder** – Ein optionaler Parameter zum Festlegen eines Ordernamens im S3-Bucket. Wenn Sie diesen Parameter angeben, erstellt DMS die Time-Travel-Protokolle im Pfad `"/BucketName/BucketFolder/taskARN/YYYY/MM/DD/hh"`. Wenn Sie diesen Parameter

nicht angeben, AWS DMS wird der Standardpfad als erstellt `"/BucketName/dms-time-travel-logs/taskARN/YYYY/MM/DD/hh`.

Typ: Zeichenfolge

Erforderlich: Nein

- `EnableDeletingFromS3OnTaskDelete`— Wenn diese Option auf `true` gesetzt ist, werden die Time Travel-Protokolle aus S3 AWS DMS gelöscht, wenn die Aufgabe gelöscht wird. Der Standardwert ist `false`.

Typ: Zeichenfolge

Erforderlich: Nein

- `EnableRawData` – Wenn diese Option auf `true` gesetzt ist, werden die DML-Rohdaten (DML = Data Manipulation Language) für Time-Travel-Protokolle in der Spalte `raw_data` der Time-Travel-Protokolle angezeigt. Einzelheiten finden Sie unter [Verwenden der Time-Travel-Protokolle](#). Der Standardwert ist `false`. Wenn diese Option auf `false` gesetzt ist, wird nur der DML-Typ erfasst.

Typ: Zeichenfolge

Erforderlich: Nein

- `RawDataFormat`— In den AWS DMS Versionen 3.5.0 und höher `EnableRawData` ist wann auf eingestellt. `true` Diese Eigenschaft gibt ein Format für die Rohdaten der DML in einem Time-Travel-Protokoll an und kann wie folgt angegeben werden:
 - `"TEXT"` – Analysierte, lesbare Spaltennamen und Werte für DML-Ereignisse, die während CDC als Raw-Felder erfasst werden.
 - `"HEX"` – Der ursprüngliche Hexadezimalwert für Spaltennamen und Werte, die für DML-Ereignisse während CDC erfasst werden.

Diese Eigenschaft gilt für Oracle- und Microsoft-SQL-Server-Datenbankquellen.

Typ: Zeichenfolge

Erforderlich: Nein

- `OperationsToLog` – Gibt den Typ der DML-Operationen an, die in Time-Travel-Protokollen protokolliert werden sollen. Sie können einen der folgenden Werte angeben:
 - `"INSERT"`
 - `"UPDATE"`

- "DELETE"
- "COMMIT"
- "ROLLBACK"
- "ALL"

Der Standardwert ist "ALL".

Typ: Zeichenfolge

Erforderlich: Nein

- `MaxRecordSize` – Gibt die maximale Größe der Time-Travel-Protokolleinträge an, die für jede Zeile protokolliert werden. Verwenden Sie diese Eigenschaft, um das Wachstum der Time-Travel-Protokolle für besonders stark ausgelastete Tabellen zu steuern. Der Standardwert ist 64 KB.

Typ: Ganzzahl

Erforderlich: Nein

Weitere Informationen zum Aktivieren und Verwenden von Time-Travel-Protokollen finden Sie in den folgenden Themen.

Themen

- [Aktivieren von Time-Travel-Protokollen für eine Aufgabe](#)
- [Verwenden der Time-Travel-Protokolle](#)
- [Wie oft werden Time AWS DMS Travel-Protokolle auf S3 hochgeladen](#)

Aktivieren von Time-Travel-Protokollen für eine Aufgabe

Sie können Time Travel für eine AWS DMS Aufgabe mithilfe der zuvor beschriebenen Aufgabeneinstellungen aktivieren. Stellen Sie sicher, dass Ihre Replikationsaufgabe gestoppt wurde, bevor Sie Time Travel aktivieren.

Um Time Travel zu aktivieren, verwenden Sie den AWS CLI

1. Erstellen Sie eine JSON-Datei für die DMS-Aufgabenkonfiguration und fügen Sie einen Abschnitt `TTSettings` hinzu, wie im Folgenden dargestellt. Informationen zur Verwendung einer

Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

```
.
.
.
  },
"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. Geben Sie in einer entsprechenden Aufgabenaktion diese JSON-Datei unter Verwendung der Option `--replication-task-settings` an. Im folgenden CLI-Code-Fragment ist diese Time-Travel-Einstellungsdatei beispielsweise als Teil von `create-replication-task` angegeben.

```
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
      .
      .
      .
```

Hier lautet der Name dieser Time-Travel-Einstellungsdatei `task-settings-tt-enabled.json`.

Sie können diese Datei in ähnlicher Weise im Rahmen der Aktion `modify-replication-task` angeben.

Beachten Sie die besondere Behandlung von Time-Travel-Protokollen für die folgenden Aufgabenaktionen:

- `start-replication-task` – Wenn Sie eine Replikationsaufgabe ausführen und auf einen für Time Travel verwendeten S3-Bucket nicht zugegriffen werden kann, wird die Aufgabe als `FAILED` markiert.
- `stop-replication-task`— Wenn die Aufgabe beendet wird, werden AWS DMS sofort alle Time Travel-Protokolle, die derzeit für die Replikationsinstanz verfügbar sind, in den für Time Travel verwendeten S3-Bucket übertragen.

Während der Ausführung einer Replikationsaufgabe können Sie den Wert für `EncryptionMode` von `"SSE_KMS"` in `"SSE_S3"` ändern, aber nicht umgekehrt.

Wenn die Größe der Time-Travel-Protokolle für eine laufende Aufgabe 1 GB überschreitet, überträgt DMS die Protokolle innerhalb von fünf Minuten nach Erreichen dieser Größe an S3. Wenn nach der Ausführung einer Aufgabe kein Zugriff auf den S3-Bucket oder den KMS-Schlüssel mehr möglich ist, stoppt DMS die Übertragung von Protokollen in diesen Bucket. Wenn Sie feststellen, dass Ihre Protokolle nicht in Ihren S3-Bucket übertragen werden, überprüfen Sie Ihren S3 und Ihre AWS KMS Berechtigungen. Weitere Informationen darüber, wie oft DMS diese Protokolle an S3 überträgt, finden Sie unter [Wie oft werden Time AWS DMS Travel-Protokolle auf S3 hochgeladen](#).

Um Time Travel für eine bestehende Aufgabe von der Konsole aus zu aktivieren, verwenden Sie die JSON-Editor-Option unter Aufgabeneinstellungen, um einen Abschnitt `TTS settings` hinzuzufügen.

Verwenden der Time-Travel-Protokolle

Time-Travel-Protokolldateien sind CSV-Dateien mit folgenden Feldern.

```
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
```

Sobald Ihre Time-Travel-Protokolle in S3 verfügbar sind, können Sie mit Tools wie Amazon Athena direkt darauf zugreifen und sie abfragen. Sie können die Protokolle auch wie jede andere Datei von S3 herunterladen.

Das folgende Beispiel zeigt ein Time-Travel-Protokoll, in dem Transaktionen für eine Tabelle namens `mytable` protokolliert sind. Die Zeilenenden für das folgende Protokoll wurden aus Gründen der besseren Lesbarkeit hinzugefügt.

```
"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'"
```


Wie oft werden Time AWS DMS Travel-Protokolle auf S3 hochgeladen

Um die Speichernutzung Ihrer Replikationsinstanz zu minimieren, laden Sie in AWS DMS regelmäßigen Abständen Time Travel-Protokolle von ihr ab.

Die Time-Travel-Protokolle werden in folgenden Fällen in Ihren Amazon-S3-Bucket übertragen:

- Wenn die aktuelle Größe der Protokolle 1 GB überschreitet, werden die Protokolle innerhalb von fünf Minuten auf S3 AWS DMS hochgeladen. Somit AWS DMS können bis zu 12 Anrufe pro Stunde an S3 und AWS KMS für jede laufende Aufgabe getätigt werden.
- AWS DMS lädt die Protokolle jede Stunde auf S3 hoch, unabhängig von der Größe der Protokolle.
- Wenn eine Aufgabe gestoppt wird, werden die Zeitreiseprotokolle AWS DMS sofort nach S3 hochgeladen.

Einstellungen für das Protokollieren von Aufgaben

Logging verwendet Amazon CloudWatch, um Informationen während des Migrationsprozesses zu protokollieren. Mit den Einstellungen der Protokollierungsaufgabe können Sie angeben, welche Komponentenaktivitäten protokolliert und welche Menge an Informationen in das Protokoll geschrieben wird. Einstellungen für die Protokollierung werden in eine JSON-Datei geschrieben. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Sie können die CloudWatch Protokollierung auf verschiedene Arten aktivieren. Sie können die `EnableLogging` Option AWS Management Console bei der Erstellung einer Migrationaufgabe auswählen. Sie können die `EnableLogging` Option auch auf `true` einstellen, wenn Sie eine Aufgabe mithilfe der AWS DMS API erstellen. Sie können auch `"EnableLogging": true` in JSON im Abschnitt Protokollierung der Aufgabeneinstellungen angeben.

Wenn Sie `EnableLogging` auf `true` festlegen, werden der CloudWatch Gruppenname und der Streamname wie folgt AWS DMS zugewiesen. Sie können diese Werte nicht direkt festlegen.

- `CloudWatchLogGroup`: `dms-tasks-<REPLICATION_INSTANCE_IDENTIFIER>`
- `CloudWatchLogStream`: `dms-task-<REPLICATION_TASK_EXTERNAL_RESOURCE_ID>`

`<REPLICATION_INSTANCE_IDENTIFIER>` ist die ID der Replikations-Instance.

`<REPLICATION_TASK_EXTERNAL_RESOURCE_ID>` ist der Wert des Abschnitts `<resourcename>`

des Aufgaben-ARN. Hinweise zur AWS DMS Generierung von Ressourcen-ARNs finden Sie unter [Erstellung eines Amazon-Ressourcennamens \(ARN\) für AWS DMS](#)

CloudWatch ist in AWS Identity and Access Management (IAM) integriert, und Sie können angeben, welche CloudWatch Aktionen ein Benutzer in Ihrem AWS Konto ausführen kann. Weitere Informationen zur Arbeit mit IAM in CloudWatch finden Sie unter [Identitäts- und Zugriffsmanagement für Amazon CloudWatch](#) und [Protokollierung von CloudWatch Amazon-API-Aufrufen](#) im CloudWatch Amazon-Benutzerhandbuch.

Zum Löschen der Aufgabenprotokolle können Sie `DeleteTaskLogs` im JSON-Code im Protokollierungsabschnitt der Aufgabeneinstellungen angeben.

Sie können eine Protokollierung für die folgenden Arten von Ereignissen festlegen:

- `FILE_FACTORY` – Die Datei-Factory verwaltet Dateien, die für die Stapelanwendung und das Stapelladen verwendet werden, und verwaltet Amazon-S3-Endpunkte.
- `METADATA_MANAGER` – Der Metadatenmanager verwaltet die Quell- und Zielmetadaten, die Partitionierung und den Tabellenstatus während der Replikation.
- `SORTER` – Der `SORTER` empfängt eingehende Ereignisse aus dem Prozess `SOURCE_CAPTURE`. Die Ereignisse werden in Transaktionen gebündelt und an die `TARGET_APPLY`-Servicekomponente übergeben. Wenn der Prozess `SOURCE_CAPTURE` Ereignisse schneller generiert, als die `TARGET_APPLY`-Komponente sie verarbeiten kann, speichert die `SORTER`-Komponente die zurückgestellten Ereignisse auf der Festplatte oder in einer Auslagerungsdatei zwischen. Zwischengespeicherte Ereignisse sind eine häufige Ursache für mangelnden Speicherplatz in Replikations-Instances.

Die `SORTER`-Servicekomponente verwaltet zwischengespeicherte Ereignisse, erfasst CDC-Statistiken und meldet die Latenz von Aufgaben.

- `SOURCE_CAPTURE` – Daten zur fortlaufenden Replikation (CDC) werden aus der Quelldatenbank oder dem Quellservice erfasst und an die `SORTER`-Servicekomponente weitergegeben.
- `SOURCE_UNLOAD` – Daten werden beim vollständigen Laden aus der Quelldatenbank oder dem Quellservice entladen.
- `TABLES_MANAGER` – Der Tabellenmanager verfolgt die erfassten Tabellen, verwaltet die Reihenfolge der Tabellenmigration und erfasst Tabellenstatistiken.
- `TARGET_APPLY` – Daten und Data Definition Language (DDL)-Anweisungen werden auf die Zieldatenbank angewendet.
- `TARGET_LOAD` – Daten werden in die Zieldatenbank geladen.

- **TASK_MANAGER** – Der Task Manager verwaltet laufende Aufgaben und unterteilt Aufgaben in Unteraufgaben für die parallele Datenverarbeitung.
- **TRANSFORMATION** – Transformationsereignisse im Zusammenhang mit Tabellenzuweisungen. Weitere Informationen finden Sie unter [Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen](#).
- **VALIDATOR/ VALIDATOR_EXT** – Die Servicekomponente VALIDATOR überprüft, ob die Daten korrekt von der Quelle zum Ziel migriert wurden. Weitere Informationen finden Sie unter [Datenvalidierung](#).

Die folgenden Protokollierungskomponenten generieren eine große Anzahl von Protokollen, wenn der Protokollschweregrad `LOGGER_SEVERITY_DETAILED_DEBUG` verwendet wird:

- `COMMON`
- `ADDONS`
- `DATA_STRUCTURE`
- `COMMUNICATION`
- `FILE_TRANSFER`
- `FILE_FACTORY`

Andere Protokollierungsstufen als `DEFAULT` werden für diese Komponenten bei der Problembehandlung selten benötigt. Es wird nicht empfohlen, die Protokollierungsebene `DEFAULT` für diese Komponenten zu ändern, sofern dies nicht ausdrücklich vom AWS Support gewünscht wird.

Nachdem Sie einen der oben genannten Werte angegeben haben, können Sie die Menge der protokollierten Informationen angeben, wie in der folgenden Liste dargestellt.

Die Schweregrade sind von der niedrigsten bis zur höchsten Informationsstufe angeordnet. Die höheren Stufen enthalten immer Informationen von den niedrigeren Stufen.

- `LOGGER_SEVERITY_ERROR` – Fehlermeldungen werden in das Protokoll geschrieben.
- `LOGGER_SEVERITY_WARNING` – Warnungen und Fehlermeldungen werden in das Protokoll geschrieben.
- `LOGGER_SEVERITY_INFO` – Informationsmeldungen, Warnungen und Fehlermeldungen werden in das Protokoll geschrieben.

- `LOGGER_SEVERITY_DEFAULT` – Informationsmeldungen, Warnungen und Fehlermeldungen werden in das Protokoll geschrieben.
- `LOGGER_SEVERITY_DEBUG` – Debug-Nachrichten, Informationsmeldungen, Warnungen und Fehlermeldungen werden in das Protokoll geschrieben.
- `LOGGER_SEVERITY_DETAILED_DEBUG` – Alle Informationen werden in das Protokoll geschrieben.

Das folgende JSON-Beispiel zeigt Aufgabeneinstellungen für die Protokollierung aller Aktionen und Schweregrade.

```
...
  "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
},
...
```

Kontrolltabellen-Aufgabeneinstellungen

Steuertabellen enthalten Informationen zu einer AWS DMS Aufgabe. Sie bieten auch nützliche Statistiken, mit denen Sie sowohl die aktuelle Migrationsaufgabe als auch zukünftige Aufgaben planen und verwalten können. Sie können diese Aufgabeneinstellungen in einer JSON-Datei anwenden oder indem Sie in der AWS DMS Konsole auf der Seite „Aufgabe erstellen“ die Option Erweiterte Einstellungen auswählen. Die Tabelle „Ausnahmen anwenden“ (`dmslogs.awsdms_apply_exceptions`) wird immer in Datenbankzielen erstellt. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

AWS DMS erstellt Steuertabellen nur bei Aufgaben mit Volllast und CDC oder nur CDC und nicht bei Aufgaben vom Typ „Nur Volllast“.

Für Aufgaben mit vollständigem Laden und CDC (Migration vorhandener Daten und Replikation fortlaufender Änderungen) und für reine CDC-Aufgaben (nur Replikation von Datenänderungen) können Sie auch zusätzliche Tabellen erstellen, beispielsweise folgende:

- Replikationsstatus (`dmslogs.awsdms_status`) – Diese Tabelle enthält Details zur aktuellen Aufgabe. Dazu gehören der Aufgabenstatus, die Größe des von der Aufgabe genutzten Speichers und die Anzahl an Änderungen, die noch nicht auf das Ziel angewendet wurden. Diese Tabelle gibt auch die Position in der Quelldatenbank an, an der gerade gelesen AWS DMS wird. Außerdem gibt sie an, ob sich die Aufgabe in der Phase des vollständigen Ladens oder der Change Data Capture (CDC) befindet.
- Ausgesetzte Tabellen (`dmslogs.awsdms_suspended_tables`) – Diese Tabelle enthält eine Liste der ausgesetzten Tabellen sowie Angaben dazu, warum die Tabellen ausgesetzt wurden.

- Replikationsverlauf (dmslogs.aws_dms_history) – Diese Tabelle enthält Informationen zum Replikationsverlauf. Diese Informationen enthalten die Anzahl und das Volumen der während der Aufgabe verarbeiteten Datensätze, die Latenz am Ende einer CDC-Aufgabe und andere Statistiken.

Die Tabelle "Apply Exceptions (Ausnahmen anwenden)" (dmslogs.aws_dms_apply_exceptions) enthält die folgenden Parameter.

Spalte	Typ	Beschreibung
TASK_NAME	nvarchar	Die Ressourcen-ID der AWS DMS Aufgabe. Die Ressourcen-ID ist im Aufgaben-ARN zu finden.
TABLE_OWNER	nvarchar	Der Eigentümer der Tabelle.
TABLE_NAME	nvarchar	Der Name der Tabelle.
ERROR_TIME	Zeitstempel	Die Uhrzeit, zu der die Ausnahme (Fehler) auftrat.
STATEMENT	nvarchar	Die Anweisung, die ausgeführt wurde, als der Fehler auftrat.
ERROR	nvarchar	Name und Beschreibung des Fehlers.

Die Tabelle „Replication Status (Replikationsstatus)“ (dmslogs.aws_dms_status) enthält den aktuellen Status der Aufgabe und der Zieldatenbank. Sie hat die folgenden Einstellungen.

Spalte	Typ	Beschreibung
SERVER_NAME	nvarchar	Der Name des Rechners, auf dem die Replikationsaufgabe ausgeführt wird.

Spalte	Typ	Beschreibung
TASK_NAME	nvarchar	Die Ressourcen-ID der AWS DMS Aufgabe. Die Ressourcen-ID ist im Aufgaben-ARN zu finden.
TASK_STATUS	varchar	<p>Einer der folgenden Werte:</p> <ul style="list-style-type: none"> • FULL LOAD • CHANGE PROCESSING (CDC) • LÄUFT NICHT <p>Aufgabenstatus ist auf FULL LOAD eingestellt, solange mindestens eine Tabelle vollständig geladen wird. Nachdem alle Tabellen geladen wurden, ändert sich der Aufgabenstatus zu CHANGE PROCESSING, wenn CDC aktiviert ist. Die Aufgabe ist auf NICHT AUSGEFÜHRT gesetzt, bevor Sie die Aufgabe starten oder nachdem die Aufgabe abgeschlossen ist.</p>
STATUS_TIME	Zeitstempel	Der Zeitstempel des Aufgabens tatus.
PENDING_CHANGES	int	Die Anzahl der Änderungs datensätze, die in der Quelldate nbank festgeschrieben und im Arbeitsspeicher und auf der Festplatte Ihrer Replikations-Insta nce zwischengespeichert wurden.

Spalte	Typ	Beschreibung
DISK_SWAP_SIZE	int	Der Speicherplatz, der für alte oder ausgelagerte Transaktionen verwendet wird.
TASK_MEMORY	int	Aktuell verwendeter Speicher in MB.
SOURCE_CURRENT_POSITION	varchar	Die Position in der Quelldatenbank, aus AWS DMS gerade gelesen wird.
SOURCE_CURRENT_TIMESTAMP	Zeitstempel	Der Zeitstempel in der Quelldatenbank, aus der gerade gelesen wird.
SOURCE_TAIL_POSITION	varchar	Die Position der ältesten Starttransaktion, die nicht ausgeführt wurde. Dieser Wert ist die neueste Position, die Sie wieder einstellen können, ohne Änderungen zu verlieren.
SOURCE_TAIL_TIMESTAMP	Zeitstempel	Der Zeitstempel der ältesten Starttransaktion, die nicht ausgeführt wurde. Dieser Wert ist der neueste Zeitstempel, den Sie wieder einstellen können, ohne Änderungen zu verlieren.
SOURCE_TIMESTAMP_APPLIED	Zeitstempel	Der Zeitstempel des letzten Transaktions-Commits. In einem Prozess der Massenanwendung ist dieser Wert der Zeitstempel für den Commit der letzten Transaktion im Stapel.

Die Tabelle „Ausgesetzte Tabellen“ (`dmslogs.aws_dms_suspended_tables`) enthält die folgenden Parameter.

Spalte	Typ	Beschreibung
SERVER_NAME	nvarchar	Der Name des Rechners, auf dem die Replikationsaufgabe ausgeführt wird.
TASK_NAME	nvarchar	Der Name der Aufgabe AWS DMS
TABLE_OWNER	nvarchar	Der Eigentümer der Tabelle.
TABLE_NAME	nvarchar	Der Name der Tabelle.
SUSPEND_REASON	nvarchar	Der Grund für die Aussetzung.
SUSPEND_TIMESTAMP	Zeitstempel	Der Zeitpunkt, zu dem die Aussetzung erfolgt ist.

Die Tabelle "Replication History (Replikationsverlauf)" (`dmslogs.aws_dms_history`) enthält die folgenden Parameter.

Spalte	Typ	Beschreibung
SERVER_NAME	nvarchar	Der Name des Rechners, auf dem die Replikationsaufgabe ausgeführt wird.
TASK_NAME	nvarchar	Die Ressourcen-ID der AWS DMS Aufgabe. Die Ressourcen-ID ist im Aufgaben-ARN zu finden.
TIMESLOT_TYPE	varchar	Einer der folgenden Werte: <ul style="list-style-type: none"> • FULL LOAD • CHANGE PROCESSING (CDC)

Spalte	Typ	Beschreibung
		Wenn die Aufgabe sowohl Full Load als auch CDC ausführt, werden zwei Verlaufsdatensätze in das Zeitfenster geschrieben.
TIMESLOT	Zeitstempel	Der letzte Zeitstempel des Zeitfensters.
TIMESLOT_DURATION	int	Die Dauer des Zeitfensters in Minuten.
TIMESLOT_LATENCY	int	Die Ziellatenz am Ende des Zeitfensters in Sekunden. Dieser Wert gilt nur für CDC-Zeitslots.
RECORDS	int	Die Anzahl der während des Zeitfensters verarbeiteten Datensätze.
TIMESLOT_VOLUME	int	Das verarbeitete Datenvolumen in MB.

Die Tabelle mit den Validierungsfehlern (`awsdms_validation_failures_v1`) enthält alle Fehler bei der Datenvalidierung für eine Aufgabe. Weitere Informationen finden Sie unter [Behebung von Datenvalidierungsfehlern](#).

Weitere Einstellungen der Kontrolltabelle sind Folgende:

- `HistoryTimeslotInMinutes` – Verwenden Sie diese Option, um die Länge der einzelnen Zeitfenster in der Replikationsverlaufstabelle anzugeben. Der Standardwert ist 5 Minuten.
- `ControlSchema`— Verwenden Sie diese Option, um den Namen des Datenbankschemas für die Steuertabellen für das AWS DMS Ziel anzugeben. Wenn Sie für diese Option keine Informationen eingeben, werden die Tabellen wie folgt an den Standardspeicherort in der Datenbank kopiert:
 - PostgreSQL, öffentlich
 - Oracle, das Zielschema
 - Microsoft SQL Server, dbo in der Zieldatenbank

- MySQL, awsdms_control
- MariaDB, awsdms_control
- Amazon Redshift, öffentlich
- DynamoDB, erstellt als einzelne Tabellen in der Datenbank
- IBM Db2 LUW, awsdms_control

Stream-Puffer-Aufgabeneinstellungen

Sie können Stream-Puffereinstellungen mithilfe von festlegen AWS CLI, einschließlich der folgenden. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

- `StreamBufferCount` – Mit dieser Option können Sie die Anzahl der Datenstrompuffer für die Migrationsaufgabe angeben. Die Standardanzahl für Stream-Puffer ist 3. Ein höherer Wert dieser Einstellung könnte das Extrahieren der Daten beschleunigen. Allerdings ist diese Leistungssteigerung in hohem Maße abhängig von der Umgebung der Migration, einschließlich dem Quellsystem und der Instance-Klasse des Replikationsservers. Der Standardwert ist für die meisten Situationen ausreichend.
- `StreamBufferSizeInMB` – Verwenden Sie diese Option, um die maximale Größe jedes Datenstrompuffers anzugeben. Die Standardgröße ist 8 MB. Möglicherweise müssen Sie den Wert für diese Option erhöhen, wenn Sie mit sehr großen LOBs arbeiten. Sie müssen möglicherweise den Wert auch dann erhöhen, wenn Sie eine Meldung in den Protokolldateien erhalten, dass die Größe des Stream-Puffers nicht ausreicht. Zur Berechnung der Größe dieser Option können Sie folgende Formel verwenden: $[\text{Max LOB size (or LOB chunk size)}] * [\text{number of LOB columns}] * [\text{number of stream buffers}] * [\text{number of tables loading in parallel per task}(\text{MaxFullLoadSubTasks})] * 3$
- `CtrlStreamBufferSizeInMB` – Verwenden Sie diese Option, um die Größe des Kontroll-Stream-Puffers festzulegen. Der Wert wird in Megabyte angegeben und kann zwischen 1 und 8 liegen. Der Standardwert ist 5. Möglicherweise müssen Sie den Wert für diese Option erhöhen, wenn Sie mit einer sehr großen Anzahl Tabellen arbeiten, etwa Zehntausende von Tabellen.

Einstellungen für die Optimierung der Verarbeitung von Änderungen

Die folgenden Einstellungen bestimmen, wie Änderungen an Zieltabellen während der Change Data Capture (CDC) AWS DMS behandelt werden. Mehrere dieser Einstellungen sind abhängig vom

Wert des Ziel-Metadatenparameters `BatchApplyEnabled`. Weitere Informationen zum Parameter `BatchApplyEnabled` erhalten Sie unter [Ziel-Metadaten-Aufgabeneinstellungen](#). Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Zu den Einstellungen für die Optimierung der Verarbeitung von Änderungen zählen die Folgenden:

Die folgenden Einstellungen gelten nur, wenn der Ziel-Metadatenparameter `BatchApplyEnabled` auf `true` festgelegt ist.

- `BatchApplyPreserveTransaction` – Wenn `true` festgelegt ist, wird die Transaktionsintegrität bewahrt und ein Stapel enthält garantiert alle Änderungen innerhalb einer Transaktion von der Quelle. Der Standardwert ist `true`. Diese Einstellung gilt nur für Oracle-Zielendpunkte.

Wenn auf `false` festgelegt, kann es zu temporären Ausfällen bei der Transaktionsintegrität kommen, um die Leistung zu verbessern. Es gibt keine Garantie, dass alle Änderungen innerhalb einer Transaktion von der Quelle in einem einzigen Stapel auf das Ziel angewendet werden.

Standardmäßig werden Änderungen in einem Transaktionsmodus AWS DMS verarbeitet, wodurch die Transaktionsintegrität gewahrt wird. Wenn Sie sich temporäre Ausfälle bei der Transaktionsintegrität leisten können, können Sie stattdessen die Option `batch optimized apply` verwenden. Bei dieser Option werden Transaktionen auf effiziente Weise gruppiert und in Stapeln angewendet, um die Effizienz zu erhöhen. Die Verwendung der `batch-optimierten Apply-Option` verstößt fast immer gegen Einschränkungen der referenziellen Integrität. Wir empfehlen daher, diese Einschränkungen während des Migrationsprozesses zu deaktivieren und im Rahmen des Cutover-Prozesses wieder zu aktivieren.

- `BatchApplyTimeoutMin`— Legt die Mindestdauer in Sekunden fest, die zwischen den einzelnen Anwendungen von AWS DMS Batch-Änderungen vergeht. Der Standardwert lautet 1.
- `BatchApplyTimeoutMax`— Legt die maximale AWS DMS Wartezeit in Sekunden zwischen den einzelnen Anwendungen von Batchänderungen fest, bevor das Timeout überschritten wird. Der Standardwert lautet 30.
- `BatchApplyMemoryLimit` – Legt die maximale Speichermenge (in MB) für die Vorverarbeitung im `Batch-optimized-Apply-Modus` fest. Der Standardwert lautet 500.
- `BatchSplitSize` – Legt die maximale Anzahl der Änderungen fest, die in einem einzigen Stapel angewendet werden. Der Standardwert 0 bedeutet, dass kein Grenzwert angewendet wird.

Die folgenden Einstellungen gelten nur, wenn der Ziel-Metadatenparameter `BatchApplyEnabled` auf `false` festgelegt ist.

- `MinTransactionSize` – Legt die minimale Anzahl der in jede Transaktion einzubeziehenden Änderungen fest. Der Standardwert lautet 1000.
- `CommitTimeout`— Legt die maximale Zeit in Sekunden fest, für die Transaktionen stapelweise gesammelt werden AWS DMS sollen, bevor ein Timeout deklariert wird. Der Standardwert lautet 1.

Bei der bidirektionalen Replikation gilt die folgende Einstellung nur, wenn der Ziel-Metadatenparameter `BatchApplyEnabled` auf `false` gesetzt ist.

- `LoopbackPreventionSettings` – Diese Einstellungen bieten für jede fortlaufende Replikationsaufgabe bei jedem Aufgabenpaar, das an einer bidirektionalen Replikation beteiligt ist, eine Verhinderung von Loopbacks. Mit der Loopback-Verhinderung wird unterbunden, dass identische Änderungen in beiden Richtungen der bidirektionalen Replikation angewendet werden, wodurch Daten fehlerhaft werden können. Weitere Hinweise zur bidirektionalen Replikation finden Sie unter [Durchführen der bidirektionalen Replikation](#).

AWS DMS versucht, Transaktionsdaten im Speicher zu behalten, bis die Transaktion vollständig an die Quelle, das Ziel oder beide übergeben wurde. Allerdings werden Transaktionen, die größer sind als der zugewiesene Speicher oder die innerhalb des festgelegten Zeitraums nicht gespeichert wurden, auf die Festplatte geschrieben.

Die folgenden Einstellungen gelten für die Optimierung der Verarbeitung von Änderungen unabhängig von dem Modus der Änderungsverarbeitung.

- `MemoryLimitTotal` – Legt die maximale Größe (in MB) fest, die alle Transaktionen im Arbeitsspeicher belegen können, bevor sie auf die Festplatte geschrieben werden. Der Standardwert lautet 1024.
- `MemoryKeepTime` – Legt die maximale Zeit in Sekunden fest, die jede Transaktion im Arbeitsspeicher bleiben kann, bevor sie auf die Festplatte geschrieben wird. Die Dauer wird ab dem Zeitpunkt berechnet, zu dem mit der Erfassung der Transaktion AWS DMS begonnen wurde. Der Standardwert lautet 60.
- `StatementCacheSize` – Legt die maximale Anzahl der vorbereiteten Anweisungen fest, die auf dem Server gespeichert werden, um später ausgeführt zu werden, wenn Änderungen auf das Ziel angewendet werden. Der Standardwert lautet 50. Der maximale Wert beträgt 200.

Das folgende Beispiel zeigt, wie Aufgabeneinstellungen für Change Processing Tuning in einer JSON-Datei für Aufgabeneinstellungen angezeigt werden:

```
"ChangeProcessingTuning": {
  "BatchApplyPreserveTransaction": true,
  "BatchApplyTimeoutMin": 1,
  "BatchApplyTimeoutMax": 30,
  "BatchApplyMemoryLimit": 500,
  "BatchSplitSize": 0,
  "MinTransactionSize": 1000,
  "CommitTimeout": 1,
  "MemoryLimitTotal": 1024,
  "MemoryKeepTime": 60,
  "StatementCacheSize": 50
}
```

Um die Häufigkeit von Schreibvorgängen auf ein Amazon-S3-Ziel während einer Datenreplikationsaufgabe zu steuern, können Sie die zusätzlichen Verbindungsattribute `cdcMaxBatchInterval` und `cdcMinFileSize` konfigurieren. Dies kann zu einer besseren Leistung bei der Analyse der Daten ohne zusätzliche überflüssige Aufgaben führen. Weitere Informationen finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#).

Aufgabeneinstellungen zur Datenvalidierung

Sie können sicherstellen, dass Ihre Daten korrekt von der Quelle zum Ziel migriert wurden. Wenn Sie die Validierung für eine Aufgabe aktivieren, AWS DMS beginnt der Vergleich der Quell- und Zieldaten unmittelbar nach dem vollständigen Laden einer Tabelle. Weitere Informationen zur Aufgabendatenvalidierung, deren Anforderungen, den Umfang des Datenbank-Supports und die gemeldeten Metriken finden Sie unter [Datenvalidierung mit AWS DMS](#). Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Zu den Datenvalidierungseinstellungen und deren Werten gehören:

- `EnableValidation` – Aktiviert die Datenvalidierung bei Einstellung von „true“. Andernfalls wird die Validierung für die Aufgabe deaktiviert. Der Standardwert ist "false".
- `ValidationMode` – Steuert, wie DMS die Daten in der Zieltabelle anhand der Quelltable validiert. AWS DMS bietet diese Einstellung für künftige Erweiterungsmöglichkeiten. Derzeit ist der

Standard- und einzig gültige Wert `ROW_LEVEL`. AWS DMS validiert alle Zeilen zwischen den Quell- und Zieltabellen.

- `FailureMaxCount` – Gibt die maximale Anzahl der Datensätze an, bei denen die Validierung fehlschlagen kann, bevor die Validierung für die Aufgabe ausgesetzt wird. Der Standardwert lautet 10.000. Wenn Sie möchten, dass die Validierung unabhängig von der Anzahl der Datensätze, bei denen die Validierung fehlgeschlagen ist, fortgesetzt wird, setzen Sie diesen Wert höher als die Anzahl der Datensätze in der Quelle.
- `HandleCollationDiff` – Wenn diese Option auf `true` gesetzt ist, werden bei der Validierung Unterschiede bei der Spaltensortierung in PostgreSQL- und Microsoft-SQL-Server-Endpunkten berücksichtigt, wenn zu vergleichende Quellen- und Zieldatensätze identifiziert werden. Andernfalls werden jegliche Unterschiede bei der Spaltensortierung für die Validierung ignoriert. Die Spaltensortierung kann die Reihenfolge der Zeilen bestimmen, was für die Datenvalidierung wichtig ist. Das Festlegen von `HandleCollationDiff` auf "true" behebt diese Sortierungsunterschiede automatisch und verhindert Fehlalarme bei der Datenvalidierung. Der Standardwert ist `false`.
- `RecordFailureDelayInMinutes` – Gibt die Verzögerung in Minuten an, bevor Informationen über das Fehlschlagen der Validierung gemeldet werden.
- `RecordFailureDelayLimitInMinutes` – Gibt die Verzögerung an, bevor Informationen über das Fehlschlagen der Validierung gemeldet werden. Normalerweise verwendet AWS DMS die Aufgabenlatenz, um die tatsächliche Verzögerung von Änderungen, die am Ziel durchgeführt werden, zu erkennen und Fehlalarme zu verhindern. Diese Einstellung überschreibt den tatsächlichen Verzögerungswert und Sie können eine größere Verzögerung festlegen, bevor Validierungsmetriken gemeldet werden. Der Standardwert lautet 0.
- `RecordSuspendDelayInMinutes` – Gibt die Wartezeit in Minuten an, bevor die Validierung von Tabellen aufgrund des in `FailureMaxCount` festgelegten Fehlerschwellenwerts ausgesetzt wird.
- `SkipLobColumns`— Wenn diese Option auf `true` gesetzt ist, wird die Datenvalidierung für alle LOB-Spalten im Teil der Aufgabenvalidierung AWS DMS übersprungen. Der Standardwert ist `false`.
- `TableFailureMaxCount` – Gibt die maximale Anzahl der Zeilen in einer Tabelle an, bei denen die Validierung fehlschlagen kann, bevor sie für die Tabelle ausgesetzt wird. Der Standardwert lautet 1.000.
- `ThreadCount`— Gibt die Anzahl der Ausführungs-Threads an, die bei der Validierung AWS DMS verwendet werden. Jeder Thread wählt `not-yet-validated` Daten aus der Quelle und dem Ziel aus, um sie zu vergleichen und zu validieren. Der Standardwert ist 5. Wenn Sie eine `ThreadCount` höhere Zahl angeben, AWS DMS kann die Validierung schneller abgeschlossen werden. Allerdings

führt AWS DMS dann auch mehr simultane Abfragen aus, sodass bei Quelle und Ziel mehr Ressourcen verbraucht werden.

- `ValidationOnly` – Wenn diese Option auf `true` gesetzt ist, führt die Aufgabe eine Datenvalidierung durch, ohne eine Migration oder Replikation von Daten vorzunehmen. Der Standardwert ist `false`. Sie können die Einstellung `ValidationOnly` nicht ändern, nachdem die Aufgabe erstellt wurde.

Sie müssen `TargetTablePrepMode` auf `DO_NOTHING` (die Standardeinstellung für reine Validierungsaufgaben) und den Migrationstyp auf einen der folgenden Werte festlegen:

- **Volllast** — Stellen Sie den Migrationstyp der Aufgabe in der AWS DMS Konsole auf **Vorhandene Daten migrieren ein**. Oder legen Sie in der AWS DMS API den Migrationstyp auf `FULL-LOAD` fest.
- **CDC** – Legen Sie für Migrationstyp in der AWS DMS -Konsole **Nur Replizieren von Datenänderungen** fest. Oder legen Sie in der AWS DMS API den Migrationstyp auf `CDC` fest.

Unabhängig vom ausgewählten Migrationstyp werden Daten während einer reinen Validierungsaufgabe nicht wirklich migriert oder repliziert.

Weitere Informationen finden Sie unter [Reine Validierungsaufgaben](#).

Important

Die Einstellung `ValidationOnly` ist unveränderlich. Nach dem Erstellen einer Aufgabe kann sie für diese Aufgabe nicht mehr geändert werden.

- `ValidationPartialLobSize` – Gibt an, ob Sie eine teilweise Validierung für LOB-Spalten durchführen möchten, anstatt alle in der Spalte gespeicherten Daten zu validieren. Dies ist möglicherweise nützlich, wenn Sie nur einen Teil der LOB-Daten und nicht den gesamten LOB-Datensatz migrieren. Der Wert wird in KB-Einheiten angegeben. Der Standardwert ist 0. Dies bedeutet, dass AWS DMS alle LOB-Spaltendaten validiert. "`ValidationPartialLobSize`": 32Bedeutet beispielsweise, dass AWS DMS nur die ersten 32 KB der Spaltendaten sowohl in der Quelle als auch im Ziel validiert werden.
- `PartitionSize` – Gibt die Stapelgröße der Datensätze an, die für den Vergleich aus der Quelle und dem Ziel gelesen werden sollen. Der Standardwert ist 10 000.
- `ValidationQueryCdcDelaySeconds` – Die Zeitspanne, um die sich die erste Validierungsabfrage bei jeder CDC-Aktualisierung sowohl an der Quelle als auch am Ziel verzögert. Bei einer hohen Migrationslatenz kann diese Einstellung dazu beitragen, Ressourcenkonflikte

zu reduzieren. Bei einer reinen Validierungsaufgabe werden für diese Option automatisch 180 Sekunden festgelegt. Der Standardwert ist 0.

Beispielsweise kann mit der folgenden JSON die Datenvalidierung mit der doppelten Standard-Anzahl an Threads ausgeführt werden. Außerdem kommt dies für die Unterschiede in der Reihenfolge von Datensätzen auf, die durch Unterschiede bei der Spaltensortierung in PostgreSQL-Endpunkten bedingt wurden. Auch wird eine Validierungs-Berichtverzögerung bereitgestellt, um für zusätzliche Zeit aufzukommen, die benötigt wird, um Validierungsfehler zu verarbeiten.

```
"ValidationSettings": {
  "EnableValidation": true,
  "ThreadCount": 10,
  "HandleCollationDiff": true,
  "RecordFailureDelayLimitInMinutes": 30
}
```

Note

AWS DMS verwendet für einen Oracle-Endpunkt DBMS_CRYPTO, um BLOBs zu validieren. Wenn Ihr Oracle-Endpunkt BLOBs verwendet, erteilen Sie dem Benutzerkonto, das auf den Oracle-Endpunkt zugreift, die Berechtigung execute für DBMS_CRYPTO. Führen Sie dazu die folgende Anweisung aus.

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

Aufgabeneinstellungen für den Umgang mit der DDL-Änderungsverarbeitung

Die folgenden Einstellungen bestimmen, wie DDL-Änderungen (Data Definition Language) für Zieltabellen während der Change Data Capture (CDC) AWS DMS behandelt werden. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Zu den Aufgabeneinstellungen zur Handhabung der Änderungsverarbeitungs-DDL gehören folgende:

- `HandleSourceTableDropped` – Legen Sie diese Option auf `true` fest, um die Zieltabelle zu löschen, wenn die Quelltable gelöscht wird.
- `HandleSourceTableTruncated` – Setzen Sie diese Option auf `true`, um die Zieltabelle zu kürzen, wenn die Quelltable gekürzt wird.
- `HandleSourceTableAltered` – Setzen Sie diese Option auf `true`, um die Zieltabelle zu ändern, wenn die Quelltable geändert wird.

Es folgt ein Beispiel dafür, wie Aufgabeneinstellungen, die Änderungsverarbeitungs-DDL verarbeiten, in einer JSON-Datei für die Aufgabeneinstellung angezeigt werden:

```
"ChangeProcessingDdlHandlingPolicy": {  
  "HandleSourceTableDropped": true,  
  "HandleSourceTableTruncated": true,  
  "HandleSourceTableAltered": true  
},
```

Note

Weitere Informationen dazu, welche DDL-Anweisungen für einen bestimmten Endpunkt unterstützt werden, finden Sie in dem Abschnitt zur Beschreibung des betreffenden Endpunkts.

Einstellungen der Zeichenersetzungsaufgabe

Sie können angeben, dass Ihre Replikationsaufgabe Zeichenersetzungen in der Zieldatenbank für alle Quelldatenbankspalten mit dem Datentyp AWS DMS STRING oder WSTRING durchführt. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Sie können die Zeichenersetzung für jede Aufgabe mit Endpunkten aus den folgenden Quell- und Zieldatenbanken konfigurieren:

- Quelldatenbanken:
 - Oracle
 - Microsoft SQL Server

- MySQL
- PostgreSQL
- SAP Adaptive Server Enterprise (ASE)
- IBM Db2 (LUW)
- Zieldatenbanken:
 - Oracle
 - Microsoft SQL Server
 - MySQL
 - PostgreSQL
 - SAP Adaptive Server Enterprise (ASE)
 - Amazon-Redshift

Sie können Zeichenersetzungen mithilfe des Parameters `CharacterSetSettings` in Ihren Aufgabeneinstellungen angeben. Diese Zeichenersetzungen erfolgen für Zeichen, die mit dem Unicode-Codepunkt in Hexadezimalnotation angegeben werden. Sie können die Ersetzungen in zwei Phasen in der folgenden Reihenfolge implementieren, wenn beide angegeben sind:

1. Individuelle Zeichenersetzung — AWS DMS kann die Werte ausgewählter Zeichen in der Quelle durch die angegebenen Ersatzwerte der entsprechenden Zeichen auf dem Ziel ersetzen. Verwenden Sie das `CharacterReplacements`-Array in `CharacterSetSettings`, um alle Quellzeichen mit den von Ihnen angegebenen Unicode-Codepunkten auszuwählen. Verwenden Sie dieses Array auch, um die Ersatz-Codepunkte für die entsprechenden Zeichen auf dem Ziel anzugeben.

Um alle Zeichen in der Quelle auszuwählen, die einen bestimmten Codepunkt haben, legen Sie eine Instance von `SourceCharacterCodePoint` im `CharacterReplacements`-Array auf diesen Codepunkt fest. Geben Sie dann den Ersatz-Codepunkt für alle gleichwertigen Zielzeichen an, indem Sie die entsprechende Instance von `TargetCharacterCodePoint` in diesem Array festlegen. Um Zielzeichen zu löschen, anstatt sie zu ersetzen, setzen Sie die entsprechenden Instances von `TargetCharacterCodePoint` auf Null (0). Sie können beliebig viele verschiedene Werte von Zielzeichen ersetzen oder löschen, indem Sie zusätzliche Paare von `SourceCharacterCodePoint`- und `TargetCharacterCodePoint`-Einstellungen im `CharacterReplacements`-Array angeben. Wenn Sie denselben Wert für mehrere Instances von `SourceCharacterCodePoint` angeben, gilt der Wert der letzten entsprechenden Einstellung von `TargetCharacterCodePoint` für das Ziel.

Angenommen, Sie geben z. B. die folgenden Werte für `CharacterReplacements` an.

```
"CharacterSetSettings": {
  "CharacterReplacements": [ {
    "SourceCharacterCodePoint": 62,
    "TargetCharacterCodePoint": 61
  }, {
    "SourceCharacterCodePoint": 42,
    "TargetCharacterCodePoint": 41
  }
]
}
```

In diesem Beispiel werden alle Zeichen mit dem Quellcodepunkt-Hexadezimalwert 62 auf dem Ziel durch Zeichen mit dem Codepunktwert 61 AWS DMS ersetzt. AWS DMS Ersetzt außerdem alle Zeichen mit dem Quellcodepunkt 42 auf dem Ziel durch Zeichen mit dem Codepunktwert 41. Anders gesagt, ersetzt AWS DMS alle Instances des Buchstaben ' b ' auf dem Ziel durch den Buchstaben ' a '. AWS DMS Ersetzt auf ähnliche Weise alle Vorkommen des Buchstabens ' B ' auf dem Ziel durch den Buchstaben ' A '.

2. Überprüfung und Ersetzung des Zeichensatzes — Nachdem alle einzelnen Zeichen ersetzt wurden, AWS DMS kann sichergestellt werden, dass alle Zielzeichen gültige Unicode-Codepunkte in dem von Ihnen angegebenen Zeichensatz enthalten. Sie verwenden `CharacterSetSupport` in `CharacterSetSettings`, um diese Verifizierung und Modifizierung von Zielzeichen zu konfigurieren. Um den Verifizierungszeichensatz anzugeben, legen Sie `CharacterSet` in `CharacterSetSupport` auf den Zeichenfolgenwert des Zeichensatzes fest. (Nachstehend finden Sie die möglichen Werte für `CharacterSet`.) Sie können die ungültigen Zielzeichen auf eine der folgenden Arten AWS DMS ändern:

- Geben Sie einen einzigen Ersatz-Unicode-Codepunkt für alle ungültigen Zielzeichen an, unabhängig von ihrem aktuellen Codepunkt. Um diesen Ersatz-Codepunkt zu konfigurieren, setzen Sie `ReplaceWithCharacterCodePoint` in `CharacterSetSupport` auf den angegebenen Wert.
- Konfigurieren Sie das Löschen aller ungültigen Zielzeichen, indem Sie `ReplaceWithCharacterCodePoint` auf Null (0) setzen.

Angenommen, Sie geben z. B. die folgenden Werte für `CharacterSetSupport` an.

```
"CharacterSetSettings": {
```

```

"CharacterSetSupport": {
  "CharacterSet": "UTF16_PlatformEndian",
  "ReplaceWithCharacterCodePoint": 0
}
}

```

In diesem Beispiel AWS DMS werden alle Zeichen auf dem Ziel gelöscht, die im "UTF16_PlatformEndian" Zeichensatz ungültig sind. Daher werden alle mit dem Hexadezimalwert 2FB6 angegebenen Zeichen gelöscht. Dieser Wert ist ungültig, da es sich um einen 4-Byte-Unicode-Codepunkt handelt und UTF16-Zeichensätze nur Zeichen mit 2-Byte-Codepunkten akzeptieren.

Note

Die Replikationsaufgabe schließt alle angegebenen Zeichenersetzungen ab, bevor globale Transformationen oder Transformationen auf Tabellenebene gestartet werden, die Sie über die Tabellenzuordnung angeben. Weitere Informationen zur Tabellenzuweisung finden Sie unter [Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen](#).

Die Zeichenersetzung unterstützt keine LOB-Datentypen. Dazu gehören alle Datentypen, die DMS als LOB-Datentyp betrachtet. Beispielsweise wird der Datentyp Extended in Oracle als LOB betrachtet. Weitere Hinweise zu Quelldatentypen finden Sie unter [Quelldatentypen für Oracle](#).

Die Werte, die für AWS DMS CharacterSet unterstützen, sind in der folgenden Tabelle aufgeführt.

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

Vorher-Abbild-Aufgabeneinstellungen

Wenn Sie CDC-Aktualisierungen auf ein Daten-Streaming-Ziel wie Kinesis oder Apache Kafka schreiben, können Sie die ursprünglichen Werte einer Zeile in der Quelldatenbank anzeigen, bevor sie durch eine Aktualisierung geändert werden. Um dies zu ermöglichen, AWS DMS füllt es ein Vorher-Abbild der Aktualisierungsereignisse auf der Grundlage von Daten aus, die von der Quelldatenbank-Engine bereitgestellt werden. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Hierfür verwenden Sie den Parameter `BeforeImageSettings`, der jeder Aktualisierungsoperation ein neues JSON-Attribut hinzufügt, wobei Werte aus dem Quelldatenbanksystem erfasst werden.

Stellen Sie sicher, dass Sie `BeforeImageSettings` nur auf Aufgaben für vollständiges Laden plus CDC oder auf reine CDC-Aufgaben anwenden. Bei Aufgaben für vollständiges Laden plus CDC werden vorhandene Daten migriert und fortlaufende Änderungen repliziert. Bei reinen CDC-Aufgaben werden nur Datenänderungen repliziert.

Wenden Sie `BeforeImageSettings` nicht auf Nur-Volllast-Aufgaben an.

Für `BeforeImageSettings` sind folgende Optionen möglich:

- `EnableBeforeImage` – Aktiviert Vorher-Abbilder, wenn auf `true` gesetzt. Der Standardwert ist `false`.
- `FieldName` – Weist dem neuen JSON-Attribut einen Namen zu. Wann `EnableBeforeImage true` ist, ist `FieldName` erforderlich und darf nicht leer sein.
- `ColumnFilter` – Gibt eine Spalte an, die mithilfe von Vorher-Abbildern hinzugefügt werden soll. Wenn Sie nur Spalten hinzufügen möchten, die Teil der Primärschlüssel der Tabelle sind, verwenden Sie den Standardwert `pk-only`. Wenn Sie eine Spalte hinzufügen möchten, die einen Vorher-Abbild-Wert hat, verwenden Sie `all`. Beachten Sie, dass das Vorher-Abbild keine LOB-Datentypen (LOB = Large Binary Object) wie CLOB und BLOB unterstützt.

Das folgende Beispiel veranschaulicht die Verwendung von `BeforeImageSettings`.

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```

Weitere Informationen zu den Vorher-Abbild-Einstellungen für Kinesis, einschließlich zusätzlicher Tabellenzuweisungseinstellungen, finden Sie unter [Verwenden eines Vorher-Abbilds zum Anzeigen von Originalwerten von CDC-Zeilen für einen Kinesis-Datenstrom als Ziel](#).

Weitere Informationen zu den Vorher-Abbild-Einstellungen für Kafka, einschließlich zusätzlicher Tabellenzuordnungseinstellungen, finden Sie unter [Verwenden eines Vorher-Abbilds zum Anzeigen von Originalwerten von CDC-Zeilen für Apache Kafka als Ziel](#).

Aufgabeneinstellungen zur Fehlerbehandlung

Sie können das Fehlerbehandlungsverhalten der Replikationsaufgabe mit den folgenden Einstellungen festlegen. Informationen zur Verwendung einer Aufgabenkonfigurationsdatei zum Festlegen von Aufgabeneinstellungen finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

- `DataErrorPolicy`— Legt fest, welche Aktion AWS DMS ergreift, wenn ein Fehler im Zusammenhang mit der Datenverarbeitung auf Datensatzebene auftritt. Einige Beispiele für Fehler bei der Datenverarbeitung sind Konvertierungsfehler, Fehler bei der Transformation und ungültige Daten. Der Standardwert ist `LOG_ERROR`.
 - `IGNORE_RECORD` – Die Aufgabe wird fortgesetzt und die Daten für diesen Datensatz werden ignoriert. Der Fehlerzähler für die Eigenschaft `DataErrorEscalationCount` wird erhöht. Wenn Sie also ein Limit für Fehler für eine Tabelle festlegen, wird dieser Fehler auf dieses Limit angerechnet.
 - `LOG_ERROR` – Die Aufgabe wird fortgesetzt und der Fehler wird in das Aufgabenprotokoll geschrieben.
 - `SUSPEND_TABLE` – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
 - `STOP_TASK` – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
- `DataTruncationErrorPolicy` – Legt die Aktion fest, die AWS DMS ausführt, wenn Daten gekürzt sind. Der Standardwert ist `LOG_ERROR`.

- **IGNORE_RECORD** – Die Aufgabe wird fortgesetzt und die Daten für diesen Datensatz werden ignoriert. Der Fehlerzähler für die Eigenschaft `DataErrorEscalationCount` wird erhöht. Wenn Sie also ein Limit für Fehler für eine Tabelle festlegen, wird dieser Fehler auf dieses Limit angerechnet.
- **LOG_ERROR** – Die Aufgabe wird fortgesetzt und der Fehler wird in das Aufgabenprotokoll geschrieben.
- **SUSPEND_TABLE** – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
- **STOP_TASK** – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
- **DataErrorEscalationPolicy** – Legt die Aktion fest, die AWS DMS ausführt, wenn die maximale Anzahl an Fehlern (im Parameter `DataErrorEscalationCount` festgelegt) erreicht ist. Der Standardwert ist **SUSPEND_TABLE**.
 - **SUSPEND_TABLE** – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
 - **STOP_TASK** – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
- **DataErrorEscalationCount** – Legt die maximale Anzahl an Fehlern fest, die für einen bestimmten Datensatz auftreten können. Wenn diese Anzahl erreicht ist, werden die Daten für die Tabelle, die den fehlerhaften Datensatz enthält, gemäß der Richtlinie behandelt, die in `DataErrorEscalationPolicy` festgelegt ist. Der Standardwert ist 0.
- **EventErrorPolicy**— Legt fest, welche Aktion AWS DMS ergreift, wenn beim Senden eines aufgabenbezogenen Ereignisses ein Fehler auftritt. Mögliche Werte
 - **IGNORE** – Die Aufgabe wird fortgesetzt und alle mit diesem Ereignis verknüpften Daten werden ignoriert.
 - **STOP_TASK** – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
- **TableErrorPolicy** – Legt die Aktion fest, die AWS DMS ausführt, wenn bei der Verarbeitung von Daten oder Metadaten für eine bestimmte Tabelle ein Fehler auftritt. Dieser Fehler gilt nur für allgemeine Tabellendaten und ist kein Fehler, der sich auf einen bestimmten Datensatz bezieht. Der Standardwert ist **SUSPEND_TABLE**.
 - **SUSPEND_TABLE** – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
 - **STOP_TASK** – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.

- `TableErrorEscalationPolicy` – Legt die Aktion fest, die AWS DMS ausführt, wenn die maximale Anzahl an Fehlern (über den Parameter `TableErrorEscalationCount` festgelegt) erreicht ist. Die Standard- und einzige Benutzereinstellung ist `STOP_TASK`: Die Aufgabe wird angehalten und ein manueller Eingriff ist erforderlich.
- `TableErrorEscalationCount` – Die maximale Anzahl an Fehlern, die bei allgemeinen Daten oder Metadaten für eine bestimmte Tabelle auftreten können. Wenn diese Anzahl erreicht ist, werden die Daten für die Tabelle gemäß der Richtlinie behandelt, die in `TableErrorEscalationPolicy` festgelegt wurde. Der Standardwert ist 0.
- `RecoverableErrorCount` – Die maximale Anzahl an Versuchen zum Neustart einer Aufgabe, wenn ein Umgebungsfehler auftritt. Nachdem das System so oft wie angegeben versucht hat, die Aufgabe neu zu starten, wird die Aufgabe angehalten und ein manueller Eingriff ist erforderlich. Der Standardwert ist -1, was bedeutet, dass versucht werden soll, die Aufgabe AWS DMS auf unbestimmte Zeit neu zu starten. Legen Sie diesen Wert auf 0 fest, um nie zu versuchen, eine Aufgabe neu zu starten. Tritt ein schwerwiegender Fehler auf, wird der Versuch, die Aufgabe neu zu starten, nach sechs Versuchen AWS DMS beendet.
- `RecoverableErrorInterval`— Die Anzahl der Sekunden, die AWS DMS zwischen den Versuchen wartet, eine Aufgabe neu zu starten. Der Standardwert ist 5.
- `RecoverableErrorThrottling` – Wenn diese Option aktiviert ist, wird das Intervall zwischen den Versuchen zum Neustart einer Aufgabe in einer Reihe verlängert, basierend auf dem Wert von `RecoverableErrorInterval`. Wenn für `RecoverableErrorInterval` beispielsweise 5 Sekunden festgelegt sind, erfolgt der nächste Wiederholungsversuch nach 10 Sekunden, dann nach 20 Sekunden, dann nach 40 Sekunden usw. Der Standardwert ist `true`.
- `RecoverableErrorThrottlingMax`— Die maximale Anzahl von Sekunden, die AWS DMS zwischen Versuchen wartet, eine Aufgabe neu zu starten, falls diese Option aktiviert ist. `RecoverableErrorThrottling` Der Standardwert ist 1800.
- `RecoverableErrorStopRetryAfterThrottlingMax`— Wenn diese Option auf `true` gesetzt ist, wird der Neustart der Aufgabe beendet, sobald die maximale Anzahl von Sekunden zwischen Wiederherstellungsversuchen AWS DMS erreicht ist, pro. `RecoverableErrorThrottlingMax`
- `ApplyErrorDeletePolicy` – Legt fest, welche Aktion AWS DMS ausführt, wenn es zu einem Konflikt mit einem DELETE-Vorgang kommt. Der Standardwert ist `IGNORE_RECORD`. Folgende Werte sind möglich:
 - `IGNORE_RECORD` – Die Aufgabe wird fortgesetzt und die Daten für diesen Datensatz werden ignoriert. Der Fehlerzähler für die Eigenschaft `ApplyErrorEscalationCount` wird erhöht.

Wenn Sie also ein Limit für Fehler für eine Tabelle festlegen, wird dieser Fehler auf dieses Limit angerechnet.

- **LOG_ERROR** – Die Aufgabe wird fortgesetzt und der Fehler wird in das Aufgabenprotokoll geschrieben.
- **SUSPEND_TABLE** – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
- **STOP_TASK** – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
- **ApplyErrorInsertPolicy** – Legt fest, welche Aktion AWS DMS ausführt, wenn es zu einem Konflikt mit einem INSERT-Vorgang kommt. Der Standardwert ist LOG_ERROR. Folgende Werte sind möglich:
 - **IGNORE_RECORD** – Die Aufgabe wird fortgesetzt und die Daten für diesen Datensatz werden ignoriert. Der Fehlerzähler für die Eigenschaft `ApplyErrorEscalationCount` wird erhöht. Wenn Sie also ein Limit für Fehler für eine Tabelle festlegen, wird dieser Fehler auf dieses Limit angerechnet.
 - **LOG_ERROR** – Die Aufgabe wird fortgesetzt und der Fehler wird in das Aufgabenprotokoll geschrieben.
 - **SUSPEND_TABLE** – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
 - **STOP_TASK** – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
 - **INSERT_RECORD** – Wenn ein bereits bestehender Zieldatensatz denselben Primärschlüssel wie der eingefügte Quelldatensatz aufweist, wird der Zieldatensatz aktualisiert.
- **ApplyErrorUpdatePolicy** – Legt fest, welche Aktion AWS DMS ausführt, wenn es zu einem Konflikt bezüglich fehlender Daten mit einem UPDATE-Vorgang kommt. Der Standardwert ist LOG_ERROR. Folgende Werte sind möglich:
 - **IGNORE_RECORD** – Die Aufgabe wird fortgesetzt und die Daten für diesen Datensatz werden ignoriert. Der Fehlerzähler für die Eigenschaft `ApplyErrorEscalationCount` wird erhöht. Wenn Sie also ein Limit für Fehler für eine Tabelle festlegen, wird dieser Fehler auf dieses Limit angerechnet.
 - **LOG_ERROR** – Die Aufgabe wird fortgesetzt und der Fehler wird in das Aufgabenprotokoll geschrieben.

- `SUSPEND_TABLE` – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
- `STOP_TASK` – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
- `UPDATE_RECORD`— Wenn der Zieldatensatz fehlt, wird der fehlende Zieldatensatz in die Zieltabelle eingefügt. AWS DMS deaktiviert die LOB-Spaltenunterstützung für die Aufgabe vollständig. Wenn diese Option ausgewählt wird, muss die vollständige ergänzende Protokollierung für alle Quelltabellenspalten aktiviert werden, wenn Oracle die Quelldatenbank ist.
- `ApplyErrorEscalationPolicy`— Legt fest, welche Aktion AWS DMS ergreift, wenn die maximale Anzahl von Fehlern (mit dem `ApplyErrorEscalationCount` Parameter festgelegt) erreicht ist. Die Standardeinstellung ist `LOG_ERROR`:
 - `LOG_ERROR` – Die Aufgabe wird fortgesetzt und der Fehler wird in das Aufgabenprotokoll geschrieben.
 - `SUSPEND_TABLE` – Die Aufgabe wird fortgesetzt, Daten aus der Tabelle mit dem fehlerhaften Datensatz werden jedoch in einen Fehlerstatus verschoben und die Daten werden nicht repliziert.
 - `STOP_TASK` – Die Aufgabe wird gestoppt und ein manuelles Eingreifen ist erforderlich.
- `ApplyErrorEscalationCount` – Legt die maximale Anzahl der `APPLY`-Konflikte fest, die während einer Prozessänderung für eine bestimmte Tabelle auftreten können. Wenn diese Anzahl erreicht ist, werden die Daten für die Tabelle gemäß der Richtlinie behandelt, die im Parameter `ApplyErrorEscalationPolicy` festgelegt wurde. Der Standardwert ist 0.
- `ApplyErrorFailOnTruncationDdl` – Legen Sie für diese Option `true` fest, damit die Aufgabe fehlschlägt, wenn an einer der nachverfolgten Tabellen während CDC eine Kürzung vorgenommen wird. Der Standardwert ist `false`.

Dieser Ansatz funktioniert nicht mit PostgreSQL Version 11.x oder niedriger oder einem anderen Quellendpunkt, der die DDL-Tabellenkürzung nicht repliziert.

- `FailOnNoTablesCaptured` – Legen Sie für diese Option `true` fest, damit die Aufgabe fehlschlägt, wenn die für eine Aufgabe definierten Tabellenzuweisungen beim Start der Aufgabe keine Tabellen finden. Der Standardwert ist `false`.
- `FailOnTransactionConsistencyBreached` – Diese Option gilt für Aufgaben, die für CDC Oracle als Quelle verwenden. Der Standardwert lautet „false“. Geben Sie dies mit `true` an, damit

eine Aufgabe fehlschlägt, wenn eine Transaktion länger als der angegebene Timeoutwert geöffnet ist und verworfen werden kann.

Wenn eine CDC-Aufgabe mit Oracle gestartet wird, AWS DMS wartet eine begrenzte Zeit, bis die älteste offene Transaktion geschlossen ist, bevor CDC gestartet wird. Wenn die älteste offene Transaktion erst geschlossen wird, wenn das Timeout erreicht ist, wird CDC in den meisten Fällen AWS DMS gestartet, wobei diese Transaktion ignoriert wird. Wenn diese Option auf `true` gesetzt ist, schlägt die Aufgabe fehl.

- `FullLoadIgnoreConflicts`— Stellen Sie diese Option so ein, `true` dass die Fehler „Keine betroffenen Zeilen“ und „Duplikate“ bei der Anwendung zwischengespeicherter Ereignisse AWS DMS ignoriert werden. Wenn diese Option auf `false` gesetzt ist, werden alle Fehler AWS DMS gemeldet, anstatt sie zu ignorieren. Der Standardwert ist `true`.

Beachten Sie, dass Fehler beim Laden von Tabellen in Redshift als Ziel unter `STL_LOAD_ERRORS` gemeldet werden. Weitere Informationen finden Sie unter [STL_LOAD_ERRORS](#) im Datenbankentwicklerhandbuch für Amazon Redshift.

Speichern der Aufgabeneinstellungen

Sie können die Aufgabeneinstellungen als JSON-Datei speichern, falls Sie die Einstellungen für eine andere Aufgabe wiederverwenden möchten. Aufgabeneinstellungen zum Kopieren in eine JSON-Datei finden Sie im Abschnitt [Übersichtsdetails einer Aufgabe](#).

Note

Wenn Sie Aufgabeneinstellungen für andere Aufgaben wiederverwenden, entfernen Sie alle Attribute `CloudWatchLogGroup` und `CloudWatchLogStream`. Andernfalls wird der folgende Fehler angezeigt: `SYSTEM ERROR MESSAGE:Task Settings CloudWatchLogGroup oder cannot be CloudWatchLogStream cannot be set on create.`

Zum Beispiel enthält die folgende JSON-Datei gespeicherte Einstellungen für eine Aufgabe.

```
{
  "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
}

```

Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS

Große binäre Objekte (LOBs) lassen sich manchmal nur schwierig zwischen Systemen migrieren. AWS DMS bietet eine Reihe von Optionen, die Ihnen bei der Optimierung von LOB-Spalten helfen. Informationen darüber, welche und wann Datentypen als LOBs betrachtet werden AWS DMS, finden Sie in der AWS DMS Dokumentation.

Beim Migrieren von Daten von einer Datenbank in eine andere, sollten Sie überdenken, wie Ihre LOBs gespeichert werden. Dies gilt insbesondere für heterogene Migrationen. Wenn Sie dies möchten, müssen Sie die LOB-Daten nicht migrieren.

Wenn Sie sich für LOBs entschieden haben, können Sie die anderen LOB-Einstellungen festlegen:


- Der LOB-Modus bestimmt, wie LOBs verarbeitet werden:
 - Vollständiger LOB-Modus — Im vollständigen LOB-Modus werden alle LOBs unabhängig von ihrer Größe von der Quelle zum Ziel AWS DMS migriert. In dieser Konfiguration sind keine Informationen über die zu AWS DMS erwartende maximale Größe der zu erwartenden LOBs vorhanden. Folglich werden die LOBs einzeln migriert. Der vollständige LOB-Modus kann recht langsam sein.
 - Limitierter LOB-Modus – Im limitierten LOB-Modus können Sie eine maximale LOB-Größe festlegen, die von DMS akzeptiert werden sollte. Dies ermöglicht es DMS, vorab Speicher zuzuweisen und die LOB-Daten massenweise zu laden. LOBs, die die maximale LOB-Größe überschreiten, werden abgeschnitten und in der Protokolldatei wird eine entsprechende Warnung ausgegeben. Im limitierten LOB-Modus kann gegenüber dem vollständigen LOB-Modus eine erhebliche Leistungssteigerung erzielt werden. Wir empfehlen Ihnen, nach Möglichkeit den eingeschränkten LOB-Modus zu verwenden. Der empfohlene Höchstwert liegt bei 102 400 Kilobyte (100 MB).

Note

Wenn die Option „Maximale LOB-Größe (KB)“ mit einem Wert von mehr als 63 KB verwendet wird, beeinträchtigt dies die Leistung eines vollständigen Ladevorgangs, der für die Ausführung im limitierten LOB-Modus konfiguriert ist. Beim vollständigen Laden weist DMS Speicher zu, indem der Wert für die maximale LOB-Größe (KB) mit der Commit-Rate multipliziert und das Produkt mit der Anzahl der LOB-Spalten multipliziert wird. Wenn DMS diesen Speicher nicht vorab zuweisen kann, beginnt DMS, SWAP-Speicher zu verbrauchen. Dies wirkt sich auf die Leistung beim vollständigen Laden aus. Wenn Sie also bei Verwendung des limitierten LOB-Modus Leistungsprobleme feststellen, sollten Sie eine Verringerung der Commit-Rate in Betracht ziehen, bis Sie ein akzeptables Leistungsniveau erreichen. Sie können auch darüber nachdenken, den Inline-LOB-Modus für unterstützte Endpunkte zu verwenden, sobald Sie die LOB-Verteilung für die Tabelle kennen.


Um die begrenzte LOB-Größe zu überprüfen, müssen Sie für `ValidationPartialLobSize` denselben Wert wie für `LobMaxSize (KB)` angeben.

- **Inline-LOB-Modus** – Im Inline-LOB-Modus legen Sie die maximale LOB-Größe fest, die DMS inline überträgt. LOBs, die kleiner als die angegebene Größe sind, werden inline übertragen. LOBs, die größer als die angegebene Größe sind, werden im vollständigen LOB-Modus repliziert. Sie können diese Option für die Replikation kleiner und großer LOBs auswählen, wenn die meisten LOBs klein sind. Für Endpunkte, die den vollständigen LOB-Modus nicht unterstützen, wie S3 und Redshift, unterstützt DMS den Inline-LOB-Modus nicht.

 Note

Bei Oracle werden LOBs als VARCHAR-Datentypen behandelt, wann immer dies möglich ist. Dieser Ansatz bedeutet, dass sie in großen Mengen aus der Datenbank AWS DMS abgerufen werden, was deutlich schneller ist als bei anderen Methoden. Die maximale Größe eines VARCHAR in Oracle beträgt 32 K. Wenn Oracle Ihre Quelldatenbank ist, ist eine auf weniger als 32 K begrenzte LOB-Größe daher optimal.

- Wenn eine Aufgabe zur Ausführung im eingeschränkten LOB-Modus konfiguriert wurde, wird mit der Option `Max LOB size (K)` (Maximale LOB-Größe (K)) die maximale LOB-Größe festgelegt, die von AWS DMS akzeptiert wird. Alle LOBs, die diesen Wert überschreiten, werden auf diesen Wert gekürzt.
- Wenn eine Aufgabe für die Verwendung des vollständigen LOB-Modus konfiguriert ist, werden LOBs stückweise AWS DMS abgerufen. Die Option `LOB chunk size (K)` (LOB-Teilegröße) bestimmt die Größe der einzelnen Teile. Wenn Sie diese Option festlegen, achten Sie besonders auf die maximale Paketgröße, die gemäß Ihrer Netzwerkkonfiguration zulässig ist. Wenn die LOB-Blockgröße die maximal zulässige Paketgröße überschreitet, erhalten Sie möglicherweise Trennungsfehler. Der empfohlene Wert für `LobChunkSize` ist 64 Kilobyte. Wenn für `LobChunkSize` ein höherer Wert als 64 Kilobyte verwendet wird, kann dies zu Aufgabenfehlern führen.
- Wenn eine Aufgabe für die Ausführung im Inline-LOB-Modus konfiguriert ist, bestimmt die Einstellung `InLineLobMaxSize`, welche LOBs DMS inline überträgt.

 Note

Sie können LOB-Datentypen nur mit Tabellen und Ansichten verwenden, die einen Primärschlüssel enthalten.

Weitere Informationen zu den Aufgabeneinstellungen zum Angeben dieser Optionen finden Sie unter [Ziel-Metadaten-Aufgabeneinstellungen](#)

Erstellen mehrerer Aufgaben

In einigen Migrationsszenarien müssen Sie möglicherweise mehrere Migrationsschritte erstellen. Aufgaben funktionieren unabhängig voneinander und können gleichzeitig ausgeführt werden. Jede Aufgabe hat ihren eigenen Prozess für das anfängliche Laden, CDC und das Lesen von Protokollen. Tabellen, die über DML (Datenmanipulationssprache) miteinander in Beziehung stehen, müssen Teil der gleichen Aufgabe sein.

Einige Gründe für das Erstellen mehrerer Aufgaben für eine Migration sind Folgende:

- Die Zieltabellen für die Aufgaben liegen auf verschiedenen Datenbanken, z. B., wenn Sie ein System erweitern oder in mehrere Systeme unterteilen.
- Sie möchten die Migration einer großen Tabelle mithilfe von Filtern in mehrere Aufgaben unterteilen.

Note

Da jede Aufgabe ihren eigenen Prozess für das Erfassen von Änderungen und das Lesen von Protokollen hat, werden Änderungen nicht über Aufgaben hinweg koordiniert. Vergewissern Sie sich daher bei Verwendung mehrerer Aufgaben für die Durchführung einer Migration, dass jede einzelne Quelltransaktion vollständig innerhalb einer einzigen Aufgabe enthalten ist. Sie können mehrere Aufgaben verwenden, um eine Migration durchzuführen, wenn keine einzelne Transaktion auf verschiedene Aufgaben aufgeteilt ist.

Erstellen von Aufgaben für die laufende Replikation mit AWS DMS

Sie können eine AWS DMS-Aufgabe erstellen, die fortlaufenden Änderungen aus dem Quelldatenspeicher erfasst. Diese Erfassung kann erfolgen, während Sie Ihre Daten migrieren. Sie können auch eine Aufgabe erstellen, die laufende Änderungen erfasst, nachdem Sie Ihre erste (Full-Load) Migration auf einen unterstützten Zieldatenspeicher abgeschlossen haben. Dieser Prozess wird als fortlaufende Replikation oder Change Data Capture (CDC) bezeichnet. AWS DMS verwendet diesen Prozess beim Replizieren von laufenden Änderungen aus einem Quelldatenspeicher. Bei

diesem Prozess werden Änderungen an den Datenbankprotokollen mithilfe der Datenbank-Engine der nativen API erfasst.

Note

Sie können Ansichten nur mithilfe von Full-Load-Aufgaben migrieren. Wenn es sich bei Ihrer Aufgabe entweder um eine reine CDC-Aufgabe oder eine Full-Load-Aufgabe handelt, die CDC nach ihrem Abschluss startet, enthält die Migration nur Tabellen aus der Quelle. Mit einer Full-Load-Aufgabe können Sie Ansichten oder eine Kombination aus Tabellen und Ansichten migrieren. Weitere Informationen finden Sie unter [Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON](#).

Jede Quell-Engine verfügt über spezifische Konfigurationsanforderungen für die Bereitstellung dieses Änderungsstroms für ein bestimmtes Benutzerkonto. Die meisten Engines erfordern eine zusätzliche Konfiguration, damit der Erfassungsprozess die Änderungsdaten sinnvoll und ohne Datenverlust verwerten kann. Zum Beispiel erfordert Oracle eine zusätzliche Protokollierung und MySQL eine binäre Protokollierung auf Zeilenebene.

Um fortlaufende Änderungen aus der Quelldatenbank zu lesen, verwendet AWS DMS zum Lesen von Änderungen aus den Transaktionsprotokollen der Quell-Engine jeweils für eine Engine spezifische API-Aktionen. Nachfolgend finden Sie einige Beispiele dafür, wie AWS DMS dabei vorgeht:

- Für Oracle nutzt AWS DMS entweder die Oracle LogMiner-API oder die binäre Leser-API (bfile-API), um fortlaufende Änderungen zu lesen. AWS DMS liest fortlaufende Änderungen aus den Online- oder Archiv-Redo-Protokollen basierend auf der Systemänderungsnummer (SCN).
- Für Microsoft SQL Server verwendet AWS DMS MS Replication oder MS-CDC, um Informationen in das SQL Server-Transaktionsprotokoll zu schreiben. Anschließend wird die Funktion `fn_dblog()` oder `fn_dump_dblog()` in SQL Server zum Lesen der Änderungen im Transaktionsprotokoll basierend auf der Log-Sequenznummer (LSN) verwendet.
- Für MySQL liest AWS DMS Änderungen aus den zeilenbasierten Binärprotokollen und migriert diese Änderungen in die Zieldatenbank.
- Für PostgreSQL richtet AWS DMS logische Replikations-Slots ein und verwendet das `test_decoding`-Plugin, um Änderungen aus der Quelldatenbank zu lesen und in die Zieldatenbank zu migrieren.
- Für Amazon RDS als Quelle sollten Sie sicherstellen, dass zur Einrichtung von CDC Sicherungen aktiviert sind. Wir empfehlen außerdem, sicherzustellen, dass die Quelldatenbank so konfiguriert

ist, dass Änderungsprotokolle ausreichend lange beibehalten werden – in der Regel sind 24 Stunden genug. Spezifische Einstellungen für die einzelnen Endpunkte finden Sie in den folgenden Themen:

- Amazon RDS für Oracle: [Konfiguration einer AWS-verwalteten Oracle-Quelle für AWS DMS](#).
- Amazon RDS für MySQL und Aurora MySQL: [Verwendung einer AWS-verwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#).
- Amazon RDS für SQL Server: [Einrichten der laufenden Replikation auf einer SQL-Server-DB-Instance in der Cloud](#).
- Amazon RDS für PostgreSQL und Aurora PostgreSQL: PostgreSQL behält automatisch das erforderliche WAL bei.

Es gibt zwei Arten von fortlaufenden Replikationsaufgaben:

- Vollständiger Ladevorgang plus CDC – Die Aufgabe migriert vorhandene Daten und aktualisiert dann die Zieldatenbank basierend auf den Änderungen an der Quelldatenbank.
- Nur CDC – Die Aufgabe migriert laufende Änderungen, nachdem Sie Daten in Ihrer Zieldatenbank vorliegen haben.

Durchführen der Replikation von einem CDC-Startpunkt aus

Sie können eine fortlaufende AWS DMS-Replikationsaufgabe (reine Change Data Capture-Aufgabe) von mehreren Punkten aus starten. Diese umfassen u. a. folgende:

- Ab einer benutzerdefinierten CDC-Startzeit – Sie können AWS DMS über die AWS Management Console oder die AWS CLI einen Zeitstempel für den Zeitpunkt angeben, an dem die Replikation beginnen soll. AWS DMS startet dann eine fortlaufende Replikationsaufgabe ab dieser benutzerdefinierten CDC-Startzeit. AWS DMS wandelt den angegebenen Zeitstempel (in UTC) in einen nativen Startpunkt um, wie z. B. eine LSN für SQL Server oder eine SCN für Oracle. AWS DMS verwendet Engine-spezifische Methoden, um zu ermitteln, wo genau die Migration aufgabe basierend auf dem Änderungs-Stream der Quell-Engine gestartet werden soll.

Note

Nur wenn für das Verbindungsattribut `StartFromContext` der erforderliche Zeitstempel festgelegt wird, bietet Db2 als Quelle eine benutzerdefinierte CDC-Startzeit.

PostgreSQL als Quelle unterstützt die benutzerdefinierte CDC-Startzeit nicht. Der Grund hierfür ist, dass die PostgreSQL-Datenbank-Engine keine Möglichkeit hat, einen Zeitstempel zu einem LSN oder SCN zuzuordnen, wie es Oracle und SQL Server tun.

- Ab einem nativen CDC-Startpunkt – Sie können auch ab einem nativen Punkt im Transaktionsprotokoll der Quell-Engine starten. In manchen Fällen kann dieser Ansatz besser geeignet sein, da ein Zeitstempel mehrere native Punkte im Transaktionsprotokoll angeben kann. AWS DMS unterstützt diese Funktion für die folgenden Quellendpunkte:
 - SQL Server
 - PostgreSQL
 - Oracle
 - MySQL
 - MariaDB

Wenn die Aufgabe erstellt wird, markiert AWS DMS den CDC-Startpunkt und dieser kann nicht geändert werden. Erstellen Sie eine neue Aufgabe, wenn Sie einen anderen CDC-Startpunkt verwenden möchten.

Bestimmen eines nativen CDC-Startpunkts

Ein nativer CDC-Startpunkt ist ein Punkt im Datenbank-Engine-Protokoll, der einen Zeitpunkt festlegt, ab dem Sie mit der CDC beginnen können. Nehmen wir beispielsweise an, dass bereits ein Massendaten-Dump auf das Ziel angewendet wurde. Sie können den nativen Startpunkt für die fortlaufende reine Replikationsaufgabe suchen. Wählen Sie den Startpunkt für die reine Replikationsaufgabe sorgfältig aus, um Dateninkonsistenzen zu vermeiden. DMS erfasst Transaktionen, die nach dem ausgewählten CDC-Startpunkt gestartet wurden.

Nachstehend finden Sie Beispiele dafür, wie Sie den nativen CDC-Startpunkt über unterstützte Quell-Engines finden:

SQL Server

Bei SQL Server besteht eine Log-Sequenznummer (LSN) aus drei Teilen:

- Virtual Log File (VLF)-Sequenznummer
- Anfangsversatz eines Protokollblocks
- Slot-Nummer

Die LSN lautet z. B.: 00000014:00000061:0001

Mit der Funktion `fn_dblog()` oder `fn_dump_dblog()` in SQL Server können Sie den Startpunkt einer SQL Server-Migrationsaufgabe basierend auf den Sicherungseinstellungen Ihres Transaktionsprotokolls anfordern.

Um den nativen CDC-Startpunkt mit SQL Server zu verwenden, erstellen Sie eine Veröffentlichung in einer beliebigen Tabelle, die an der fortlaufenden Replikation teilnimmt. AWS DMS erstellt die Veröffentlichung automatisch, wenn Sie CDC ohne Verwendung eines nativen CDC-Startpunkts verwenden.

PostgreSQL

Sie können einen CDC-Wiederherstellungsprüfpunkt für Ihre PostgreSQL-Quelldatenbank verwenden. Dieser Prüfpunktwert wird an verschiedenen Punkten generiert, während eine laufende Replikationsaufgabe für Ihre Quelldatenbank (die übergeordnete Aufgabe) ausgeführt wird. Weitere Informationen zu Prüfpunkten im Allgemeinen finden Sie unter [Verwendung eines Kontrollpunkts als CDC-Startpunkt](#).

Zur Ermittlung des Prüfpunkts, der als Ihr nativer Startpunkt verwendet werden soll, verwenden Sie Ihre Datenbankansicht `pg_replication_slots` oder die Übersichtsdetails Ihrer übergeordneten Aufgabe aus der AWS Management Console.

So finden Sie die Übersichtsdetails für Ihre übergeordnete Aufgabe auf der Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als IAM-Benutzer angemeldet sind, müssen Sie über die entsprechenden Berechtigungen für den Zugriff auf AWS DMS verfügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich die Option Database migration tasks (Datenbankmigrationsaufgaben) aus.
3. Wählen Sie Ihre übergeordnete Aufgabe aus der Liste auf der Seite Database migration tasks (Datenbankmigrationsaufgaben) aus. Dadurch wird die Seite der übergeordneten Aufgaben geöffnet, auf der die Übersichtsdetails angezeigt werden.
4. Suchen Sie unter Change Data Capture (CDC), Change Data Capture (CDC) start position (CDC-Startposition) und Change Data Capture (CDC) Recovery Checkpoint (CDC-Wiederherstellungsprüfpunkt) nach dem Prüfpunktwert.

Der Wert ähnelt dem folgenden:

```
checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0
```

Hier ist es die Komponente `4AF/B00000D0`, die Sie benötigen, um diesen nativen CDC-Startpunkt anzugeben. Setzen Sie beim Erstellen der CDC-Aufgabe den DMS-API-Parameter `CdcStartPosition` auf diesen Wert, um die Replikation an diesem Startpunkt für Ihre PostgreSQL-Quelle zu beginnen. Weitere Informationen zum Verwenden der AWS CLI zum Erstellen dieser CDC-Aufgabe finden Sie unter [Aktivieren von CDC mit einer von AWS verwalteten PostgreSQL-DB-Instance mit AWS DMS](#).

Oracle

Eine Systemänderungsnummer (SCN) ist ein logischer, interner Zeitstempel von Oracle-Datenbanken. SCNs dienen zur Anordnung von Ereignissen, die innerhalb der Datenbank auftreten. Dies ist erforderlich, um die ACID-Eigenschaften einer Transaktion zu erfüllen. Oracle-Datenbanken verwenden SCNs, um den Speicherort zu markieren, an dem alle Änderungen auf die Festplatte geschrieben wurden, sodass eine Wiederherstellungsaktion keine bereits geschriebenen Änderungen übernimmt. Oracle verwendet SCNs zudem zur Markierung des Punkts, an dem für einen Satz von Daten kein Redo vorhanden ist, sodass die Wiederherstellung enden kann.

Führen Sie den folgenden Befehl aus, um die aktuelle SCN in einer Oracle-Datenbank zu erhalten.

```
SELECT CURRENT_SCN FROM V$DATABASE
```

Wenn Sie die SCN oder den Zeitstempel verwenden, um eine CDC-Aufgabe zu starten, verpassen Sie die Ergebnisse aller offenen Transaktionen und können diese Ergebnisse nicht migrieren. Offene Transaktionen sind Transaktionen, die vor der Startposition der Aufgabe gestartet und nach der Startposition der Aufgabe festgeschrieben wurden. Sie können die SCN und den Zeitstempel identifizieren, um eine CDC-Aufgabe an einem Punkt zu starten, der alle offenen Transaktionen umfasst. Weitere Informationen finden Sie unter [Transaktionen](#) in der Oracle-Dokumentation. Ab Version 3.5.1 unterstützt AWS DMS offene Transaktionen für eine reine CDC-Aufgabe mithilfe der Endpunkteinstellung `openTransactionWindow`, wenn Sie die SCN oder den Zeitstempel zum Starten der Aufgabe verwenden.

Bei Verwendung der Einstellung `openTransactionWindow` müssen Sie das Zeitfenster in Minuten für die Bearbeitung offener Transaktionen angeben. AWS DMS verschiebt die Erfassungsposition und findet die neue Position, an der die Datenerfassung gestartet werden soll. AWS DMS verwendet die neue Startposition zum Scannen aller offenen Transaktionen aus den erforderlichen Oracle-Redo- oder archivierten Redo-Protokollen.

MySQL

Vor der Veröffentlichung von MySQL Version 5.6.3 bestand die Log-Sequenznummer (LSN) für MySQL aus einer 4-Byte-Ganzzahl ohne Vorzeichen. In MySQL Version 5.6.3 wurde die LSN mit der Erhöhung des Redo-Log-Dateigrößenlimits von 4 GB auf 512 GB zu einer 8-Byte-Ganzzahl ohne Vorzeichen. Die Erhöhung berücksichtigt, dass weitere Byte zum Speichern zusätzlicher Größeninformationen erforderlich waren. Anwendungen, die auf MySQL 5.6.3 oder höher basieren und LSN-Werte nutzen, sollten zum Speichern und Vergleichen von LSN-Werten 64-Bit- anstelle von 32-Bit-Variablen verwenden. Weitere Informationen zu MySQL LSNs finden Sie in der [MySQL-Dokumentation](#).

Führen Sie den folgenden Befehl aus, um die aktuelle LSN in einer MySQL-Datenbank zu erhalten.

```
mysql> show master status;
```

Die Abfrage gibt einen Binärprotokoll-Dateinamen, die Position und einige andere Werte zurück. Der native CDC-Startpunkt ist eine Kombination aus dem Namen der Binärprotokolldatei und der Position, z. B. `mysql-bin-changelog.000024:373`. In diesem Beispiel ist `mysql-bin-changelog.000024` der Binärprotokoll-Dateiname und 373 die Position, an der AWS DMS die Erfassung der Änderungen beginnen muss.

Verwendung eines Kontrollpunkts als CDC-Startpunkt

Eine fortlaufende Replikationsaufgabe migriert Änderungen und AWS DMS legt von Zeit zu Zeit AWS DMS-spezifische Prüfpunktangaben im Zwischenspeicher ab. Der Prüfpunkt, den AWS DMS erstellt, enthält Informationen, sodass die Replikations-Engine den Wiederherstellungspunkt für den Änderungsstream kennt. Sie können den Prüfpunkt verwenden, um in der Zeitleiste der Änderungen zurückzugehen und eine fehlgeschlagene Migrationsaufgabe wiederherzustellen. Mit einem Prüfpunkt können Sie auch eine weitere laufende Replikationsaufgabe für ein anderes Ziel zu einem bestimmten Zeitpunkt starten.

Zum Abrufen der Prüfpunktinformationen gibt es drei Möglichkeiten:

- Führen Sie die API-Operation `DescribeReplicationTasks` aus und sehen Sie sich die Ergebnisse an. Sie können die Informationen nach Aufgabe filtern und nach dem Prüfpunkt suchen. Sie können den letzten Kontrollpunkt abrufen, wenn sich die Aufgabe im angehaltenen oder fehlgeschlagenen Zustand befindet. Diese Informationen gehen verloren, wenn die Aufgabe gelöscht wird.
- Zeigen Sie die Metadatentabelle mit dem Namen `awsdms_txn_state` auf der Ziel-Instance an. Sie können die Tabelle abfragen, um Prüfpunktinformationen abzurufen. Zum Erstellen der Metadatentabelle legen Sie den `TaskRecoveryTableEnabled`-Parameter auf `Yes` fest, wenn Sie eine Aufgabe erstellen. Mit dieser Einstellung schreibt AWS DMS kontinuierlich Kontrollpunktangaben in die Ziel-Metadaten-Tabelle. Diese Informationen gehen verloren, wenn eine Aufgabe gelöscht wird.

Das folgende Beispiel zeigt einen Prüfpunkt in der Metadaten-Tabelle:

```
checkpoint:V1#34#00000132/0F000E48#0#0#*#0#121
```

- Wählen Sie im Navigationsbereich Datenbankmigrationsaufgaben und anschließend Ihre übergeordnete Aufgabe aus der Liste aus, die auf der Seite „Datenbankmigrationsaufgaben“ angezeigt wird. Die Seite der übergeordneten Aufgabe wird mit Übersichtsdetails geöffnet. Suchen Sie unter `Change Data Capture (CDC)`, `Change Data Capture (CDC) start position (CDC-Startposition)` und `Change Data Capture (CDC) Recovery Checkpoint (CDC-Wiederherstellungsprüfpunkt)` nach dem Prüfpunktwert. Der Prüfpunktwert ähnelt dem folgenden:

```
checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0
```

Anhalten einer Aufgabe an einem Commit- oder Server-Zeitpunkt

Mit der Einführung nativer CDC-Startpunkte hat AWS DMS auch die Möglichkeit, eine Aufgabe zu den folgenden Zeitpunkten anzuhalten:

- Commit-Zeitpunkt in der Quelle
- Server-Zeitpunkt auf der Replikations-Instance

Sie können eine Aufgabe ändern und eine Uhrzeit zum Anhalten in koordinierter Weltzeit (UTC) wie erforderlich festlegen. Die Aufgabe wird basierend auf dem Commit- oder Server-Zeitpunkt, den Sie festgelegt haben, automatisch gestoppt. Wenn Sie zum Zeitpunkt der Aufgabenerstellung wissen, wann die Migrationsaufgabe enden soll, können Sie im Rahmen der Erstellung auch den betreffenden Zeitpunkt zum Stoppen festlegen.

Note

Beim ersten Start einer neuen AWS DMS-Serverless-Replikation kann es bis zu 40 Minuten dauern, bis alle Ressourcen initialisiert sind. Beachten Sie, dass die Option `server_time` erst verfügbar ist, wenn die Ressourceninitialisierung abgeschlossen ist.

Durchführen der bidirektionalen Replikation

Sie können AWS DMS-Aufgaben verwenden, um eine bidirektionale Replikation zwischen zwei Systemen durchzuführen. Bei der bidirektionalen Replikation replizieren Sie Daten aus derselben Tabelle (oder einem Satz von Tabellen) zwischen zwei Systemen in beide Richtungen.

Beispielsweise können Sie eine Tabelle MITARBEITER von Datenbank A nach Datenbank B kopieren und Änderungen an der Tabelle von Datenbank A nach Datenbank B replizieren. Sie können auch Änderungen an der Tabelle MITARBEITER von Datenbank B zurück nach A replizieren. Sie führen also eine bidirektionale Replikation durch.

Note

Die bidirektionale AWS DMS-Replikation ist nicht als vollständige Multi-Master-Lösung mit einem Primärknoten, Konfliktlösung usw. gedacht.

Verwenden Sie die bidirektionale Replikation für Situationen, in denen Daten auf verschiedenen Knoten operativ getrennt sind. Anders ausgedrückt: Angenommen, Sie lassen ein Datenelement von einer Anwendung ändern, die auf Knoten A ausgeführt wird, und Knoten A führt eine bidirektionale Replikation mit Knoten B durch. Dieses Datenelement auf Knoten A wird niemals von einer Anwendung geändert, die auf Knoten B ausgeführt wird.

AWS DMS unterstützt die bidirektionale Replikation auf diesen Datenbank-Engines:

- Oracle
- SQL Server
- MySQL
- PostgreSQL
- Amazon Aurora MySQL-Compatible Edition

- Aurora PostgreSQL-Compatible Edition

Erstellen von bidirektionalen Replikationsaufgaben

Um eine bidirektionale AWS DMS-Replikation zu ermöglichen, konfigurieren Sie Quell- und Ziel-Endpunkte für beide Datenbanken (A und B). Konfigurieren Sie z. B. einen Quell-Endpunkt für Datenbank A, einen Quell-Endpunkt für Datenbank B, einen Ziel-Endpunkt für Datenbank A und einen Ziel-Endpunkt für Datenbank B.

Erstellen Sie dann zwei Aufgaben: eine Aufgabe für Quelle A zum Verschieben von Daten nach Ziel B und eine weitere Aufgabe für Quelle B zum Verschieben von Daten nach Ziel A. Stellen Sie außerdem sicher, dass jede Aufgabe mit einer Loopback-Verhinderung konfiguriert ist. Auf diese Weise wird verhindert, dass identische Änderungen an den Zielen beider Aufgaben vorgenommen werden und somit die Daten für mindestens eine der beiden Aufgaben fehlerhaft werden. Weitere Informationen finden Sie unter [Verhindern eines Loopbacks](#).

Die einfachste Methode ist, mit identischen Datensätzen sowohl in Datenbank A als auch in Datenbank B zu beginnen. Erstellen Sie dann zwei Nur-CDC-Aufgaben, eine Aufgabe, um Daten von A nach B zu replizieren, und eine weitere Aufgabe, um Daten von B nach A zu replizieren.

Um AWS DMS zur Instanziierung eines neuen Datensatzes (Datenbank) auf Knoten B von Knoten A zu verwenden, gehen Sie wie folgt vor:

1. Verwenden Sie eine Aufgabe zum vollständigen Laden und eine CDC-Aufgabe, um Daten von Datenbank A nach B zu verschieben. Stellen Sie sicher, dass während dieser Zeit keine Anwendungen Daten in Datenbank B ändern.
2. Wenn das vollständige Laden abgeschlossen ist und bevor Anwendungen Daten in Datenbank B ändern dürfen, notieren Sie die Uhrzeit oder die CDC-Startposition von Datenbank B. Anweisungen finden Sie unter [Durchführen der Replikation von einem CDC-Startpunkt aus](#).
3. Erstellen Sie eine Nur-CDC-Aufgabe, bei der die Daten von Datenbank B zurück nach A unter Verwendung dieser Startzeit oder CDC-Startposition verschoben werden.

Note

Nur eine Aufgabe in einem bidirektionalen Paar kann eine Aufgabe für vollständiges Laden und CDC sein.

Verhindern eines Loopbacks

Um das Verhindern eines Loopbacks anzuzeigen, nehmen wir an, dass AWS DMS in einer Aufgabe T1 Änderungsprotokolle aus der Quelldatenbank A liest und die Änderungen auf die Zieldatenbank B anwendet.

Als Nächstes liest eine zweite Aufgabe, T2, Änderungsprotokolle aus der Quelldatenbank B und wendet diese wieder auf die Zieldatenbank A an. Bevor T2 dies tut, muss DMS sicherstellen, dass dieselben Änderungen, die von der Quelldatenbank A in der Zieldatenbank B vorgenommen wurden, nicht auch in der Quelldatenbank A vorgenommen werden. Anders ausgedrückt: DMS muss sicherstellen, dass diese Änderungen nicht auf die Zieldatenbank A zurückreflektiert (zurückgeschleift) werden. Andernfalls können die Daten in Datenbank A beschädigt werden.

Um ein Loopback (eine Schleifenschaltung) von Änderungen zu verhindern, fügen Sie die folgenden Aufgabeneinstellungen zu jeder bidirektionalen Replikationsaufgabe hinzu. Dadurch wird sichergestellt, dass eine Loopback-Datenbeschädigung nicht in beide Richtungen auftritt.

```
{
  . . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": Boolean,
    "SourceSchema": String,
    "TargetSchema": String
  },

  . . .
}
```

Über die Aufgabeneinstellungen von `LoopbackPreventionSettings` wird festgestellt, ob eine Transaktion neu ist oder ein Echo von der Aufgabe für die entgegengesetzte Replikation. Wenn das AWS DMS eine Transaktion auf eine Zieldatenbank anwendet, aktualisiert es eine DMS-Tabelle (`awsdms_loopback_prevention`) mit einem Hinweis auf die Änderung. Bevor jede Transaktion auf ein Ziel angewendet wird, ignoriert das DMS jede Transaktion, die einen Verweis auf diese `awsdms_loopback_prevention`-Tabelle enthält. Daher wendet es die Änderung nicht an.

Nehmen Sie diese Aufgabeneinstellungen bei jeder Replikationsaufgabe in ein bidirektionales Paar auf. Diese Einstellungen aktivieren die Loopback-Verhinderung. Sie geben auch das Schema für jede Quell- und Zieldatenbank in der Aufgabe an, die die `awsdms_loopback_prevention`-Tabelle für jeden Endpunkt enthält.

Damit jede Aufgabe ein solches Echo identifizieren und ausschalten kann, setzen Sie `EnableLoopbackPrevention` auf `true`. Um an der Quelle ein Schema anzugeben, das `awsdms_loopback_prevention` enthält, setzen Sie `SourceSchema` auf den Namen für dieses Schema in der Quelldatenbank. Um am Ziel ein Schema anzugeben, das dieselbe Tabelle enthält, setzen Sie `TargetSchema` auf den Namen für dieses Schema in der Zieldatenbank.

Im folgenden Beispiel werden die `SourceSchema`- und `TargetSchema`-Einstellungen für eine Replikationsaufgabe T1 und ihre Aufgabe T2 für die entgegengesetzte Replikation mit entgegengesetzten Einstellungen angegeben.

Die Einstellungen für die Aufgabe T1 sind wie folgt.

```
{
  . . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "LOOP-DATA",
    "TargetSchema": "loop-data"
  },

  . . .
}
```

Die Einstellungen für die Aufgabe T2 für die entgegengesetzte Replikation sind wie folgt.

```
{
  . . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "loop-data",
    "TargetSchema": "LOOP-DATA"
  },

  . . .
}
```

Note

Wenn Sie die AWS CLI verwenden, verwenden Sie nur die Befehle `create-replication-task` oder `modify-replication-task`, um die `LoopbackPreventionSettings` in Ihren Aufgaben für die bidirektionale Replikation zu konfigurieren.

Einschränkungen der bidirektionalen Replikation

Die bidirektionale Replikation für AWS DMS weist folgende Einschränkungen auf:

- Die Loopback-Verhinderung verfolgt nur DML-Anweisungen (Data Manipulation Language). AWS DMS unterstützt die Verhinderung des DDL-Loopback (Data Definition Language) nicht. Konfigurieren Sie dazu eine der Aufgaben in einem bidirektionalen Paar, um DDL-Anweisungen herauszufiltern.
- Aufgaben, bei denen die Loopback-Verhinderung verwendet wird, unterstützen nicht die Übergabe von Änderungen in Stapeln. Um eine Aufgabe mit Loopback-Verhinderung zu konfigurieren, stellen Sie sicher, dass `BatchApplyEnabled` auf `false` gesetzt ist.
- Die bidirektionale DMS-Replikation umfasst keine Konflikterkennung oder -lösung. Um Dateninkonsistenzen zu erkennen, verwenden Sie die Datenvalidierung für beide Aufgaben.

Ändern einer Aufgabe

Sie können eine Aufgabe ändern, wenn Sie die Aufgabeneinstellungen, Tabellenzuweisung oder andere Einstellungen ändern müssen. Sie können auch Bewertungen vor der Migration aktivieren und ausführen, bevor Sie die geänderte Aufgabe ausführen. Eine Aufgabe in der Konsole können Sie ändern, indem Sie die Aufgabe auswählen und anschließend auf **Ändern** klicken. Sie können auch den CLI-Befehl oder die API-Operation [ModifyReplicationTask](#) verwenden.

Es gibt einige Einschränkungen für das Ändern einer Aufgabe. Diese umfassen u. a. folgende:

- Sie können den Quell- oder Zielendpunkt einer Aufgabe nicht ändern.
- Sie können den Migrationstyp einer Aufgabe nicht ändern.
- Aufgaben, die ausgeführt wurden, müssen den Status **Angehalten** oder **Fehlgeschlagen** aufweisen, damit sie geändert werden können.

Verschieben einer Aufgabe

Sie können eine Aufgabe auf eine andere Replikations-Instance verschieben, wenn eine der folgenden Situationen auf Ihren Anwendungsfall zutrifft.

- Sie verwenden derzeit eine Instance eines bestimmten Typs und möchten zu einem anderen Instance-Typ wechseln.
- Ihre aktuelle Instance ist durch viele Replikationsaufgaben überlastet und Sie möchten die Last auf mehrere Instances verteilen.
- Ihr Instance-Speicher ist voll und als Alternative zur Skalierung von Speicher oder Rechenleistung möchten Sie Aufgaben von dieser Instance auf eine leistungsstärkere Instance verschieben.
- Sie möchten ein neu veröffentlichtes Feature von AWS DMS verwenden, ohne eine neue Aufgabe zu erstellen und die Migration neu zu starten. Stattdessen ziehen Sie es vor, eine Replikations-Instance mit einer neuen AWS DMS-Version zu starten, die dieses Feature unterstützt, und die vorhandene Aufgabe auf diese Instance zu verschieben.

Eine Aufgabe in der Konsole können Sie verschieben, indem Sie die Aufgabe auswählen und anschließend auf Verschieben klicken. Sie können auch den CLI-Befehl oder die API-Operation `MoveReplicationTask` verwenden, um die Aufgabe zu verschieben. Sie können eine Aufgabe verschieben, deren Zielendpunkt eine beliebige Datenbank-Engine ist.

Stellen Sie sicher, dass die Zielreplikations-Instance über ausreichend Speicherplatz für die Aufgabe verfügt, die verschoben wird. Skalieren Sie andernfalls den Speicher, um Platz für Ihre Zielreplikations-Instance zu schaffen, bevor Sie die Aufgabe verschieben.

Stellen Sie außerdem sicher, dass Ihre Zielreplikations-Instance mit derselben oder einer höheren AWS DMS-Engine-Version wie die aktuelle Replikations-Instance erstellt wurde.

Note

- Sie können eine Aufgabe nicht in dieselbe Replikations-Instance verschieben, in der sie sich derzeit befindet.
- Sie können die Einstellungen einer Aufgabe während des Verschiebens nicht ändern.
- Eine ausgeführte Aufgabe muss den Status Angehalten, Fehlgeschlagen oder Verschieben fehlgeschlagen aufweisen, damit Sie sie verschieben können.

Es gibt zwei Aufgabenstatus im Zusammenhang mit dem Verschieben einer DMS-Aufgabe: Verschieben und Verschieben fehlgeschlagen. Weitere Informationen zu diesen Aufgabenstatus finden Sie unter [Aufgabenstatus](#).

Nach dem Verschieben einer Aufgabe können Sie Bewertungen vor der Migration aktivieren und ausführen, um nach Fehlern zu suchen, bevor Sie die verschobene Aufgabe ausführen.

Erneutes Laden von Tabellen während einer Aufgabe

Während eine Aufgabe ausgeführt wird, können Sie eine Zieldatenbanktabelle mit Daten aus der Quelle erneut laden. Möglicherweise möchten Sie eine Tabelle neu laden, wenn während der Aufgabe ein Fehler auftritt oder Daten sich aufgrund von Partitions-Operationen ändern (z. B. bei der Verwendung von Oracle). Sie können bis zu 10 Tabellen von einer Aufgabe erneut laden.

Durch das erneute Laden von Tabellen wird die Aufgabe nicht beendet.

Um eine Tabelle erneut zu laden, müssen die folgenden Bedingungen erfüllt sein:

- Die Aufgabe muss ausgeführt werden.
- Die Migrationmethode für die Aufgabe muss entweder "Full Load" oder "Full Load with CDC" sein.
- Doppelte Tabellen sind nicht zulässig.
- AWS DMS behält die vorherige Tabellendefinition bei und erstellt sie während des erneuten Ladens nicht neu. Alle DDL-Anweisungen wie ALTER TABLE ADD COLUMN oder DROP COLUMN, die auf die Tabelle angewendet werden, bevor diese neu geladen wird, können bewirken, dass das Neuladen fehlschlägt.

Note

DMS wendet die `TargetTablePrepMode`-Einstellung an, bevor die Tabelle neu geladen wird. Wenn Sie `TargetTablePrepMode` auf `DO_NOTHING` einstellen, müssen Sie die Tabelle zuerst manuell kürzen.

AWS Management Console

So laden Sie eine Tabelle mithilfe der AWS DMS-Konsole erneut

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als IAM-Benutzer angemeldet sind, müssen Sie über die entsprechenden Zugriffsberechtigungen für AWS DMS verfügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich Tasks (Aufgaben) aus.
3. Wählen Sie die ausgeführte Aufgabe mit der Tabelle, die Sie erneut laden möchten.
4. Wählen Sie die Registerkarte Table Statistics (Tabellenstatistiken).

The screenshot shows the AWS DMS console interface for a task named 'move-data'. At the top, there are buttons for 'Create task', 'Assess', 'Modify', 'Start/Resume', 'Stop', and 'Delete'. Below these is a search filter. The main table lists the task with columns: ID, Status, Source, Target, and Type. The task 'move-data' is in a 'Running' state, with source 'from-mysql-sou' and target 'to-pgsqj-target', and type 'Full Load'.

Below the task list, the 'move-data' task is selected, and the 'Table statistics' tab is active. This tab contains buttons for 'Revalidate' and 'Reload table data', with the latter being circled in red. Below the buttons is another search filter and a table of table statistics.

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. Wählen Sie die Tabelle aus, die Sie neu laden möchten. Wenn die Aufgabe nicht mehr ausgeführt wird, können Sie die Tabelle nicht neu laden.
6. Wählen Sie Reload table data (Tabellendaten neu laden).

Wenn AWS DMS das erneute Laden einer Tabelle vorbereitet, ändert die Konsole den Tabellenstatus in Table is being reloaded (Tabelle wird erneut geladen).

Verwenden der Tabellenzuweisung zum Angeben von Aufgabeneinstellungen

Die Zuweisung von Tabellen verwendet mehrere Arten von Regeln für die Angabe der Datenquelle, des Quellschemas, der Daten und aller Transformationen, die während der Aufgabe auftreten sollten.

Sie können mithilfe von Tabellenzuweisung einzelne zu migrierende Tabellen in einer Datenbank und das für die Migration zu verwendende Schema bestimmen.

Wenn Sie mit der Tabellenzuweisung arbeiten, können Sie mithilfe von Filtern Daten angeben, die aus Tabellenspalten repliziert werden sollen. Außerdem können Sie Transformationen verwenden, um ausgewählte Schemata, Tabellen oder Ansichten zu ändern, bevor sie in die Zieldatenbank geschrieben werden.

Themen

- [Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole](#)
- [Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON](#)
- [Auswahlregeln und Aktionen](#)
- [Platzhalter bei der Tabellenzuordnung](#)
- [Transformationsregeln und Aktionen](#)
- [Verwenden von Transformationsregelausdrücken zum Definieren von Spalteninhalten](#)
- [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#)


Note

Wenn Sie mit der Tabellenzuweisung für einen MongoDB-Quellendpunkt arbeiten, können Sie Filter verwenden, um Daten anzugeben, die repliziert werden sollen, und einen Datenbanknamen anstelle von `schema_name` angeben. Sie können den Standard-"`%`" verwenden.

Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole

Sie können den verwenden AWS Management Console , um Tabellenzuordnungen durchzuführen, einschließlich der Angabe von Tabellenauswahl und Transformationen. In der Konsole können Sie im Bereich Where (Wo) Schema, Tabelle und Aktion (Einschließen oder Ausschließen) angeben. Im Bereich Filter können Sie den Spaltennamen in einer Tabelle sowie die Bedingungen angeben, die für eine Replikationsaufgabe gelten sollen. Zusammen erstellen diese beiden Aktionen eine Auswahlregel.

Sie können Transformationen in einer Tabellenzuweisung einschließen, nachdem Sie mindestens eine Auswahlregel angegeben haben. Sie können Transformationen dazu verwenden, ein Schema oder eine Tabelle umzubenennen, einem Schema oder einer Tabelle ein Präfix oder Suffix anzufügen oder eine Tabellenspalte zu entfernen.

 Note

AWS DMS unterstützt nicht mehr als eine Transformationsregel pro Schema-, Tabellen- oder Spaltenebene.

Das folgende Verfahren zeigt, wie Auswahlregeln basierend auf einer Tabelle namens **Customers** in einem Schema mit der Bezeichnung **EntertainmentAgencySample** eingerichtet werden.

So geben Sie eine Tabellenauswahl, Filterkriterien und Transformationen mithilfe der Konsole an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als IAM-Benutzer angemeldet sind, müssen Sie über die entsprechenden Zugriffsberechtigungen für AWS DMS verfügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie auf der Seite Dashboard Datenbankmigrationsaufgaben aus.
3. Wählen Sie Aufgabe erstellen aus.
4. Geben Sie im Abschnitt Aufgabenkonfiguration die Aufgabeninformationen ein, einschließlich Aufgaben-ID, Replikations-Instance, Quelldatenbank-Endpunkt, Zieldatenbank-Endpunkt und Migrationstyp.

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. Wählen Sie im Abschnitt **Tabellenzuordnung** den Schemanamen und den Tabellennamen aus. Sie können bei der Angabe des Schema- oder Tabellennamens „%“ als Platzhalterzeichen verwenden. Informationen zu anderen Platzhaltern, die Sie verwenden können, finden Sie unter [the section called “Platzhalter bei der Tabellenzuordnung”](#). Geben Sie die auszuführende Aktion an, um vom Filter definierte Daten ein- oder auszuschließen.

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. Geben Sie Filterinformationen mithilfe der Links **Add column filter** (Spaltenfilter hinzufügen) und **Add condition** (Bedingung hinzufügen) an.
 - a. Wählen Sie **Add column filter** (Spaltenfilter hinzufügen) aus, um eine Spalte und Bedingungen anzugeben.
 - b. Wählen Sie **Add condition** (Bedingung hinzufügen) aus, um zusätzliche Bedingungen hinzuzufügen.

Das folgende Beispiel zeigt einen Filter für die Tabelle **Customers**, die **AgencyIDs** zwischen **01** und **85** enthält.

Source filters [Info](#) **Add column filter**

▼ Column filter 1

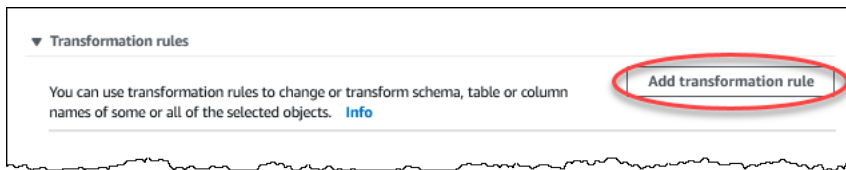
Column name
AgencyId

Condition 1
Equal to or between two values
01
85

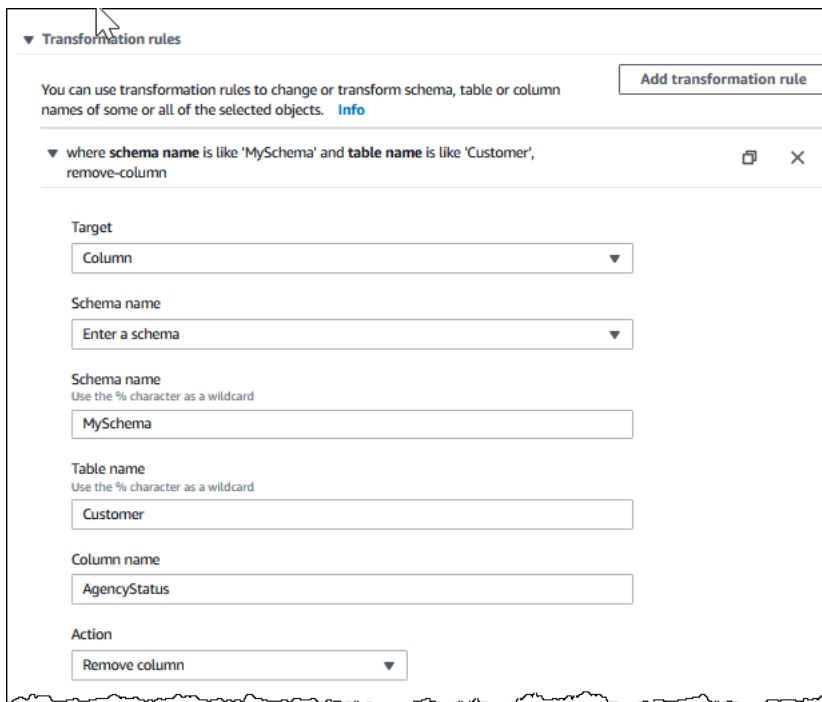
Add condition

7. Wenn Sie die gewünschte Auswahl getroffen haben, wählen Sie **Auswahlregel hinzufügen** aus.

- Nachdem Sie mindestens eine Auswahlregel erstellt haben, können Sie eine Transformation zu der Aufgabe hinzufügen. Wählen Sie Add transformation rule (Transformationsregel hinzufügen) aus.



- Wählen Sie das Ziel aus, das Sie transformieren möchten, und geben Sie die angefragten zusätzlichen Informationen ein. Das folgende Beispiel zeigt eine Transformation, bei der die Spalte **AgencyStatus** aus der Tabelle **Customer** gelöscht wird.



- Wählen Sie Add transformation rule (Transformationsregel hinzufügen) aus.
- Wählen Sie Create task aus.

Note

AWS DMS unterstützt nicht mehr als eine Transformationsregel pro Schema- oder Tabellenebene.

Festlegen der Tabellenauswahl- und Transformationsregeln mit JSON

Zum Festlegen der Tabellenzuweisungen, die Sie während der Migration anwenden möchten, können Sie eine JSON-Datei erstellen. Wenn Sie eine Migrationsaufgabe mithilfe der Konsole erstellen, können Sie nach dieser JSON-Datei suchen oder das JSON direkt in das Tabellenzuweisungsfeld eingeben. Wenn Sie die CLI oder API verwenden, um Migrationen durchzuführen, können Sie diese Datei mit dem Parameter `TableMappings` der API-Operation `CreateReplicationTask` oder `ModifyReplicationTask` angeben.

AWS DMS kann nur JSON-Dateien mit Tabellenzuordnungen mit einer Größe von bis zu 2 MB verarbeiten. Wir empfehlen, dass Sie bei der Arbeit mit DMS-Aufgaben die Größe der JSON-Datei für die Zuordnungsregel unter dem Grenzwert von 2 MB halten. Dadurch werden unerwartete Fehler bei der Erstellung oder Änderung von Aufgaben vermieden. Wenn eine Zuordnungsregeldatei den Grenzwert von 2 MB überschreitet, empfehlen wir, die Tabellen auf mehrere Aufgaben aufzuteilen, um die Größe der Zuordnungsregeldatei so zu reduzieren, dass sie unter diesem Limit bleibt.

Sie können angeben, mit welchen Tabellen, Ansichten und Schemata Sie arbeiten möchten. Sie können auch Tabellen-, Ansichts- und Schematransformationen durchführen und Einstellungen dazu festlegen, wie AWS DMS einzelne Tabellen und Ansichten lädt. Sie erstellen mit den folgenden Regeltypen Tabellenzuweisungsregeln für diese Optionen:

- `selection`-Regeln – Identifizieren die Typen und Namen der Quelltabellen, Ansichten und Schemata, die geladen werden sollen. Weitere Informationen finden Sie unter [Auswahlregeln und Aktionen](#).
- `transformation`-Regeln – Geben bestimmte Änderungen oder Ergänzungen an bestimmten Quelltabellen und Schemata in der Quelle an, bevor sie auf dem Ziel geladen werden. Weitere Informationen finden Sie unter [Transformationsregeln und Aktionen](#).

Um den Inhalt neuer und vorhandener Spalten zu definieren, können Sie auch einen Ausdruck innerhalb einer Transformationsregel verwenden. Weitere Informationen finden Sie unter [Verwenden von Transformationsregelausdrücken zum Definieren von Spalteninhalten](#).

- `table-settings`-Regeln – Legen fest, wie DMS-Aufgaben die Daten für einzelne Tabellen laden. Weitere Informationen finden Sie unter [Regeln und Operationen für Tabellen- und Sammlungseinstellungen](#).

Note

Für Amazon-S3-Ziele können Sie auch S3-Objekte markieren, die mithilfe des `post-processing`-Regeltyps und der `add-tag`-Regelaktion ausgewählten Tabellen und Schemata zugeordnet sind. Weitere Informationen finden Sie unter [Markieren von Amazon-S3-Objekten](#).

Für die folgenden Ziele können Sie mithilfe des `object-mapping`-Regeltyps angeben, wie und wo ausgewählte Schemata und Tabellen auf das Ziel migriert werden:

- Amazon DynamoDB – Weitere Informationen finden Sie unter [Verwenden der Objektzuordnung zum Migrieren von Daten zu DynamoDB](#).
- Amazon Kinesis – Weitere Informationen finden Sie unter [Verwenden der Objektzuweisung zum Migrieren von Daten zu einem Kinesis-Datenstrom](#).
- Apache Kafka – Weitere Informationen finden Sie unter [Verwenden der Objektzuweisung zum Migrieren von Daten zu einem Kafka-Thema](#).


Auswahlregeln und Aktionen

Mithilfe der Tabellenzuweisung können Sie angeben, mit welchen Tabellen, Ansichten oder Schemata Sie arbeiten möchten, indem Sie Auswahlregeln und Aktionen verwenden. Für Tabellenzuweisungsregeln, die den Regeltyp "Auswahl" verwenden, können Sie die folgenden Werte anwenden.

Parameter	Mögliche Werte	Beschreibung
<code>rule-type</code>	<code>selection</code>	Eine Auswahlregel. Definieren Sie mindestens eine Auswahlregel, wenn Sie eine Tabellenzuweisung angeben.
<code>rule-id</code>	Ein numerischer Wert.	Ein eindeutiger numerischer Wert zur Identifizierung der Regel.
<code>rule-name</code>	Ein alphanumerischer Wert.	Ein eindeutiger Name zur Identifizierung der Regel.

Parameter	Mögliche Werte	Beschreibung
<code>rule-action</code>	<code>include, exclude, explicit</code>	Ein Wert, der das bzw. die Objekte, die von der Regel ausgewählt wurden, einschließt oder ausschließt. Wenn <code>explicit</code> angegeben ist, können Sie ein Objekt, das einer explizit angegebenen Tabelle oder einem Schema entspricht, auswählen und hinzufügen.

Parameter	Mögliche Werte	Beschreibung
<code>object-locator</code>	<p>Ein Objekt mit den folgenden Parametern:</p> <ul style="list-style-type: none"> • <code>schema-name</code> – Der Name des Schemas. • <code>table-name</code> – Der Name der Tabelle • (Optional) <code>table-type</code> – <code>table</code> <code>view</code> <code>all</code>, um anzugeben, ob sich <code>table-name</code> nur auf Tabellen oder Ansichten oder auf Tabellen und Ansichten bezieht. Der Standardwert ist <code>table</code>. <p>AWS DMS lädt Ansichten nur bei einer Aufgabe mit vollständigem Laden. Wenn Sie nur Volllast- und CDC-Aufgaben (Change Data Capture) haben, konfigurieren Sie mindestens eine <code>full-load-only</code> Aufgabe, um Ihre Ansichten zu laden.</p> <p>Nicht alle Zielendpunkte akzeptieren Ansichten als Replikationsquelle, auch nicht bei Volllast (z. B. Amazon OpenSearch Service). Prüfen Sie die Einschränkungen Ihres Zielendpunkts.</p>	<p>Der Name der einzelnen Schemen und Tabellen oder Ansichten, für die die Regel gilt. Sie können auch angeben, ob eine Regel nur Tabellen, nur Ansichten oder Tabellen und Ansichten enthält. Wenn die <code>rule-action</code> entweder <code>include</code> oder <code>exclude</code> ist, können Sie das Prozentzeichen "%" als Platzhalter für alle oder einen Teil der Werte des <code>schema-name</code> - und <code>table-name</code> -Parameters verwenden. Informationen zu anderen Platzhaltern, die Sie verwenden können, finden Sie unter the section called "Platzhalter bei der Tabellenzuordnung". Auf diese Weise können Sie diese Elemente aufeinander abstimmen:</p> <ul style="list-style-type: none"> • Eine einzige Tabelle, Ansicht oder Sammlung in einem einzigen Schema • Eine einzige Tabelle, Ansicht oder Sammlung in einigen oder allen Schemata • Einige oder alle Tabellen und Ansichten in einem einzigen Schema oder Sammlungen in einer einzigen Datenbank • Einige oder alle Tabellen und Ansichten in einigen oder allen Schemata oder Sammlungen in einigen oder allen Datenbanken

Parameter	Mögliche Werte	Beschreibung
		<p>Wenn die <code>rule-action</code> <code>explicit</code> ist, können Sie nur den genauen Namen einer einzelnen Tabelle oder Ansicht und eines Schemas angeben (ohne Platzhalter).</p> <p>Zu den unterstützten Quellen für Ansichten gehören:</p> <ul style="list-style-type: none">• Oracle• Microsoft SQL Server• PostgreSQL• IBM Db2 (LUW)• IBM Db2 z/OS• SAP Adaptive Server Enterprise (ASE)• MySQL• AURORA• AURORA Serverless• MariaDB <div data-bbox="971 1325 1511 1780" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>AWS DMS lädt niemals eine Quellansicht in eine Zielansicht. Eine Quellansicht wird in eine gleichwertige Tabelle auf dem Ziel mit demselben Namen wie die Ansicht auf der Quelle geladen.</p></div>

Parameter	Mögliche Werte	Beschreibung
		<p>Zu den unterstützten Quellen für Datenbanken, die Sammlungen enthalten, gehören:</p> <ul style="list-style-type: none"> • MongoDB • Amazon DocumentDB
<code>load-order</code>	Eine positive Ganzzahl. Der maximale Wert beträgt 2.147.483.647.	Die Priorität für das Laden von Tabellen und Ansichten. Tabellen und Ansichten mit höheren Werten werden zuerst geladen.
<code>filters</code>	Ein Array von -Objekten.	Ein oder mehrere Objekte zum Filtern der Quelle. Sie geben Objektparameter zum Filtern auf einer einzelnen Spalte in der Quelle an. Sie geben einige Objekte zum Filtern auf mehreren Spalten an. Weitere Informationen finden Sie unter Verwenden von Quellfiltern .

Example Migrieren aller Tabellen in einem Schema

Im folgenden Beispiel werden alle Tabellen aus einem Schema mit dem Namen Test in Ihrer Quelle zu Ihrem Zielpunkt migriert.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "Test",
        "table-name": "%"
      },
      "rule-action": "include"
    }
  ]
}
```

```

    }
  ]
}

```

Example Migrieren einiger Tabellen in einem Schema

Im folgenden Beispiel werden alle Tabellen außer denjenigen, die mit DMS beginnen, aus einem Schema mit dem Namen Test in Ihrer Quelle zu Ihrem Zielendpunkt migriert.

```

{
  "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 Migrieren einer bestimmten einzelnen Tabelle in einem einzelnen Schema

Im folgenden Beispiel wird die Customer-Tabelle aus dem NewCust-Schema in Ihrer Quelle zu Ihrem Zielendpunkt migriert.

```

{
  "rules": [
    {
      "rule-type": "selection",

```

```

        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {
            "schema-name": "NewCust",
            "table-name": "Customer"
        },
        "rule-action": "explicit"
    }
]
}

```

Note

Sie können explizit auf mehreren Tabellen und Schemata auswählen, indem Sie mehrere Auswahlregeln angeben.

Example Migrieren von Tabellen in einer festgelegten Reihenfolge

Beim folgenden Beispiel werden zwei Tabellen migriert. Die Tabelle `loadfirst` (mit Priorität 1) wird vor der Tabelle `loadsecond` initialisiert.

```

{
  "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` gilt für die Tabelleninitialisierung. Das Laden einer nachfolgenden Tabelle wartet nicht darauf, dass das Laden einer vorherigen Tabelle abgeschlossen ist, wenn der Wert `MaxFullLoadSubTasks` größer als 1 ist.

Example Migrieren einiger Ansichten in einem Schema

Im folgenden Beispiel werden einige Ansichten aus einem Schema mit dem Namen `Test` in Ihrer Quelle in gleichwertige Tabellen in Ihrem Ziel migriert.

```

{
  "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 Migrieren aller Tabellen und Ansichten in einem Schema

Im folgenden Beispiel werden alle Tabellen und Ansichten aus einem Schema mit dem Namen `report` in Ihrer Quelle in gleichwertige Tabellen in Ihrem Ziel migriert.

```

{

```

```

"rules": [
  {
    "rule-type": "selection",
    "rule-id": "3",
    "rule-name": "3",
    "object-locator": {
      "schema-name": "report",
      "table-name": "%",
      "table-type": "all"
    },
    "rule-action": "include"
  }
]
}

```

Platzhalter bei der Tabellenzuordnung

In diesem Abschnitt werden Platzhalter beschrieben, die Sie verwenden können, wenn Sie die Schema- und Tabellennamen für die Tabellenzuordnung angeben.

Platzhalter	Übereinstimmungen
%	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'.

Für Oracle-Quell- und Zielendpunkte können Sie das zusätzliche `escapeCharacter`-Verbindungsattribut verwenden, um ein Escape-Zeichen anzugeben. Ein Escape-Zeichen ermöglicht Ihnen, ein bestimmtes Platzhalterzeichen in Ausdrücken so zu verwenden, als ob es kein Platzhalterzeichen wäre. `escapeCharacter=#` Ermöglicht Ihnen beispielsweise, „#“ zu verwenden,

damit ein Platzhalterzeichen in einem Ausdruck wie in diesem Beispielcode als normales Zeichen fungiert.

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "542485267",
      "rule-name": "542485267",
      "object-locator": { "schema-name": "R00T", "table-name": "TEST#_T%" },
      "rule-action": "include",
      "filters": []
    }
  ]
}
```

Hier sorgt das Escape-Zeichen '#' dafür, dass sich das Platzhalterzeichen '_' wie ein normales Zeichen verhält. AWS DMS wählt Tabellen im genannten Schema ausR00T, wobei jede Tabelle einen Namen mit TEST_T einem Präfix hat.

Transformationsregeln und Aktionen

Sie verwenden die Transformationsaktionen, um alle Transformationen anzugeben, die Sie auf das ausgewählte Schema, Tabelle oder Ansicht anwenden möchten. Transformationsregeln sind optional.

Einschränkungen

- Sie können nicht mehr als eine Transformationsregelaktion auf dasselbe Objekt (Schema, Tabelle, Spalte, Tabellen-Tablespace oder Index-Tablespace) anwenden. Sie können mehrere Transformationsregelaktionen auf jeder Ebene anwenden, solange jede Transformationsaktion auf ein anderes Objekt angewendet wird.
- Bei Tabellennamen und Spaltennamen in Transformationsregeln muss die Groß- und Kleinschreibung beachtet werden. Beispielsweise müssen Sie Tabellennamen und Spaltennamen für eine Oracle- oder Db2-Datenbank in Großbuchstaben angeben.
- Transformationen werden für Spaltennamen in Sprachen mit Schreibrichtung von rechts nach links nicht unterstützt.
- Transformationen können nicht für Spalten durchgeführt werden, deren Name Sonderzeichen (z. B. #, \, /, -) enthält.

- Die einzige unterstützte Transformation für Spalten, die BLOB/CLOB-Datentypen zugeordnet sind, besteht darin, die Spalte auf dem Ziel abzulegen.
- AWS DMS unterstützt nicht die Replikation von zwei Quelltabellen in eine einzelne Zieltabelle. AWS DMS repliziert Datensätze von Tabelle zu Tabelle und von Spalte zu Spalte gemäß den Transformationsregeln der Replikationsaufgabe. Die Objektnamen müssen eindeutig sein, um Überschneidungen zu vermeiden.

Beispielsweise hat eine Quelltable eine Spalte mit dem Namen ID und die entsprechende Zieltabelle hat eine bereits existierende Spalte mit dem Namen id. Wenn eine Regel eine ADD-COLUMN-Anweisung verwendet, um eine neue Spalte mit dem Namen id hinzuzufügen, und eine SQLite-Anweisung, um die Spalte mit benutzerdefinierten Werten zu füllen, erzeugt dies ein doppeltes, mehrdeutiges Objekt mit dem Namen id und dies wird nicht unterstützt.

Werte

Für Tabellenzuweisungsregeln, die den Regeltyp "Transformation" verwenden, können Sie die folgenden Werte anwenden.

Parameter	Mögliche Werte	Beschreibung
<code>rule-type</code>	<code>transformation</code>	Ein Wert, mit dem die Regel auf jedes von der Auswahlregel festgelegte Objekt angewendet wird. Verwenden Sie <code>transformation</code> , falls nicht anders angegeben.
<code>rule-id</code>	Ein numerischer Wert.	Ein eindeutiger numerischer Wert zur Identifizierung der Regel. Wenn Sie mehrere Transformationsregeln für dasselbe Objekt angeben (Schema, Tabelle, Spalte, tabellenübergreifender Bereich oder Index-Tabellenbereich), AWS DMS wendet die Transformationsregel mit der niedrigeren Regel-ID an.

Parameter	Mögliche Werte	Beschreibung
<code>rule-name</code>	Ein alphanumerischer Wert.	Ein eindeutiger Name zur Identifizierung der Regel.

Parameter	Mögliche Werte	Beschreibung
<code>object-locator</code>	<p>Ein Objekt mit den folgenden Parametern:</p> <ul style="list-style-type: none"> • <code>schema-name</code> – Der Name des Schemas. Für MongoDB- und Amazon-DocumentDB-Endpunkte ist dies der Name der Datenbank, die eine Reihe von Sammlungen enthält. • <code>table-name</code> – Der Name der Tabelle, Ansicht oder Sammlung. • <code>table-tablespace-name</code> – Der Name eines vorhandenen Tabellenspaces. • <code>index-tablespace-name</code> – Der Name eines vorhandenen Index-Tabellenspaces. • <code>column-name</code> – Der Name einer vorhandenen Spalte. • <code>data-type</code> – Der Name eines vorhandenen Spaltentyps. 	<p>Der Name der einzelnen Schemata, Tabellen oder Ansichten, Tabellen-Tabellenräume, Index-Tabellenräume und Spalten, für die die Regel gilt. Sie können das Prozentzeichen „%“ als Platzhalter für alle oder einen Teil der Werte der einzelnen <code>object-locator</code>-Parameter verwenden, ausgenommen <code>data-type</code>. Auf diese Weise können Sie diese Elemente aufeinander abstimmen:</p> <ul style="list-style-type: none"> • Eine einzige Tabelle oder Ansicht in einem einzigen Schema • Eine einzige Tabelle oder Ansicht in einigen oder allen Schemas • Einige oder alle Tabellen und Ansichten in einem einzigen Schema • Einige oder alle Tabellen und Ansichten in einigen oder allen Schemas • Eine oder mehrere Spalten in der angegebenen Tabelle(n), Ansicht(en) oder Schema(ta). • Die Spalten mit einem gegebenen <code>data-type</code>, wenn mehrere Spalten angegeben werden. Die möglichen Werte von <code>data-type</code> finden Sie unter <code>data-type</code> nach dieser Tabelle beschrieben.

Parameter	Mögliche Werte	Beschreibung
		<p>Darüber hinaus ist der Parameter <code>table-tablespace-name</code> oder <code>index-tablespace-name</code> nur verfügbar, um mit einem Oracle-Quellendpunkt übereinzustimmen. Sie können entweder <code>table-tablespace-name</code> oder <code>index-tablespace-name</code> in einer einzelnen Regel angeben, aber nicht beides. Daher können Sie eines der folgenden Elemente abstimmen:</p> <ul style="list-style-type: none">• Ein einzelner, einige oder alle Tabellen-Tabellenräume• Ein einzelner, einige oder alle Index-Tabellenräume

Parameter	Mögliche Werte	Beschreibung
<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>Die Transformation, die Sie auf das Objekt anwenden möchten. Bei allen Transformationsregelaktionen wird zwischen Groß- und Kleinschreibung unterschieden.</p> <p>Der <code>add-column</code> -Wert des <code>rule-action</code> -Parameters fügt einer Tabelle eine Spalte hinzu. Sie können jedoch keine neue Spalte mit dem gleichen Namen wie eine vorhandene Spalte derselben Tabelle hinzufügen.</p> <p><code>add-column</code> gibt bei Verwendung mit den Parametern <code>expression</code> und <code>data-type</code> den Wert der neuen Spaltendaten an.</p> <p>Der <code>change-data-type</code> -Wert für <code>rule-action</code> ist nur für <code>column</code>-Regelziele verfügbar.</p> <p>Der <code>include-column</code> -Wert des <code>rule-action</code> -Parameters ändert den Modus der Tabelle, so dass standardmäßig alle Spalten gelöscht und die angegebenen Spalten eingeschlossen werden. Durch mehrmaliges Aufrufen der <code>include-column</code> -Regel werden mehrere Spalten in das Ziel aufgenommen.</p> <p>Sie können eine <code>define-primary-key</code> -Regel nicht verwenden, wenn die Regel einen</p>

Parameter	Mögliche Werte	Beschreibung
		<p>Platzhalter (%) in einem Schema- oder Tabellennamen enthält.</p> <p>Für eine bestehende Aufgabe werden Transformationsregelaktionen, die das Zieltabellenschema ändern, wie <code>remove-column</code>, <code>rename</code> oder <code>add-prefix</code>, erst wirksam, wenn Sie die Aufgabe neu starten. Wenn Sie die Aufgabe nach dem Hinzufügen der Transformationsregel fortsetzen, stellen Sie möglicherweise ein unerwartetes Verhalten für die geänderte Spalte fest, zu dem auch fehlende Spaltendaten gehören können. Ein Neustart der Aufgabe ist erforderlich, um sicherzustellen, dass die Transformationsregel ordnungsgemäß funktioniert.</p>
<code>rule-target</code>	<code>schema, table, column, table-tablespace, index-tablespace</code>	<p>Der Objekttyp, den Sie transformieren.</p> <p>Die Werte <code>table-tablespace</code> und <code>index-tablespace</code> sind nur für einen Oracle-Zielendpunkt verfügbar.</p> <p>Stellen Sie sicher, dass Sie einen Wert für den Parameter als Teil des Namens <code>object-locator</code>: <code>table-tablespace-name</code> oder <code>index-tablespace-name</code> angeben.</p>

Parameter	Mögliche Werte	Beschreibung
<code>value</code>	Ein alphanumerischer Wert, der den Benennungsregeln für den Zieltyp folgt.	Der neue Wert für Aktionen, die eine Eingabe erfordern, z. B. <code>rename</code> .
<code>old-value</code>	Ein alphanumerischer Wert, der den Benennungsregeln für den Zieltyp folgt.	Der alte Wert für Aktionen, die Ersatz erfordern, z. B. <code>replace-prefix</code> .

Parameter	Mögliche Werte	Beschreibung
data-type	<p>type – Der Datentyp, der verwendet werden soll, wenn rule-action add-column ist, bzw. der Ersatzdatentyp, wenn rule-action change-data-type ist.</p> <p>Oder der Name des Ersetzungsdentyps, wenn rule-action change-data-type , der Wert von column-name "%" ist, und ein zusätzlicher data-type -Parameter zur Identifizierung des vorhandenen Datentyps in der object-locator enthalten ist.</p> <p>AWS DMS unterstützt Transformationen von Spaltendatentypen für die folgenden DMS-Datentypen: "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 – Wenn der hinzugefügte Spalten- oder Ersetzungsdentyp eine Genauigkeit aufweist, wird ein ganzzahliger Wert zur Angabe der Genauigkeit angegeben.</p>	<p>Im Folgenden sehen Sie ein Beispiel für einen data-type -Parameter zur Angabe des vorhandenen Datentyps, der ersetzt werden soll.</p> <pre>{ "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>

Parameter	Mögliche Werte	Beschreibung
	<p><code>scale</code> – Wenn der hinzugefügte Spalten- oder der Ersatzdatentyp eine Skala, einen Ganzzahlwert oder einen Datums-/Uhrzeitwert hat, um den Maßstab anzugeben.</p> <p><code>length</code> – Die Länge der neuen Spaltendaten (bei Verwendung mit <code>add-column</code>)</p>	Hier wird die Spalte <code>col10</code> der Tabelle <code>table_t</code> zum Datentyp <code>string</code> geändert.

Parameter	Mögliche Werte	Beschreibung
expression	Ein alphanumerischer Wert, der der SQLite-Syntax folgt.	<p>Bei Verwendung mit dem <code>rule-action</code> -Wert <code>rename-schema</code> gibt der <code>expression</code> -Parameter ein neues Schema an. Bei Verwendung mit dem <code>rule-action</code> -Wert <code>rename-table</code> gibt <code>expression</code> eine neue Tabelle an. Bei Verwendung mit dem <code>rule-action</code> -Wert <code>rename-column</code> gibt <code>expression</code> einen neuen Spaltennamen an.</p> <p>Bei Verwendung mit dem <code>rule-action</code> -Wert <code>add-column</code> gibt <code>expression</code> Daten an, aus denen eine neue Spalte besteht.</p> <p>Beachten Sie, dass für diesen Parameter nur Ausdrücke unterstützt werden. Operatoren und Befehle werden nicht unterstützt.</p> <p>Weitere Informationen zum Verwenden von Ausdrücken für Transformationsregeln finden Sie unter Verwenden von Transformationsregelausdrücken zum Definieren von Spalteninhalten.</p> <p>Weitere Informationen zu SQLite-Ausdrücken finden Sie unter Verwenden von SQLite-Funktionen zum Erstellen von Ausdrücken.</p>

Parameter	Mögliche Werte	Beschreibung
<code>primary-key-def</code>	<p>Ein Objekt mit den folgenden Parametern:</p> <ul style="list-style-type: none">• <code>name</code> – Der Name eines neuen Primärschlüssels oder eines eindeutigen Index für die Tabelle oder Ansicht.• (Optional) <code>origin</code> – Der Typ des eindeutigen Schlüssels, der definiert werden soll: <code>primary-key</code> (Standard) oder <code>unique-index</code>.• <code>columns</code> – Ein Array von Zeichenfolgen mit den Namen von Spalten in der Reihenfolge, in der sie im Primärschlüssel oder eindeutigen Index aufscheinen.	<p>Dieser Parameter kann den Namen, Typ und Inhalt eines eindeutigen Schlüssels in der transformierten Tabelle oder Ansicht definieren. Dies geschieht, wenn <code>rule-action</code> auf <code>define-primary-key</code> und <code>rule-target</code> auf <code>table</code> gesetzt ist. Standardmäßig ist der eindeutige Schlüssel als Primärschlüssel definiert.</p>

Parameter	Mögliche Werte	Beschreibung
<p><code>before-image-def</code></p>	<p>Ein Objekt mit den folgenden Parametern:</p> <ul style="list-style-type: none"> • <code>column-prefix</code> – Ein Wert, der einem Spaltennamen vorangestellt wird. Der Standardwert ist <code>BI_</code>. • <code>column-suffix</code> – Ein Wert, der an den Spaltennamen angehängt wird. Der Standardwert ist leer. • <code>column-filter</code> – Erfordert einen der folgenden Werte: <code>pk-only</code> (Standard), <code>non-lob</code> (optional) und <code>all</code> (optional). 	<p>Dieser Parameter definiert eine Benennungskonvention zum Identifizieren der Vorher-Abbild-Spalten und gibt einen Filter an, um zu identifizieren, für welche Quellspalten Vorher-Abbild-Spalten auf dem Ziel erstellt werden können. Sie können diesen Parameter angeben, wenn <code>rule-action</code> auf <code>add-before-image-columns</code> und <code>rule-target</code> auf <code>column</code> gesetzt ist.</p> <p>Setzen Sie nicht <code>column-prefix</code> und <code>column-suffix</code> auf leere Zeichenfolgen.</p> <p>Wählen Sie bei <code>column-filter</code> folgende Optionen:</p> <ul style="list-style-type: none"> • <code>pk-only</code> – Zum Hinzufügen nur von Spalten, die Teil der Primärschlüssel der Tabelle sind. • <code>non-lob</code> – Zum Hinzufügen nur von Spalten, die nicht den Typ LOB haben. • <code>all</code> – Zum Hinzufügen einer Spalte, die einen Vorher-Abbild-Wert hat. <p>Weitere Informationen zur Unterstützung von Vorher-Abbildern für AWS DMS -Zielendpunkte finden Sie unter:</p> <ul style="list-style-type: none"> • Verwenden eines Vorher-Abbilds zum Anzeigen von Originalwerten

Parameter	Mögliche Werte	Beschreibung
		<ul style="list-style-type: none">von CDC-Zeilen für einen Kinesis-Datenstrom als ZielVerwenden eines Vorher-Abbilds zum Anzeigen von Originalwerten von CDC-Zeilen für Apache Kafka als Ziel

Beispiele

Example Umbenennen eines Schemas

Im folgenden Beispiel wird ein Schema von Test in Ihrer Quelle auf Test1 in Ihrem Ziel umbenannt.

```
{
  "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 Umbenennen einer Tabelle

Im folgenden Beispiel wird eine Tabelle von Actor in Ihrer Quelle auf Actor1 in Ihrem Ziel umbenannt.

```
{
  "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 Umbenennen einer Spalte

Im folgenden Beispiel wird eine Spalte in Tabelle Actor von first_name in Ihrer Quelle auf fname in Ihrem Zielpunkt umbenannt.

```
{
  "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 Umbenennen eines Oracle-Tabellen-Tabellenraums

Im folgenden Beispiel wird der Tabellen-Tabellenraum mit dem Namen SetSpace für eine Tabelle mit dem Namen Actor in Ihrer Oracle-Quelle auf SceneTblSpace in Ihrem Oracle-Zielendpunkt umbenannt.

```

{
  "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 Umbenennen eines Oracle-Index-Tabellenraums

Im folgenden Beispiel wird der Index-Tabellenraum mit dem Namen SetISpace für eine Tabelle mit dem Namen Actor in Ihrer Oracle-Quelle auf SceneIdxSpace in Ihrem Oracle-Zielendpunkt umbenannt.

```

{
  "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 Hinzufügen einer Spalte

Im folgenden Beispiel wird der Tabelle Actor im Schema test eine datetime- Spalte hinzugefügt.

```
{  
  "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 Entfernen einer Spalte

Im folgenden Beispiel wird die Tabelle mit dem Namen Actor in Ihrer Quelle so transformiert, dass alle Spalten, die mit den Zeichen col1 beginnen, in Ihrem Ziel daraus entfernt werden.


```
{
  "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 In Kleinbuchstaben konvertieren

Im folgenden Beispiel wird ein Tabellenname von ACTOR in Ihrer Quelle auf actor in Ihrem Ziel umgewandelt.

```
{
  "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 Umwandeln in Großschreibung

Im folgenden Beispiel werden alle Spalten in allen Tabellen und allen Schemata von Kleinschreibung an Ihrer Quelle in Großschreibung an Ihrem Ziel umgewandelt.

```
{
  "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 Hinzufügen eines Präfix

Im folgenden Beispiel werden alle Tabellen in Ihrer Quelle so transformiert, dass ihnen das Präfix DMS_ in Ihrem Ziel hinzugefügt wird.

```
{
  "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 Ersetzen eines Präfix

Im folgenden Beispiel werden alle Spalten, die das Präfix Pre_ in Ihrer Quelle enthalten, so transformiert, dass das Präfix durch NewPre_ in Ihrem Ziel ersetzt wird.

```
{
  "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 Entfernen eines Suffix

Im folgenden Beispiel werden alle Tabellen in Ihrer Quelle so transformiert, dass aus ihnen das Suffix `_DMS` in Ihrem Ziel entfernt wird.

```

{
  "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 Definieren eines Primärschlüssels

Das folgende Beispiel definiert einen Primärschlüssel mit dem Namen `ITEM-primary-key` auf drei Spalten der `ITEM`-Tabelle, die zu Ihrem Zielendpunkt migriert wurde.

```

{
  "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 Definieren eines eindeutigen Index

Das folgende Beispiel definiert einen eindeutigen Index mit dem Namen `ITEM-unique-idx` auf drei Spalten der `ITEM`-Tabelle, die zu Ihrem Zielendpunkt migriert wurde.

```
{
  "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 Ändern des Datentyps der Zielspalte

Im folgenden Beispiel wird der Datentyp einer Zielspalte namens `SALE_AMOUNT` von einem vorhandenen Datentyp in `int8` geändert.

```
{
```

```

"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 Hinzufügen einer Vorher-Abbild-Spalte

Für eine Quellspalte mit dem Namen emp_no fügt die Transformationsregel im folgenden Beispiel eine neue Spalte mit dem Namen BI_emp_no auf dem Ziel hinzu.

```

{
  "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"
}
}
]
}
```

Hier füllt die folgende Anweisung eine BI_emp_no-Spalte in der entsprechenden Zeile mit 1.

```
UPDATE employees SET emp_no = 3 WHERE BI_emp_no = 1;
```

Beim Schreiben von CDC-Aktualisierungen für unterstützte AWS DMS Ziele lässt sich BI_emp_no anhand der Spalte erkennen, welche Zeilen in der Spalte aktualisierte Werte enthalten. emp_no

Verwenden von Transformationsregelausdrücken zum Definieren von Spalteninhalten

Um Inhalte für neue und vorhandene Spalten zu definieren, können Sie einen Ausdruck innerhalb einer Transformationsregel verwenden. Beispielsweise können Sie mithilfe von Ausdrücken eine Spalte hinzufügen oder Quelltabellenüberschriften zu einem Ziel replizieren. Sie können Ausdrücke auch verwenden, um Datensätze in Zieltabellen als eingefügt, aktualisiert oder an der Quelle gelöscht zu kennzeichnen.

Themen

- [Hinzufügen einer Spalte mithilfe eines Ausdrucks](#)
- [Markieren von Zieldatensätzen mithilfe eines Ausdrucks](#)
- [Replizieren von Quelltabellenheadern mithilfe von Ausdrücken](#)
- [Verwenden von SQLite-Funktionen zum Erstellen von Ausdrücken](#)
- [Hinzufügen von Metadaten zu einer Zieltabelle mithilfe von Ausdrücken](#)

Hinzufügen einer Spalte mithilfe eines Ausdrucks

Verwenden Sie eine add-column-Regelaktion und ein column-Regelziel, um Tabellen mithilfe eines Ausdrucks in einer Transformationsregel Spalten hinzuzufügen.

Im folgenden Beispiel wird der ITEM-Tabelle eine neue Spalte hinzugefügt. Der Name der neuen Spalte ist FULL_NAME, der Datentyp string, und die Länge ist 50 Zeichen. Der Ausdruck verkettet die Werte zweier vorhandener Spalten, FIRST_NAME und LAST_NAME, so dass sie zu FULL_NAME

ausgewertet werden. `schema-name`, `table-name` und Ausdrucksparameter und beziehen sich auf Objekte in der Quelldatenbanktabelle. `value` und der `data-type`-Block bezieht sich auf Objekte in der Zieldatenbanktabelle.

```
{
  "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
      }
    }
  ]
}
```

Markieren von Zieldatensätzen mithilfe eines Ausdrucks

Um Datensätze in Zieltabellen als eingefügt, aktualisiert oder gelöscht in der Quelltable zu kennzeichnen, verwenden Sie einen Ausdruck in einer Transformationsregel. Der Ausdruck verwendet eine `operation_indicator`-Funktion, um Datensätze zu kennzeichnen. Datensätze, die aus der Quelle gelöscht werden, werden nicht aus dem Ziel gelöscht. Stattdessen wird der

Zieldatensatz mit einem vom Benutzer angegebenen Wert markiert, um anzuzeigen, dass er aus der Quelle gelöscht wurde.

Note

Die `operation_indicator`-Funktion funktioniert nur für Tabellen, die einen Primärschlüssel für die Quell- und die Zieldatenbank haben.

Beispielsweise fügt die folgende Transformationsregel einer Zieltabelle zuerst eine neue `Operation`-Spalte hinzu. Anschließend wird die Spalte mit dem Wert `D` aktualisiert, wenn ein Datensatz aus einer Quelltable gelöscht wird.

```
{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-target": "column",
  "object-locator": {
    "schema-name": "%",
    "table-name": "%"
  },
  "rule-action": "add-column",
  "value": "Operation",
  "expression": "operation_indicator('D', 'U', 'I')",
  "data-type": {
    "type": "string",
    "length": 50
  }
}
```

Replizieren von Quelltabellenheadern mithilfe von Ausdrücken

Standardmäßig werden Header für Quelltabellen nicht auf das Ziel repliziert. Um anzugeben, welche Header repliziert werden sollen, verwenden Sie eine Transformationsregel mit einem Ausdruck, der den Spaltenheader der Tabelle enthält.

Sie können die folgenden Spaltenkopfzeilen in Ausdrücken verwenden.

Header	Wert bei fortlaufender Replikation	Wert bei vollständigem Laden	Datentyp
AR_H_STREAM_POSITION	Der Wert der Stream-Position aus der Quelle. Dieser Wert kann je nach Quellendpunkt die Systemänderungsnummer (SCN) oder die Protokollsequenznummer (LSN) sein.	Eine leere Zeichenfolge.	STRING
AR_H_TIMESTAMP	Ein Zeitstempel, der den Zeitpunkt der Änderung angibt.	Ein Zeitstempel, der die aktuelle Uhrzeit angibt, zu der Daten am Ziel ankommen.	DATETIME (scale=7)
AR_H_COMMIT_TIMESTAMP	Ein Zeitstempel, der den Zeitpunkt des Commits angibt.	Ein Zeitstempel, der die aktuelle Uhrzeit angibt.	DATETIME (scale=7)
AR_H_OPERATION	INSERT, UPDATE oder DELETE	INSERT	STRING
AR_H_USER	Der Benutzername, die ID oder andere Informationen, die die Quelle über den Benutzer bereitstellt, der die Änderung vorgenommen hat. Dieser Header wird nur auf den Quellendpunkten von SQL Server und Oracle	Die Transformation, die Sie auf das Objekt anwenden möchten. Bei allen Transformationsregelaktionen wird zwischen Groß- und Kleinschreibung unterschieden.	STRING

Header	Wert bei fortlaufender Replikation	Wert bei vollständigem Laden	Datentyp
	(Version 11.2.0.3 und höher) unterstützt.		
AR_H_CHANGE_SEQ	Eine eindeutige inkrementierende Zahl aus der Quelldatenbank, die aus einem Zeitstempel und einer automatisch inkrementierenden Zahl besteht. Der Wert hängt vom Quelldatenbanksystem ab.	Eine leere Zeichenfolge.	STRING

Im folgenden Beispiel wird dem Ziel eine neue Spalte hinzugefügt, indem der Wert für die Stream-Position aus der Quelle verwendet wird. Bei SQL Server ist der Wert für die Stream-Position der LSN für den Quellendpunkt. Bei Oracle ist der Wert für die Stream-Position der SCN für den Quellendpunkt.

```
{
  "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
  }
}
```

```
}
```

Das folgende Beispiel fügt dem Ziel eine neue Spalte hinzu, die eine eindeutige inkrementelle Zahl aus der Quelle hat. Dieser Wert steht für eine 35-stellige eindeutige Zahl auf Aufgabenebene. Die ersten 16 Ziffern sind Teil eines Zeitstempels und die letzten 19 Ziffern sind die vom DBMS inkrementierte Record_ID-Nummer.

```
{
  "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
  }
}
```

Verwenden von SQLite-Funktionen zum Erstellen von Ausdrücken

Mit den Tabelleneinstellungen können Sie alle Einstellungen angeben, die Sie auf die ausgewählte Tabelle oder Ansicht für eine bestimmte Operation anwenden möchten. Regeln für Tabelleneinstellungen sind optional.

Note

Anstelle des Konzepts von Tabellen und Ansichten speichern MongoDB- und DocumentDB-Datenbanken Datensätze als Dokumente, die in Sammlungen zusammengefasst werden. Wenn Sie also von einer MongoDB- oder DocumentDB-Quelle migrieren, sollten Sie den Bereichssegmentierungstyp der parallelen Ladeinstellungen für ausgewählte Sammlungen und nicht für Tabellen und Ansichten berücksichtigen.

Themen

- [Verwenden eines CASE-Ausdrucks](#)
- [Beispiele](#)

Im Folgenden finden Sie Zeichenkettenfunktionen, mit denen Sie Ausdrücke für Transformationsregeln erstellen können.

Zeichenfolgenfunktionen	Beschreibung
<code>lower(x)</code>	Die <code>lower(x)</code> -Funktion gibt eine Kopie der Zeichenfolge <code>x</code> zurück, in der alle Zeichen in Kleinbuchstaben umgewandelt wurden. Die integrierte <code>lower</code> -Standardfunktion funktioniert nur für ASCII-Zeichen.
<code>upper(x)</code>	Die <code>upper(x)</code> -Funktion gibt eine Kopie der Zeichenfolge <code>x</code> zurück, in der alle Zeichen in Großbuchstaben umgewandelt wurden. Die integrierte <code>upper</code> -Standardfunktion funktioniert nur für ASCII-Zeichen.
<code>ltrim(x,y)</code>	Die <code>ltrim(x,y)</code> -Funktion gibt eine Zeichenfolge zurück, die gebildet wird, indem alle Zeichen, die in <code>y</code> vorkommen, von der linken Seite von <code>x</code> entfernt werden. Wenn es keinen Wert für <code>y</code> gibt, entfernt <code>ltrim(x)</code> Leerzeichen auf der linken Seite von <code>x</code> .
<code>replace(x,y,z)</code>	Die <code>replace(x,y,z)</code> -Funktion gibt eine Zeichenfolge zurück, die gebildet wird, indem jedes Vorkommen der Zeichenfolge <code>y</code> in der Zeichenfolge <code>x</code> durch die Zeichenfolge <code>z</code> ersetzt wird.
<code>rtrim(x,y)</code>	Die <code>rtrim(x,y)</code> -Funktion gibt eine Zeichenfolge zurück, die gebildet wird, indem alle Zeichen, die in <code>y</code> vorkommen, von der rechten Seite von <code>x</code> entfernt werden. Wenn es keinen Wert für <code>y</code> gibt, entfernt <code>rtrim(x)</code> Leerzeichen auf der rechten Seite von <code>x</code> .
<code>substr(x,y,z)</code>	Die <code>substr(x,y,z)</code> -Funktion gibt eine Teilzeichenfolge der Eingabezeichenfolge <code>x</code> zurück, die mit <code>y</code> -ten Zeichen beginnt und <code>z</code> Zeichen lang ist.

Zeichenfolgenfunktionen	Beschreibung
	<p>Wenn z nicht angegeben ist, gibt <code>substr(x, y)</code> alle Zeichen bis zum Ende der Zeichenfolge x zurück, die mit dem y-ten Zeichen beginnt. Das am weitesten links stehende Zeichen von x ist die Zahl 1. Wenn y negativ ist, wird das erste Zeichen der Teilzeichenfolge ermittelt, indem von rechts statt von links gezählt wird. Wenn z negativ ist, werden die $\text{abs}(z)$ Zeichen vor dem y-ten Zeichen zurückgegeben. Wenn es sich bei x um eine Zeichenfolge handelt, beziehen sich die Indizes der Zeichen auf tatsächliche UTF-8-Zeichen. Wenn es sich bei x um ein BLOB handelt, beziehen sich die Indizes auf Bytes.</p>
<code>trim(x, y)</code>	<p>Die <code>trim(x, y)</code>-Funktion gibt eine Zeichenfolge zurück, die gebildet wird, indem alle Zeichen, die in y vorkommen, von beiden Seiten von rechten Seite von x entfernt werden. Wenn es keinen Wert für y gibt, entfernt <code>trim(x)</code> Leerzeichen von beiden Seiten von x.</p>

Im Folgenden finden Sie LOB-Funktionen, mit denen Sie Ausdrücke für Transformationsregeln erstellen können.

LOB-Funktionen	Beschreibung
<code>hex(x)</code>	<p>Die <code>hex</code>-Funktion empfängt ein BLOB als Argument und gibt eine hexadezimale Zeichenkettenversion des BLOB-Inhalts in Großbuchstaben zurück.</p>
<code>randblob (N)</code>	<p>Die <code>randblob(N)</code>-Funktion gibt ein N-Byte-BLOB zurück, das pseudozufällige Bytes enthält. Wenn N kleiner als 1 ist, wird ein zufälliges 1-Byte-BLOB zurückgegeben.</p>
<code>zeroblob(N)</code>	<p>Die <code>zeroblob(N)</code>-Funktion gibt ein BLOB zurück, das aus N Bytes von 0x00 besteht.</p>

Im Folgenden finden Sie numerische Funktionen, mit denen Sie Ausdrücke für Transformationsregeln erstellen können.

Numerische Funktionen	Beschreibung
abs(<i>x</i>)	Die abs(<i>x</i>)-Funktion gibt den absoluten Wert des numerischen Arguments <i>x</i> zurück. Die abs(<i>x</i>)-Funktion gibt NULL zurück, wenn <i>x</i> NULL ist. Die abs(<i>x</i>)-Funktion gibt 0.0 zurück, wenn <i>x</i> eine Zeichenfolge oder ein BLOB ist, die/das nicht in einen numerischen Wert umgewandelt werden kann.
random()	Die random-Funktion gibt eine pseudozufällige Ganzzahl zwischen -9 223 372 036 854 775 808 und +9 223 372 036 854 775 807 zurück.
round(<i>x</i> , <i>y</i>)	Die round(<i>x</i> , <i>y</i>)-Funktion gibt einen Gleitkommawert <i>x</i> zurück, der auf <i>y</i> Ziffern rechts vom Dezimaltrennzeichen gerundet ist. Wenn es keinen Wert für <i>y</i> gibt, wird davon ausgegangen, dass er 0 ist.
max(<i>x</i> , <i>y</i> ...)	<p>Die max-Funktion mit mehreren Argumenten gibt das Argument mit dem Maximalwert zurück oder NULL, wenn ein Argument NULL ist.</p> <p>Die max-Funktion durchsucht ihre Argumente von links nach rechts nach einem Argument, das eine Sortierfunktion definiert. Wenn eine gefunden wird, verwendet sie diese Sortierfunktion für alle Zeichenkettenvergleiche. Wenn keines der Argumente zur max-Definition einer Sortierfunktion vorhanden ist, wird die BINARY-Sortierfunktion verwendet. Die max-Funktion ist eine einfache Funktion, wenn sie zwei oder mehr Argumente hat, aber sie funktioniert als Aggregatfunktion, wenn sie ein einziges Argument hat.</p>
min(<i>x</i> , <i>y</i> ...)	Die min-Funktion mit mehreren Argumenten gibt das Argument mit dem Mindestwert zurück.

Numerische Funktionen	Beschreibung
	<p>Die <code>min</code>-Funktion durchsucht ihre Argumente von links nach rechts nach einem Argument, das eine Sortierfunktion definiert. Wenn eine gefunden wird, verwendet sie diese Sortierfunktion für alle Zeichenkettenvergleiche. Wenn keines der Argumente zur <code>min</code>-Definition einer Sortierfunktion vorhanden ist, wird die <code>BINARY</code>-Sortierfunktion verwendet. Die <code>min</code>-Funktion ist eine einfache Funktion, wenn sie zwei oder mehr Argumente hat, aber sie funktioniert als Aggregatfunktion, wenn sie ein einziges Argument hat.</p>

Im Folgenden finden Sie `NULL`-Prüffunktionen, mit denen Sie Ausdrücke für Transformationsregeln erstellen können.

Funktionen zur <code>NULL</code> -Prüfung	Beschreibung
<code>coalesce (x,y...)</code>	<p>Die <code>coalesce</code>-Funktion gibt eine Kopie ihres ersten Arguments zurück, das nicht <code>NULL</code> ist, aber sie gibt <code>NULL</code> zurück, wenn alle Argumente <code>NULL</code> sind. Die Koaleszenzfunktion hat mindestens zwei Argumente.</p>
<code>ifnull(x,y)</code>	<p>Die <code>ifnull</code>-Funktion gibt eine Kopie ihres ersten Arguments zurück, das nicht <code>NULL</code> ist, aber sie gibt <code>NULL</code> zurück, wenn beide Argumente <code>NULL</code> sind. Die <code>ifnull</code>-Funktion hat genau zwei Argumente. Die <code>ifnull</code>-Funktion ist dieselbe wie <code>coalesce</code> mit zwei Argumenten.</p>
<code>nullif(x,y)</code>	<p>Die <code>nullif(x,y)</code>-Funktion gibt eine Kopie ihres ersten Arguments zurück, wenn die Argumente unterschiedlich sind, aber sie gibt <code>NULL</code> zurück, wenn die Argumente identisch sind.</p> <p>Die <code>nullif(x,y)</code>-Funktion durchsucht ihre Argumente von links nach rechts nach einem Argument, das eine Sortierfunktion definiert. Wenn eine gefunden wird, verwendet sie diese Sortierfunktion für alle Zeichenkettenvergleiche. Wenn keines</p>

Funktionen zur NULL-Prüfung	Beschreibung
	der Argumente zur Definition einer Sortierfunktion vorhanden ist, wird die BINARY-Sortierfunktion verwendet.

Im Folgenden finden Sie Datums- und Zeitfunktionen, mit denen Sie Ausdrücke für Transformationsregeln erstellen können.

Datums- und Zeitfunktionen	Beschreibung
<code>date(<i>timestring</i> , <i>modifier</i> , <i>modifier</i>...)</code>	Die <code>date</code> -Funktion gibt das Datum im Format JJ-MM-TT zurück.
<code>time(<i>timestring</i> , <i>modifier</i> , <i>modifier</i>...)</code>	Die <code>time</code> -Funktion gibt die Uhrzeit im Format HH:MM:SS zurück.
<code>datetime(<i>timestring</i> , <i>modifier</i> , <i>modifier</i>...)</code>	Die <code>datetime</code> -Funktion gibt das Datum und die Uhrzeit im Format JJJJ-MM-TT HH:MM:SS zurück.
<code>julianday(<i>timestrin</i> <i>g</i> , <i>modifier</i> , <i>modifier</i>...)</code>	Die <code>julianday</code> -Funktion gibt die Anzahl der Tage seit dem Mittag in Greenwich am 24. November 4714 v. Chr. zurück.
<code>strftime(<i>format</i> , <i>timestring</i> , <i>modifier</i> , <i>modifier</i>...)</code>	Die <code>strftime</code> -Funktion gibt das Datum gemäß der als erstes Argument angegebenen Formatzeichenfolge zurück, wobei eine der folgenden Variablen verwendet wird: %d: Tag des Monats %H: Stunde 00-24 %f: ** Sekundenbruchteile SS.SSS %j: Tag des Jahres 001-366 %J: ** Julianische Tageszahl %m: Monat 01–12 %M: Minute (00–59)

Datums- und Zeitfunktionen	Beschreibung
	%s: Sekunden seit dem 01.01.1970
	%S: Sekunden (00–59)
	%w: Wochentag (0–6; wobei Sonntag =0 ist)
	%W: Woche des Jahres 00–53
	%Y: Jahr 0000–9999
	%%: %

Im Folgenden finden Sie eine Hash-Funktionen, mit der Sie Ausdrücke für Transformationsregeln erstellen können.

Hash-Funktion	Beschreibung
hash_sha256(<i>x</i>)	<p>Die hash-Funktion generiert einen Hashwert für eine Eingabespalte (unter Verwendung des SHA-256-Algorithmus) und gibt den Hexadezimalwert des generierten Hashwerts zurück.</p> <p>Um die hash-Funktion in einem Ausdruck zu verwenden, fügen Sie dem Ausdruck hash_sha256(<i>x</i>) hinzu und ersetzen Sie <i>x</i> durch den Namen der Quellspalte.</p>

Verwenden eines CASE-Ausdrucks

Der SQLite-Ausdruck CASE wertet eine Liste von Bedingungen aus und gibt einen Ausdruck zurück, der auf dem Ergebnis basiert. Die Syntax wird im Folgenden angezeigt.

```

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
```

Beispiele

Example des Hinzufügens einer neuen Zeichenkettenspalte zur Zieltabelle unter Verwendung einer Case-Bedingung

Beispielsweise fügt die folgende Transformationsregel einer Zieltabelle, `employee`, zuerst eine neue Zeichenfolgenspalte, `emp_seniority`, hinzu. Sie verwendet die SQLite-Funktion `round` für die Gehaltsspalte mit einer Case-Bedingung, um zu überprüfen, ob das Gehalt 20 000 entspricht oder diesen Wert überschreitet. Ist dies der Fall, erhält die Spalte den Wert `SENIOR` und alles andere hat den Wert `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 des Hinzufügens einer neuen Datumsspalte zur Zieltabelle

Im folgenden Beispiel wird der Zieltabelle, `employee`, eine neue Datumsspalte, `createdate`, hinzugefügt. Wenn Sie die SQLite-Datumsfunktion `datetime` verwenden, wird das Datum für jede eingefügte Zeile zur neu erstellten Tabelle hinzugefügt.

```
{
  "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 des Hinzufügens einer neuen numerischen Spalte zur Zieltabelle

Im folgenden Beispiel wird der Zieltabelle, `employee`, eine neue numerische Spalte, `rounded_emp_salary`, hinzugefügt. Es verwendet die SQLite-Funktion `round`, um das gerundete Gehalt hinzuzufügen.

```
{
  "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"
    }
  }
}

```

Beispiele des Hinzufügens einer neuen Zeichenkettenspalte zur Zieltabelle unter Verwendung der Hash-Funktion

Im folgenden Beispiel wird der Zieltabelle, `employee`, eine neue Zeichenkettenspalte, `hashed_emp_number`, hinzugefügt. Die SQLite-Funktion `hash_sha256(x)` erstellt Hashwerte auf dem Ziel für die Quellspalte, `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
  }
}

```

Hinzufügen von Metadaten zu einer Zieltabelle mithilfe von Ausdrücken

Sie können die Metadateninformationen zur Zieltabelle hinzufügen, indem Sie die folgenden Ausdrücke verwenden:

- `$AR_M_SOURCE_SCHEMA` – Der Name des Quellschemas.
- `$AR_M_SOURCE_TABLE_NAME` – Der Name der Quelltable.
- `$AR_M_SOURCE_COLUMN_NAME` – Der Name einer Spalte in der Quelltable.
- `$AR_M_SOURCE_COLUMN_DATATYPE` – Der Datentyp einer Spalte in der Quelltable.

Example des Hinzufügens einer Spalte für einen Schemanamen unter Verwendung des Schemanamens aus der Quelle

Im folgenden Beispiel wird dem Ziel eine neue Spalte mit dem Namen `schema_name` hinzugefügt, indem der Schemaname aus der Quelle verwendet wird.

```
{
  "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
  }
}
```

Regeln und Operationen für Tabellen- und Sammlungseinstellungen

Mit den Tabelleneinstellungen können Sie alle Einstellungen angeben, die Sie auf die ausgewählte Tabelle oder Ansicht für eine bestimmte Operation anwenden möchten. Regeln für Tabelleneinstellungen sind optional, je nach Endpunkt- und Migrationsanforderungen.

Anstelle des Konzepts von Tabellen und Ansichten speichern MongoDB- und Amazon-DocumentDB-Datenbanken Datensätze als Dokumente, die in Sammlungen zusammengefasst werden. Eine einzelne Datenbank für einen MongoDB- oder Amazon-DocumentDB-Endpunkt ist ein bestimmter Satz von Sammlungen, die durch den Datenbanknamen identifiziert werden.

Bei der Migration von einer MongoDB- oder Amazon-DocumentDB-Quelle arbeiten Sie etwas anders mit den Einstellungen für paralleles Laden. In diesem Fall sollten Sie den Typ „Autosegmentierung“ oder „Bereichsegmentierung“ der Einstellungen für paralleles Laden für ausgewählte Sammlungen und nicht für Tabellen und Ansichten in Betracht ziehen.

Themen

- [Platzhalter in Tabelleneinstellungen sind eingeschränkt](#)
- [Verwendung des parallelen Ladens für ausgewählte Tabellen und Ansichten](#)
- [Angaben von LOB-Einstellungen für eine ausgewählte Tabelle oder Ansicht](#)
- [Beispiele für Tabelleneinstellungen](#)

Für Termersetzungsgeltern, die den Tabelleneinstellungsregeltyp geändert haben, können Sie die folgenden Parameter verwenden.

Parameter	Mögliche Werte	Beschreibung
<code>rule-type</code>	<code>table-settings</code>	Ein Wert, der die Regel auf die von der Auswahlregel festgelegte Tabelle, Ansicht oder Sammlung anwendet.
<code>rule-id</code>	Ein numerischer Wert.	Ein eindeutiger numerischer Wert zur Identifizierung der Regel.
<code>rule-name</code>	Ein alphanumerischer Wert.	Ein eindeutiger Name zur Identifizierung der Regel.
<code>object-locator</code>	Ein Objekt mit den folgenden Parametern: <ul style="list-style-type: none"> • <code>schema-name</code> – Der Name des Schemas. Für MongoDB- und Amazon-DocumentDB-Endpunkte ist dies der Name der Datenbank, die eine Reihe von Sammlungen enthält. • <code>table-name</code> – Der Name der Tabelle, Ansicht oder Sammlung. 	Der Name eines bestimmten Schemas und einer bestimmten Tabelle oder Ansicht oder der Name einer bestimmten Datenbank und Sammlung (keine Platzhalter).

Parameter	Mögliche Werte	Beschreibung
parallel-load	<p>Ein Objekt mit den folgenden Parametern:</p> <ul style="list-style-type: none"> • type – Gibt an, ob paralleles Laden aktiviert ist. <p>Wenn dies der Fall ist, gibt dieser Parameter auch den Mechanismus zur Identifizierung der Tabellen- oder Ansichtspartitionen, Unterpartitionen oder andere Segmente, die parallel geladen werden, an. Partitionen sind Segmente, die bereits definiert und in der Quelltable oder -ansicht nach Name identifiziert sind.</p> <p>Für MongoDB- und Amazon DocumentDB DocumentDB-Endpoints sind Partitionen Segmente. AWS DMS kann diese automatisch angegebenen zugehörigen Autosegmentierungsparameter berechnen. Oder Sie können diese manuell mithilfe von Bereichssegmentierungsparametern angeben.</p> <p>Nur für Oracle-Endpunkte sind Unterpartitionen eine zusätzliche Ebene von Segmenten, die in der Quelltable oder -ansicht bereits definiert und nach Name identifiziert sind. Sie können andere Segmente in der table-settings -Regel identifizieren,</p>	<p>Ein Wert, der eine parallele (Multithread-) Ladeoperation auf der von der Option object-locator identifizierten Tabelle oder Ansicht angibt. In diesem Fall können Sie parallel geladen werden. Verwenden Sie dabei eine der folgenden Methoden:</p> <ul style="list-style-type: none"> • Nach Segmenten, angegeben durch alle verfügbaren Partitionen oder Unterpartitionen. • Nach ausgewählten Partitionen und Unterpartitionen. • Per Autosegmentierung oder nach von Ihnen angegebenen bereichsbasierten Segmenten. <p>Weitere Informationen zum parallelen Laden finden Sie unter Verwendung des parallelen Ladens für ausgewählte Tabellen und Ansichten.</p>

Parameter	Mögliche Werte	Beschreibung
	<p>indem Sie Grenzen für den Bereich der Werte für eine oder mehrere Tabellen- oder Ansichtsspalten angeben.</p> <ul style="list-style-type: none">• <code>partitions</code> – Wenn <code>type partitions-list</code> ist, gibt dieser Wert alle parallel zu ladenden Partitionen an.• <code>subpartitions</code> – Nur für Oracle-Endpunkte gilt: Wenn <code>type partitions-list</code> ist, gibt dieser Wert alle parallel zu ladenden Unterpartitionen an.• <code>columns</code> – Wenn <code>type ranges</code> ist, gibt dieser Wert die Namen der Spalten an, die für die Identifizierung bereichsbasierter und parallel zu ladender Segmente verwendet werden.• <code>boundaries</code> – Wenn <code>type ranges</code> ist, gibt dieser Wert die Werte der <code>columns</code> an, die für die Identifizierung bereichsbasierter und parallel zu ladender Segmente verwendet werden.	

Parameter	Mögliche Werte	Beschreibung
type	<p>Es gilt eine der folgenden Situationen für <code>parallel-load</code> :</p> <ul style="list-style-type: none"> • <code>partitions-auto</code> – Alle Partitionen der Tabelle oder Ansicht werden parallel geladen. Jede Partition wird in ihrem eigenen Thread zugeordnet. <p>Dies ist eine erforderliche Einstellung für MongoDB- und Amazon-DocumentDB-Quellendpunkte, um die Autosegmentierungsoption einer parallelen Volllast zu verwenden.</p> <ul style="list-style-type: none"> • <code>subpartitions-auto</code> – (nur Oracle-Endpunkte) Alle Unterpartitionen der Tabelle oder Ansicht werden parallel geladen. Jede Unterpartition wird einem eigenen Thread zugeordnet. • <code>partitions-list</code> – Alle angegebenen Partitionen der Tabelle oder Ansicht werden parallel geladen. Nur für Oracle-Endpunkte: Alle angegebenen Unterpartitionen der Tabelle oder Ansicht werden parallel geladen. Jede Partition und Unterpartition, die Sie angeben, wird einem eigenen Thread zugeordnet. Sie identifizieren die parallel zu ladenden Partitionen und Unterpartitionen anhand der Partitionsnamen (<code>partitions</code>) und 	<p>Der Mechanismus zum Identifizieren der Tabellen-, Ansichts- oder Sammlungspartitionen, Unterpartitionen oder Segmente, die parallel geladen werden.</p>

Parameter	Mögliche Werte	Beschreibung
	<p>Unterpertitionsnamen (subpartitions).</p> <ul style="list-style-type: none"> • <code>ranges</code> – Alle bereichsspezifischen Segmente der Tabelle, Ansicht oder Sammlung werden parallel geladen. Jedes Tabellen-, Views- oder Sammlungssegment, das Sie identifizieren, wird seinem eigenen Thread zugeordnet. Sie geben diese Segmente nach Spaltennamen (<code>columns</code>) und Spaltenwerten (<code>boundaries</code>) an. <p>PostgreSQL-Endpunkte unterstützen nur diese Art paralleler Ladevorgänge. MongoDB und Amazon DocumentDB als Quellendpunkte unterstützen sowohl diesen Bereichssegmentierungstyp als auch den Autosegmentierungstyp einer parallelen Volllast (<code>partitions-auto</code>).</p> <ul style="list-style-type: none"> • <code>none</code> – Die Tabelle, Ansicht oder Sammlung wird unabhängig von ihren Partitionen oder Unterpertitionen in einer Single-Thread-Aufgabe (Standard) geladen. Weitere Informationen finden Sie unter Erstellen einer Aufgabe. 	

Parameter	Mögliche Werte	Beschreibung
<code>number-of-partitions</code>	(Optional) Wenn <code>type</code> für bestimmte Sammlungen eines MongoDB- oder Amazon-Endpunkts <code>partitions-auto</code> ist, gibt dieser Parameter die Gesamtzahl der Partitionen (Segmente) an, die für die Migration verwendet werden. Der Standardwert ist 16.	Gibt die Namen der Partitionen an, die parallel geladen werden.
<code>collection-count-from-metadata</code>	(Optional) Wenn <code>type</code> gilt <code>partitions-auto</code> für bestimmte Sammlungen eines MongoDB- oder Amazon DocumentDB DocumentDB-Endpunkts und dieser Parameter auf gesetzt ist <code>true</code> , AWS DMS verwendet eine geschätzte Sammlungsanzahl, um die Anzahl der Partitionen zu bestimmen. Wenn dieser Parameter auf gesetzt ist <code>false</code> , wird die tatsächliche AWS DMS Anzahl der Sammlungen verwendet. Der Standardwert ist <code>true</code> .	Gibt an, ob eine geschätzte Sammlungsanzahl oder die tatsächliche Sammlungsanzahl verwendet werden soll, um die Anzahl der Partitionen zu berechnen, die parallel geladen werden sollen.

Parameter	Mögliche Werte	Beschreibung
<code>max-records-skip-per-page</code>	(Optional) Wenn <code>type</code> für bestimmte Sammlungen eines MongoDB- oder Amazon-DocumentDB-Endpunkts <code>partitions-auto</code> ist, ist dies die Anzahl der Datensätze, die bei der Ermittlung der Grenzen für jede Partition gleichzeitig übersprungen werden sollen. AWS DMS verwendet einen Ansatz zum seitenweisen Überspringen, um die Mindestgrenze für eine Partition zu bestimmen. Der Standardwert ist 10 000.	Gibt die Anzahl der Datensätze an, die bei der Festlegung der Grenzen für jede Partition gleichzeitig übersprungen werden sollen. Das Festlegen eines relativ großen Werts vom Standard aus kann zu Cursor-Timeouts und Aufgabenfehlern führen. Das Festlegen eines relativ niedrigen Werts vom Standard aus führt zu mehr Operationen pro Seite und zu einem langsameren vollständigen Laden.
<code>batch-size</code>	(Optional) Wenn <code>type</code> für bestimmte Sammlungen eines MongoDB- oder Amazon-Endpunkts <code>partitions-auto</code> ist, begrenzt dieser Ganzzahlwert die Anzahl der Dokumente, die in einem Round-Trip-Stapel zurückgegeben werden. Wenn die Stapelgröße Null (0) ist, verwendet der Cursor die vom Server definierte maximale Stapelgröße. Der Standardwert ist 0.	Gibt die maximale Anzahl von Dokumenten an, die in einem Stapel zurückgegeben werden. Für jeden Stapel ist ein Roundtrip zum Server erforderlich.
<code>partitions</code>	Wenn <code>type</code> <code>partitions-list</code> ist, handelt es sich um ein Array von Zeichenfolgen, die die Namen der parallel zu ladenden Partitionen angeben.	Die Namen der Partitionen, die parallel geladen werden.

Parameter	Mögliche Werte	Beschreibung
<code>subpartitions</code>	(Nur Oracle-Endpunkte) Wenn <code>type partitions-list</code> ist, handelt es sich um ein Array von Zeichenfolgen, die die Namen der parallel zu ladenden Unterpartitionen angeben.	Die Namen der Unterpartitionen, die parallel geladen werden.
<code>columns</code>	Wenn <code>type ranges</code> ist, handelt es sich um ein Array von Zeichenfolgen, die die Ansicht der bereichsbasierten Tabellen-, Ansichts- oder Sammlungssegmente identifiziert, die parallel geladen werden sollen.	Die Namen der Spalten, die bereichsbasierte Tabellen-, Ansichts- oder Sammlungssegmente angeben, die parallel geladen werden sollen.
<code>boundaries</code>	Wenn <code>type ranges</code> ist, handelt es sich um ein Array mit Spalte-Wert-Arrays. Jedes Spalte-Wert-Array enthält Spaltenwerte in der von <code>columns</code> angegebenen Menge und Reihenfolge. Ein Spalte-Wert-Array gibt die obere Grenze eines Tabellen-, Ansichts- oder Sammlungssegments an. Jedes zusätzliche Spalte-Wert-Array fügt die obere Grenze für ein zusätzliches Tabellen-, Ansichts- oder Sammlungssegment hinzu. Alle derartigen bereichsbasierten Tabellen-, Ansichts- oder Sammlungssegmente werden parallel geladen.	Spaltenwerte, die bereichsbasierte Tabellen-, Ansichts- oder Sammlungspartitionen identifizieren, die parallel geladen werden sollen.

Parameter	Mögliche Werte	Beschreibung
lob-settings	<p>Ein Objekt mit den folgenden Parametern:</p> <ul style="list-style-type: none">• <code>mode</code> – Gibt den Migration sbearbeitungs-Modus für LOBs an.• <code>bulk-max-size</code> – Gibt die maximale Größe der LOBs abhängig von der <code>mode</code>-Einstellung an.	<p>Ein Wert, der die LOB-Verarbeitung für die von der Option <code>object-locator</code> identifizierte Tabelle oder Ansicht angibt. Die angegebene LOB-Verarbeitung überschreibt alle Aufgaben-LOB-Einstellungen nur für diese Tabelle oder Ansicht. Weitere Informationen zum Verwendung der LOB-Einstellungsparameter finden Sie unter Angeben von LOB-Einstellungen für eine ausgewählte Tabelle oder Ansicht.</p>

Parameter	Mögliche Werte	Beschreibung
mode	<p>Gibt die Migrationsverarbeitung für LOBs in der angegebenen Tabelle oder Ansicht unter Verwendung der folgenden Werte an:</p> <ul style="list-style-type: none">• <code>limited</code> – (Standard) Dieser Wert legt die Migration auf den eingeschränkten LOB-Modus fest, wobei alle LOBs gemeinsam mit allen anderen Spaltentypen in der Tabelle oder Ansicht migriert werden. Verwenden Sie diesen Wert, wenn Sie zumeist kleine LOBs (100 MB oder weniger) replizieren. Geben Sie außerdem einen <code>bulk-max-size</code> -Wert an (Null ist ungültig). Alle migrierte n LOBs über <code>bulk-max-size</code> abgeschnitten, um der von Ihnen eingestellten Größe zu entsprechen.• <code>unlimited</code> – Dieser Wert legt die Migration auf den vollständigen LOB-Modus fest. Verwenden Sie diesen Wert, wenn alle oder die meisten LOBs, die Sie replizieren möchten, größer als 1 GB sind. Wenn Sie den <code>bulk-max-size</code> -Wert Null angeben, werden alle LOBs im Standard-LOB-Modus (vollständig) migriert. In dieser Form des <code>unlimited</code> -Modus werden alle LOBs getrennt von anderen Spaltentypen mithilfe einer Suche aus	Der Mechanismus zum Migrieren von LOBs.

Parameter	Mögliche Werte	Beschreibung
	<p>der Quelltable oder -ansicht migriert. Wenn Sie einen <code>bulk-max-size</code> -Wert größer als Null festlegen, werden alle LOBs im Kombinations-LOB-Modus (vollständig) migriert. In dieser Form des <code>unlimited</code> -Modus werden LOBs über <code>bulk-max-size</code> unter Verwendung einer Quelltabellen- oder Quellansichtssuche migriert, ähnlich wie im Standard-LOB-Modus (vollständig). Andernfalls werden LOBs bis einschließlich dieser Größe inline migriert, ähnlich dem eingeschränkten LOB-Modus. Kein LOB wird im <code>unlimited</code> -Modus abgeschnitten, unabhängig von der Form, die Sie verwenden.</p> <ul style="list-style-type: none"> • <code>none</code> – Alle Tabellen oder Ansichten-LOBs werden entsprechend den LOB-Einstellungen der Aufgabe migriert. <p>Weitere Informationen über die LOB-Einstellungen für Aufgaben finden Sie unter Ziel-Metadaten-Aufgabeneinstellungen.</p> <p>Weitere Informationen zur Migration von LOBs und der Angabe dieser LOB-Einstellungen für Aufgaben finden Sie unter Einstellung der LOB-Unterstützung</p>	

Parameter	Mögliche Werte	Beschreibung
	für Quelldatenbanken in einer Aufgabe AWS DMS.	
<code>bulk-max-size</code>	Der Effekt dieses Werts hängt vom mode ab.	Die maximale Größe von LOBs in Kilobyte-Schritten . Geben Sie diese Option nur an, wenn Sie kleine LOBs replizieren müssen oder wenn der Zielpunkt keine unbegrenzte LOB-Größe unterstützt.

Platzhalter in Tabelleneinstellungen sind eingeschränkt

Die Verwendung des Prozent-Platzhalters ("%") in-"table-settings" Regeln wird für Quelldatenbanken nicht unterstützt, wie im Folgenden dargestellt.

```
{
  "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
  }
}
```

Wenn Sie "%" in den "table-settings" Regeln wie in der Abbildung gezeigt verwenden, wird die folgende Ausnahme AWS DMS zurückgegeben.

```
Error in mapping rules. Rule with ruleId = x failed validation. Exact
```

```
schema and table name required when using table settings rule.
```

AWS empfiehlt außerdem, nicht eine große Anzahl großer Sammlungen mithilfe einer einzigen Aufgabe mit `parallel-load` zu laden. Beachten Sie, dass von AWS DMS der Ressourcenkonflikt sowie die Anzahl der parallel geladenen Segmente um den Wert des Parameters `MaxFullLoadSubTasks` für die Aufgabeneinstellungen begrenzt wird, wobei der Höchstwert 49 beträgt.

Geben Sie stattdessen alle Sammlungen für Ihre Quelldatenbank für die größten Sammlungen an, indem Sie sie "schema-name" und "table-name" einzeln angeben. Skalieren Sie Ihre Migration außerdem korrekt. Führen Sie beispielsweise mehrere Aufgaben auf einer ausreichenden Anzahl von Replikations-Instances aus, um eine große Anzahl großer Sammlungen in Ihrer Datenbank zu verwalten.

Verwendung des parallelen Ladens für ausgewählte Tabellen und Ansichten

Um die Migration zu beschleunigen und effizienter zu gestalten, können Sie für ausgewählte Tabellen, Ansichten und Sammlungen die parallele Ladung verwenden. Mit anderen Worten: Sie können eine einzelne segmentierte Tabelle, Ansicht oder Sammlung mit mehreren Threads parallel migrieren. Zu diesem Zweck AWS DMS teilt eine Vollladeaufgabe in Threads auf, wobei jedes Tabellensegment einem eigenen Thread zugewiesen wird.

Mit diesem parallelen Ladevorgang können mehrere Threads mehrere Tabellen, Ansichten und Sammlungen vom Quellendpunkt entladen. Anschließend können Sie mehrere Threads migrieren und die gleichen Tabellen, Ansichten und Sammlungen parallel auf den Zielendpunkt laden. Bei einigen Datenbank-Engines können Sie die Tabellen und Ansichten nach vorhandenen Partitionen oder Unterpartitionen aufteilen. Für andere Datenbank-Engines können Sie Sammlungen AWS DMS automatisch nach bestimmten Parametern segmentieren lassen (Autosegmentierung). Andernfalls können Sie jede Tabelle, Ansicht oder Sammlung nach den von Ihnen angegebenen Spaltenwertbereichen segmentieren.

Paralleles Laden wird für die folgenden Quellendpunkte unterstützt:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2

- SAP Adaptive Server Enterprise (ASE)
- MongoDB (unterstützt nur die Optionen Autosegmentierung und Bereichssegmentierung einer parallelen Volllast)
- Amazon DocumentDB (unterstützt nur die Optionen Autosegmentierung und Bereichssegmentierung einer parallel Volllast)
- Amazon DynamoDB

AWS DMS Unterstützt für MongoDB- und Amazon DocumentDB DocumentDB-Endpoints die folgenden Datentypen für Spalten, die Partitionsschlüssel für die Bereichssegmentierungsoption einer parallel Volllast sind.

- Double
- String
- ObjectId
- 32-Bit-Ganzzahl
- 64-Bit-Ganzzahl

Das parallele Laden zur Verwendung mit Tabelleneinstellungsregeln wird für die folgenden Zielendpunkte unterstützt:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Amazon S3
- SAP Adaptive Server Enterprise (ASE)
- Amazon Redshift
- MongoDB (unterstützt nur die Optionen Autosegmentierung und Bereichssegmentierung einer parallelen Volllast)
- Amazon DocumentDB (unterstützt nur die Optionen Autosegmentierung und Bereichssegmentierung einer parallel Volllast)

Verwenden Sie die `MaxFullLoadSubTasks`-Aufgabeneinstellung, um die maximale Anzahl an Tabellen und Ansichten anzugeben, die parallel geladen werden.

Um die maximale Anzahl von Threads pro Tabelle oder Ansicht für die unterstützten Ziele einer Aufgabe zum parallelen Laden anzugeben, definieren Sie mehr Segmente mithilfe von Spaltenwertgrenzen.

Important

`MaxFullLoadSubTasks` steuert die Anzahl der Tabellen oder Tabellensegmente, die parallel geladen werden sollen. `ParallelLoadThreads` steuert die Anzahl der Threads, die von einer Migrationsaufgabe verwendet werden, um die Lasten parallel auszuführen. Diese Einstellungen sind multiplikativ. Daher ist die Gesamtzahl der Threads, die während einer Vollast-Aufgabe verwendet werden, ungefähr das Ergebnis des Werts `ParallelLoadThreads` multipliziert mit dem Wert von `MaxFullLoadSubTasks` (`ParallelLoadThreads * MaxFullLoadSubtasks`).

Wenn Sie Aufgaben mit einer hohen Anzahl von Vollast-Unteraufgaben und einer hohen Anzahl von parallelen Lade-Threads erstellen, kann Ihre Aufgabe zu viel Speicher verbrauchen und fehlschlagen.

Um die maximale Anzahl von Threads pro Tabelle für Amazon DynamoDB-, Amazon-Kinesis-Data-Streams-, Apache-Kafka- oder Amazon-Elasticsearch-Ziele anzugeben, verwenden Sie die Zielmetadaten-Aufgabeneinstellung `ParallelLoadThreads`.

Sie geben die Puffergröße für eine Aufgabe zum parallelen Laden bei Verwendung von `ParallelLoadThreads` mithilfe der Zielmetadaten-Aufgabeneinstellung `ParallelLoadBufferSize` an.

Die Verfügbarkeit von `ParallelLoadThreads` und `ParallelLoadBufferSize` sowie deren Einstellungen hängen vom Ziel-Endpoint ab.

Weitere Informationen zu den Einstellungen `ParallelLoadThreads` und `ParallelLoadBufferSize` finden Sie unter [Ziel-Metadaten-Aufgabeneinstellungen](#). Weitere Informationen zur `MaxFullLoadSubTasks`-Einstellung finden Sie unter [Aufgabeneinstellungen für vollständiges Laden](#). Weitere Informationen zu den Zielendpunkten finden Sie unter den entsprechenden Themen.

Wenn Sie paralleles Laden verwenden möchten, können Sie mit der `parallel-load`-Option eine Tabellenzuweisungsregel des Typs `table-settings` angeben. In der `table-settings`-Regel können Sie die Segmentierungskriterien für eine einzelne Tabelle, Ansicht oder Sammlung, die parallel geladen werden soll, angeben. Um dies zu tun, setzen Sie den `type`-Parameter der `parallel-load`-Option auf eine der verschiedenen Optionen.

Wie Sie dies tun, hängt davon ab, wie Sie die Tabelle, Ansicht oder Sammlung für das parallele Laden segmentieren möchten.

- Nach Partitionen (oder Segmenten) – Alle vorhandenen Tabellen- oder Ansichtspartitionen (oder Segmente) werden mithilfe des `partitions-auto`-Typs geladen. Oder Laden nur ausgewählter Partitionen mithilfe des `partitions-list`-Typs mit einem angegebenen Partitionen-Array.

Nur für MongoDB- und Amazon DocumentDB DocumentDB-Endpoints laden Sie alle oder bestimmte Sammlungen nach Segmenten, die AWS DMS automatisch auch unter Verwendung des `partitions-auto` Typs und zusätzlicher optionaler Parameter berechnet werden. `table-settings`

- (Nur Oracle-Endpunkte) Nach Unterpartitionen – Laden alle vorhandenen Tabellen- oder Ansichtsunterpartitionen mithilfe des `subpartitions-auto`-Typs. Oder Laden nur ausgewählter Unterpartitionen mithilfe des `partitions-list`-Typs mit einem `subpartitions`-Array.
- Nach Segmenten, die Sie definieren – Laden von Ihnen definierter Tabellen-, Ansichts- oder Sammlungssegmente mithilfe von Spalte-Wert-Grenzen. Um dies zu tun, verwenden Sie den `ranges`-Typ mit angegebenen `columns`- und `boundaries`-Arrays.

Note

PostgreSQL-Endpoints unterstützen nur diese Art paralleler Ladevorgänge. MongoDB und Amazon DocumentDB als Quellendpunkte unterstützen sowohl diesen Bereichsegmentierungstyp als auch den Autosegmentierungstyp einer parallelen Volllast (`partitions-auto`).

Um weitere Tabellen, Ansichten oder Sammlungen zum parallelen Laden zu identifizieren, geben Sie zusätzliche `table-settings`-Objekte mit `parallel-load`-Optionen an.

Die folgenden Verfahren beschreiben das Codieren von JSON für jeden parallelen Ladetyp, vom einfachsten zum kompliziertesten.

So geben Sie alle Tabellen-, Ansichts- oder Sammlungspartitionen oder alle Tabellen- oder Ansichts-Unterpitionen an

- Geben Sie `parallel-load` entweder mit dem `partitions-auto`-Typ oder dem `subpartitions-auto`-Typ an (aber nicht mit beiden).

Jede Tabellen-, Ansichts- oder Sammlungspartition oder -unterpartition wird dann automatisch einem eigenen Thread zugeordnet.

Bei manchen Endpunkten beinhaltet das parallele Laden nur dann Partitionen oder Unterpitionen nur, wenn sie bereits für die Tabelle oder Ansicht definiert sind. Für MongoDB- und Amazon DocumentDB DocumentDB-Quellendpunkte können Sie die Partitionen (oder Segmente) anhand optionaler zusätzlicher Parameter AWS DMS automatisch berechnen lassen. Dies sind beispielsweise `number-of-partitions`, `collection-count-from-metadata`, `max-records-skip-per-page` und `batch-size`.

So geben Sie ausgewählte Tabellen- oder Ansichtspartitionen, -unterpartitionen oder beides an:

1. Geben Sie `parallel-load` mit dem `partitions-list`-Typ an.
2. (Optional) Fügen Sie Partitionen ein, indem Sie ein Array von Partitionsnamen als den Wert für `partitions` angeben.

Jede angegebene Partition wird dann einem eigenen Thread zugeordnet.

 **Important**

Stellen Sie bei Oracle-Endpunkten sicher, dass sich Partitionen und Unterpitionen nicht überlappen, wenn Sie sie für das parallele Laden auswählen. Wenn Sie überlappende Partitionen und Unterpitionen verwenden, um Daten parallel zu laden, werden Einträge dupliziert oder der Vorgang schlägt aufgrund einer Verletzung der Primärschlüssel-Duplikat-Regel fehl.

3. (Optional, nur für Oracle-Endpunkte) Fügen Sie Unterpitionen ein, indem Sie ein Array von Unterpitionsnamen als den Wert für `subpartitions` angeben.

Jede angegebene Unterpition wird dann einem eigenen Thread zugeordnet.

 Note

Paralleles Laden enthält Partitionen oder Unterpartitionen nur, wenn sie bereits für die Tabelle oder Ansicht definiert sind.

Sie können Tabellen- oder Ansichtssegmente als Bereiche von Spaltenwerten angeben. Wenn Sie dies tun, achten Sie auf diese Spalteneigenschaften:

- Die Angabe indizierter Spalten verbessert die Leistung erheblich.
- Sie können bis zu 10 Spalten angeben.
- Sie können keine Spalten verwenden, um Segmentgrenzen mit den folgenden AWS DMS Datentypen zu definieren: DOUBLE, FLOAT, BLOB, CLOB und NCLOB
- Datensätze mit Nullwerten werden nicht repliziert.

So geben Sie Tabellen-, Ansichts- oder Sammlungssegmente als Bereiche von Spaltenwerten an

1. Geben Sie `parallel-load` mit dem `ranges`-Typ an.
2. Definieren Sie eine Begrenzung zwischen Tabellen- oder Ansichtssegmenten, indem Sie ein Array von Tabellenspaltennamen als den Wert für `columns` angeben. Wiederholen Sie diese Schritte für jede Spalte, für die Sie eine Begrenzung zwischen Tabellen- oder Ansichtssegmenten definieren möchten.

Beachten Sie, dass die Reihenfolge der Spalten von Bedeutung ist. Dabei ist die erste Spalte bei der Definition jeder Begrenzung am wichtigsten und die letzte Spalte ist am unwichtigsten. Einzelheiten dazu finden Sie in den folgenden Schritten und Abschnitten.

3. Definieren Sie die Datenbereiche für alle Tabellen- oder Ansichtssegmente durch Angabe eines Begrenzungsarrays als Wert für `boundaries`. Bei einem Begrenzungsarray handelt es sich um ein Array mit Spalte-Wert-Arrays. Führen Sie dazu die folgenden Schritte aus:
 - a. Geben Sie jedes Element eines Spalte-Wert-Arrays als Wert an, der jeder Spalte entspricht. Dabei stellt jedes Spalte-Wert-Array die obere Grenze für jedes Tabellen- oder Ansichtssegment dar, das Sie definieren möchten: Geben Sie jede Spalte in der gleichen Reihenfolge an, in der Sie diese Spalte in dem `columns`-Array angegeben haben.

Geben Sie Werte für DATE-Spalten in dem von der Quelle unterstützten Format an.

- b. Geben Sie jedes Array mit Spaltenwerten der Reihe nach als obere Grenze für jedes Segment vom unteren zum Segment der Tabelle oder Ansicht an. next-to-top Wenn Zeilen oberhalb der von Ihnen angegebenen oberen Grenze vorhanden sind, stellend diese Zeilen die oberen Segment der Tabelle oder Ansicht dar. Damit ist die Anzahl der bereichsbasierten Segmente potentiell eins höher, als die Anzahl der Segmentgrenzen im Grenz-Array. Jedes dieser bereichsbasierten Segmente wird einem eigenen Thread zugeordnet.

Alle Tabellendaten, die nicht Null sind, werden repliziert. Dies gilt auch, wenn Sie nicht für alle Spalten in der Tabelle oder Ansicht Datenbereiche definieren.

Beispiel: Sie definieren drei Spalte-Wert-Arrays für die Spalten, COL1, COL2 und COL3 wie folgt.

COL1	COL2	COL3
10	30	105
20	20	120
100	12	99

Sie haben drei Segmentgrenzen für einen möglichen Gesamtwert von vier Segmenten definiert.

Um die Bereiche der für jedes Segment zu replizierenden Zeilen zu identifizieren, wendet die Replikations-Instance einen Suchalgorithmus auf diese drei Spalten für jedes der vier Segmente an. Die Suche sieht wie folgt aus:

Segment 1

Replizieren Sie alle Zeilen, in denen Folgendes zutrifft: Die ersten Werte mit zwei Spalten sind kleiner oder gleich den entsprechenden Werten für die Obergrenze für Segment 1. Außerdem sind die Werte der dritten Spalte kleiner als der Wert der Obergrenze für Segment 1.

Segment 2

Replizieren Sie alle Zeilen (außer Segment 1-Zeilen), wobei Folgendes zutrifft: Die ersten Werte mit zwei Spalten sind kleiner oder gleich den entsprechenden Werten für die

Obergrenze für Segment 2. Außerdem sind die Werte der dritten Spalte kleiner als der Wert der Obergrenze für Segment 2 .

Segment 3

Replizieren Sie alle Zeilen (außer Segment 2-Zeilen), wobei Folgendes zutrifft: Die ersten Werte mit zwei Spalten sind kleiner oder gleich den entsprechenden Werten für die Obergrenze für Segment 3. Außerdem sind die Werte der dritten Spalte kleiner als der Wert für den oberen Grenzwert für Segment 3 .

Segment 4

Replizieren Sie alle verbleibenden Zeilen (außer den Zeilen Segment 1, 2 und 3).

In diesem Fall erstellt die Replikations-Instance wie folgt eine WHERE-Klausel zum Laden jedes Segment:

Segment 1

```
((COL1 < 10) OR ((COL1 = 10) AND (COL2 < 30)) OR ((COL1 = 10) AND (COL2 = 30) AND (COL3 < 105)))
```

Segment 2

```
NOT ((COL1 < 10) OR ((COL1 = 10) AND (COL2 < 30)) OR ((COL1 = 10) AND (COL2 = 30) AND (COL3 < 105))) AND ((COL1 < 20) OR ((COL1 = 20) AND (COL2 < 20)) OR ((COL1 = 20) AND (COL2 = 20) AND (COL3 < 120)))
```

Segment 3

```
NOT ((COL1 < 20) OR ((COL1 = 20) AND (COL2 < 20)) OR ((COL1 = 20) AND (COL2 = 20) AND (COL3 < 120))) AND ((COL1 < 100) OR ((COL1 = 100) AND (COL2 < 12)) OR ((COL1 = 100) AND (COL2 = 12) AND (COL3 < 99)))
```

Segment 4

```
NOT ((COL1 < 100) OR ((COL1 = 100) AND (COL2 < 12)) OR ((COL1 = 100) AND (COL2 = 12) AND (COL3 < 99)))
```

Angeben von LOB-Einstellungen für eine ausgewählte Tabelle oder Ansicht

Sie können die LOB-Aufgabeneinstellungen für eine oder mehrere Tabellen einrichten, indem Sie eine Tabellenzuordnungsregel des Typs `table-settings` mit der `lob-settings`-Option für ein oder mehrere `table-settings`-Objekte erstellen.

Das Angeben von LOB-Einstellungen für ausgewählte Tabellen oder Ansichten wird für die folgenden Quell-Endpunkte unterstützt:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2, abhängig von den `mode-` und `bulk-max-size`-Einstellungen, wie im Folgenden beschrieben
- SAP Adaptive Server Enterprise (ASE), abhängig von den `mode-` und `bulk-max-size`-Einstellungen, wie im Folgenden beschrieben

Das Angeben von LOB-Einstellungen für ausgewählte Tabellen oder Ansichten wird für die folgenden Zielendpunkte unterstützt:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- SAP ASE, je nach den `mode-` und `bulk-max-size`-Einstellungen, wie im Folgenden beschrieben

Note


Sie können LOB-Datentypen nur mit Tabellen und Ansichten verwenden, die einen Primärschlüssel enthalten.

Wenn Sie LOB-Einstellungen für eine ausgewählte Tabelle oder Ansicht verwenden möchten, erstellen Sie eine Tabellenzuweisungsregel des Typs `table-settings` mit der `lob-settings`-Option. Damit wird die LOB-Behandlung für die Tabelle oder Ansicht angegeben, die von der

`object-locator`-Option identifiziert wird. In der `table-settings`-Regel können Sie mit den folgenden Parametern ein `lob-settings`-Objekt angeben:

- `mode` – Gibt den Mechanismus für die Behandlung der LOB-Migration für die ausgewählte Tabelle oder Ansicht wie folgt an:
 - `limited` – Der eingeschränkte LOB-Standardmodus ist der schnellste und effizienteste Modus. Verwenden Sie diesen Modus nur dann, wenn alle Ihre LOBs klein (unter 100 MB) sind oder der Ziel-Endpoint keine unbegrenzte LOB-Größe unterstützt. Weiterhin gilt: Wenn Sie `limited` verwenden, müssen alle LOBs in dem Größenbereich liegen, den Sie für `bulk-max-size` eingerichtet haben.


In diesem Modus für eine vollständige Ladungsaufgabe migriert die Replikations-Instance alle LOBs inline zusammen mit anderen Spaltendatentypen als Teil des Haupttabellen- oder -ansichtsspeichers. Die Instance kürzt allerdings jedes LOB, das größer als Ihr `bulk-max-size`-Wert ist, auf die angegebene Größe. Für eine Change Data Capture (CDC)-Ladeaufgabe migriert die Instance alle LOBs mithilfe einer Quelltabellensuche unabhängig von der LOB-Größe als vollständigen LOB-Standardmodus (siehe unten). Dies geschieht unabhängig von der LOB-Größe.

 Note

Sie können Ansichten nur für Full-Load-Aufgaben migrieren.

- `unlimited` – Der Migrationsmechanismus für den vollständigen LOB-Modus hängt von dem Wert ab, den Sie wie folgt für `bulk-max-size` festgelegt haben:
 - Vollständiger LOB-Standardmodus – Wenn Sie `bulk-max-size` auf Null festlegen, migriert die Replikations-Instance alle LOBs mithilfe des vollständigen LOB-Standardmodus. Dieser Modus erfordert eine Suche in der Quelltable oder -ansicht, um jedes LOB unabhängig von seiner Größe zu migrieren. Dies führt in der Regel zu einer sehr viel langsameren Migration als im eingeschränkten LOB-Modus. Verwenden Sie diesen Modus nur, wenn alle oder die meisten Ihrer LOBs groß sind (1 GB oder größer).
 - Vollständiger LOB-Kombinationsmodus – Wenn Sie `bulk-max-size` auf einen Wert ungleich Null festlegen, verwendet dieser vollständige LOB-Modus eine Kombination aus eingeschränktem LOB-Modus und vollständigem LOB-Standardmodus. Das heißt, wenn eine LOB-Größe innerhalb Ihres `bulk-max-size`-Werts liegt, migriert die Instance das LOB inline wie im eingeschränkten LOB-Modus. Wenn die LOB-Größe diesen Wert übersteigt, migriert die Instance das LOB mithilfe einer Quelltabellen- oder Quellansichtssuche wie im

vollständigen LOB-Standardmodus. Für eine Change Data Capture (CDC)-Ladeaufgabe migriert die Instance alle LOBs mithilfe einer Quelltabellensuche unabhängig von der LOB-Größe als vollständigen LOB-Standardmodus (siehe unten). Dies geschieht unabhängig von der LOB-Größe.

 Note

Sie können Ansichten nur für Full-Load-Aufgaben migrieren.

Dieser Modus resultiert in einem Kompromiss bei der Migrationsgeschwindigkeit. Sie liegt zwischen dem schnelleren eingeschränkten LOB-Modus und dem langsameren vollständigen LOB-Standardmodus. Verwenden Sie diesen Modus nur bei einer Mischung aus kleinen und großen LOBs, wobei die meisten LOBs klein sind.

Dieser kombinierte vollständige LOB-Modus ist nur verfügbar für die folgenden Endpunkte:

- IBM Db2 als Quelle
- SAP ASE als Quelle oder Ziel

Unabhängig davon, welchen Mechanismus Sie für den `unlimited`-Modus angeben, migriert die Instance alle LOBs vollständig und ohne Kürzung.

- `none` – Die Replikations-Instance migriert LOBs in der ausgewählten Tabelle oder Ansicht anhand Ihrer LOB-Einstellungen für Aufgaben. Sie können diese Option verwenden, um Migrationsergebnisse mit und ohne LOB-Einstellungen besonders für die ausgewählte Tabelle oder Ansicht zu vergleichen.

Wenn die angegebene Tabelle oder Ansicht LOBs in die Replikation einschließt, können Sie die `BatchApplyEnabled`-Aufgabeneinstellung nur bei Verwendung des `limited`-LOB-Modus auf `true` setzen.

In einigen Fällen könnten Sie `BatchApplyEnabled` auf `true` und `BatchApplyPreserveTransaction` auf `false` setzen. In diesen Fällen setzt die Instance `BatchApplyPreserveTransaction` auf `true`, wenn die Tabelle oder Ansicht LOBs besitzt und die Quell- und Ziel-Endpunkte Oracle sind.

- `bulk-max-size` – Legen Sie diesen Wert in Kilobyte auf Null oder ungleich Null fest. Dies ist, wie bereits für vorherige Elemente beschrieben, vom `mode` abhängig. Im `limited`-Modus müssen Sie für diesen Parameter einen Wert ungleich Null setzen.

Die Instance konvertiert LOBs in das Binärformat. Aus diesem Grund multiplizieren Sie die Größe der größten LOBs, die Sie replizieren müssen, mit drei. Beispiel: Wenn Ihr größtes LOB 2 MB groß ist, legen Sie `bulk-max-size` auf 6.000 (6 MB) fest.

Beispiele für Tabelleneinstellungen

Im Folgenden finden Sie einige Beispiele, die die Verwendung der Tabelleneinstellungen illustrieren.

Example Laden einer nach Partitionen segmentierten Tabelle

Im folgenden Beispiel wird eine SALES-Tabelle effizienter in Ihre Quelle geladen, indem basierend auf allen Partitionen sie parallel geladen wird.

```
{
  "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 Laden einer nach Unterpartitionen segmentierten Tabelle

Im folgenden Beispiel wird eine SALES-Tabelle effizienter in Ihre Oracle-Quelle geladen, indem sie auf allen Unterpartitionen parallel geladen wird.

```
{
  "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 Laden einer segmentierten Tabelle entsprechend einer Partitionsliste

Im folgenden Beispiel wird eine SALES-Tabelle in Ihre Quelle geladen, indem sie entsprechend einer bestimmten Partitionsliste parallel geladen wird. Hier werden die angegebenen Partitionen nach Werten benannt, beginnend mit Teilen des englischen Alphabets, z. B. ABCD, EFGH usw.

```
{
  "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 Laden einer segmentierten Oracle-Tabelle entsprechend einer ausgewählten Liste mit Partitionen und Unterpartitionen

Im folgenden Beispiel wird eine SALES-Tabelle in Ihre Oracle-Quelle geladen, indem sie entsprechend einer ausgewählten Liste mit Partitionen und Unterpartitionen parallel geladen wird. Hier werden die angegebenen Partitionen nach Werten benannt, beginnend mit Teilen des englischen Alphabets, z. B. ABCD, EFGH usw. Die angegebenen Unterpartitionen sind nach mit Zahlen beginnenden Werten benannt, z. B. 01234 und 56789.

```

{
    "rules": [{
        "rule-type": "selection",
        "rule-id": "1",
        "rule-name": "1",
        "object-locator": {

```

```

        "schema-name": "%",
        "table-name": "%"
    },
    "rule-action": "include"
},
{
    "rule-type": "table-settings",
    "rule-id": "2",
    "rule-name": "2",
    "object-locator": {
        "schema-name": "HR",
        "table-name": "SALES"
    },
    "parallel-load": {
        "type": "partitions-list",
        "partitions": [
            "ABCD",
            "EFGH",
            "IJKL",
            "MNOP",
            "QRST",
            "UVWXYZ"
        ],
        "subpartitions": [
            "01234",
            "56789"
        ]
    }
}
]
}

```

Example Laden einer durch Spaltenwertbereiche segmentierten Tabelle

Im folgenden Beispiel wird eine SALES-Tabelle in Ihre Quelle geladen, indem sie nach den durch Bereich der SALES_NO- und REGION-Spaltenwerte angegebenen Segmente parallel geladen wird.

```

{
    "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"
            ]
        ]
    }
}
]
```

Hier werden zwei Spalten für die Segmentbereiche mit den Namen SALES_NO und REGION angegeben. Es werden zwei Grenzen mit den beiden Spaltenwertsätzen ["1000", "NORTH"] und ["3000", "WEST"] angegeben.

Diese beiden Grenzen identifizieren somit die folgenden drei Tabellensegmente, die parallel geladen werden:

Segment 1

Zeilen mit SALES_NO kleiner oder gleich 1.000 und mit REGION kleiner als "NORTH" sind. Mit anderen Worten: Verkaufszahlen bis zu 1.000 in der EAST-Region.

Segment 2

Andere Zeilen als Segment 1 mit SALES_NO kleiner oder gleich 3.000 und REGION kleiner als "WEST". Mit anderen Worten: Verkaufszahlen zwischen 1.000 und 3.000 in den Regionen „NORTH“ und „SOUTH“.

Segment 3

Alle verbleibenden Zeilen außer Segment 1 und Segment 2. Mit anderen Worten: Verkaufszahlen über 3.000 in der WEST-Region.

Example Laden von zwei Tabellen: Eine segmentiert nach Bereichen, die andere nach Partitionen

Das folgende Beispiel lädt eine SALES-Tabelle parallel nach Segmentgrenzen, die Sie identifizieren. Es lädt außerdem eine ORDERS-Tabelle parallel nach allen ihren Partitionen, wie in früheren Beispielen.

```
{
  "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 Laden einer Tabelle mit LOBs mithilfe des eingeschränkten LOB-Modus

Im folgenden Beispiel wird eine ITEMS-Tabelle, einschließlich LOBs in Ihre Quelle mithilfe des eingeschränkten LOB-Modus (Standard) mit einer maximalen nicht gekürzten Größe von 100 MB geladen. Alle LOBs, die größer als dieser Wert sind, werden auf 100 MB gekürzt. Alle LOBs werden inline mit allen anderen Spaltendatentypen geladenen.

```

{
  "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 Laden einer Tabelle mit LOBs mithilfe des vollständigen LOB-Standardmodus

Im folgenden Beispiel wird eine ITEMS-Tabelle einschließlich aller ihrer LOBs ohne Kürzung in Ihre Quelle geladen. Dies erfolgt über den vollständigen LOB-Standardmodus. Es werden alle LOBs unabhängig ihrer Größe getrennt von anderen Datentypen geladen, die eine Suche für jedes LOB in der Quelltable verwenden.

```

{
  "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 Laden einer Tabelle mit LOBs mithilfe des vollständigen LOB-Kombinationsmodus

Im folgenden Beispiel wird eine ITEMS-Tabelle einschließlich aller ihrer LOBs ohne Kürzung in Ihrer Quelle geladen. Dies erfolgt über den vollständigen LOB-Kombinationsmodus. Alle LOBs mit einer Maximalgröße von 100 MB werden zusammen mit anderen Datentypen inline geladen, wie im eingeschränkten LOB-Modus. Alle LOBs mit einer Größe von über 100 MB werden getrennt von anderen Datentypen geladen. Dieser separate Ladevorgang verwendet eine Suche für jedes dieser LOB in der Quelltable, wie im standardmäßigen vollständigen LOB-Modus.

```

{
  "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 Laden einer Tabelle mit LOBs über die LOB-Einstellungen für Aufgaben

Im folgenden Beispiel wird eine ITEMS-Tabelle mithilfe der LOB-Einstellungen für Aufgaben einschließlich aller LOBs in Ihre Quelle geladen. Die `bulk-max-size`-Einstellung von 100 MB wird ignoriert und nur für ein schnelles Zurücksetzen auf den `limited`- oder `unlimited`-Modus beibehalten.

```

{
  "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"
    }
  }
]
}

```


Verwenden von Quellfiltern

Sie können Quellfilter verwenden, um Anzahl und Art der Datensätze zu beschränken, die von Ihrer Quelle auf Ihr Ziel übertragen werden. So können Sie beispielsweise festlegen, dass nur die am Hauptsitz ansässigen Angestellten in die Zieldatenbank übertragen werden. Filter sind Teil einer Auswahlregel. Sie wenden Filter auf eine Datenspalte an.

Quellfilter müssen diese Einschränkungen befolgen:

- Eine Auswahlregel kann keine Filter oder einen oder mehrere Filter haben.
- Jeder Filter kann eine oder mehrere Filterbedingungen haben.
- Wenn mehr als ein Filter verwendet wird, wird die Liste der Filter kombiniert, als ob die Filter mit einem UND-Operator verbunden wären.
- Wenn mehr als eine Filterbedingung in einem einzigen Filter verwendet wird, wird die Liste der Filterbedingungen so kombiniert, als ob die Filterbedingungen mit einem ODER-Operator verbunden wären.
- Filter werden nur angewendet, wenn `rule-action = 'include'`.
- Filter erfordern einen Spaltennamen und eine Liste von Filterbedingungen. Filterbedingungen müssen einen Filteroperator enthalten, der je nach Operator einem Wert, zwei Werten oder keinem Wert zugeordnet ist.
- Bei Spaltennamen, Tabellennamen, Ansichtsnamen und Schemanamen wird zwischen Groß- und Kleinschreibung unterschieden. Oracle und Db2 sollten immer GROSSBUCHSTABEN verwenden.
- Filter unterstützen nur Tabellen mit exakten Namen. Filter unterstützen keine Platzhalter.

Die folgenden Einschränkungen gelten für die Verwendung von Quellfiltern:

- Filter berechnen keine Spalten von Rechts-nach-links-Sprachen.
- Wenden Sie keine Filter auf LOB-Spalten an.
- Wenden Sie Filter nur auf unveränderliche Spalten an, die nach der Erstellung nicht aktualisiert werden. Wenn Quellfilter auf veränderliche Spalten angewendet werden, die nach der Erstellung aktualisiert werden können, kann es zu einem unerwünschten Verhalten kommen.

Ein Filter zum Ausschließen oder Einschließen bestimmter Zeilen in einer Spalte schließt die angegebenen Zeilen immer aus bzw. ein, selbst wenn die Zeilen später geändert werden.

Angenommen, Sie schließen die Zeilen 1 bis 10 in Spalte A aus oder ein und später werden diese

Zeilen geändert und werden zu den Zeilen 11 bis 20. In diesem Fall werden sie weiterhin aus- bzw. eingeschlossen, auch wenn die Daten nicht mehr identisch sind.

Angenommen, eine Zeile außerhalb des Filters wird später aktualisiert (oder aktualisiert und gelöscht) und sollte dann wie vom Filter definiert aus- oder eingeschlossen werden. In diesem Fall erfolgt die Replikation am Ziel.

Erstellen von Quellfilterregeln in JSON

Sie können Quellfilter erstellen, indem Sie den JSON-Parameter `filters` einer Auswahlregel verwenden. Der `filters`-Parameter gibt ein Array von einem oder mehreren JSON-Objekten an. Jedes Objekt verfügt über Parameter, die den Quell-Filtertyp, Spaltennamen und die Filterbedingungen festlegen. Diese Filterbedingungen umfassen einen oder mehrere Filteroperatoren und Filterwerte.

Die folgende Tabelle zeigt die Parameter, die Quellfilterung in einem `filters`-Objekt festlegen.

Parameter	Wert
<code>filter-type</code>	<code>source</code>
<code>column-name</code>	Ein Parameter mit dem Namen der Quellspalte, auf die der Filter angewendet werden soll. Der Name berücksichtigt Groß- und Kleinschreibung.
<code>filter-conditions</code>	Ein Array von einem oder mehreren Objekten, die einen Parameter <code>filter-operator</code> und null oder mehr zugehörige Wertparameter enthalten, je nach Wert für <code>filter-operator</code> .
<code>filter-operator</code>	Ein Parameter mit einem der folgenden Werte: <ul style="list-style-type: none"> • <code>lte</code> – kleiner oder gleich einem Wert • <code>ste</code> – kleiner oder gleich einem Wert (<code>lte</code>-Alias) • <code>gte</code> – größer oder gleich einem Wert • <code>eq</code> – gleich einem Wert • <code>noteq</code> – nicht gleich einem Wert • <code>between</code> – gleich oder zwischen zwei Werten

Parameter	Wert
	<ul style="list-style-type: none"> • <code>notbetween</code> – nicht gleich oder zwischen zwei Werten • <code>null</code> – NULL-Werte • <code>notnull</code> – keine NULL-Werte
<p><code>value</code> oder</p> <p><code>start-value</code> und <code>end-value</code> oder</p> <p>keine Werte</p>	<p>Null oder mehr Wertparameter, die mit <code>filter-operator</code> verknüpft sind:</p> <ul style="list-style-type: none"> • Falls für <code>filter-operator</code> die Option <code>lte</code>, <code>ste</code>, <code>gte</code>, <code>eq</code> oder <code>noteq</code> festgelegt ist, verwenden Sie <code>value</code>, um einen Wertparameter anzugeben. • Falls für <code>filter-operator</code> die Option <code>between</code> oder <code>notbetween</code> festgelegt ist, verwenden Sie <code>start-value</code> und <code>end-value</code>, um zwei Wertparameter anzugeben. • Falls für <code>filter-operator</code> die Option <code>null</code> oder <code>notnull</code> festgelegt ist, geben Sie keine Wertparameter an.

Die folgenden Beispiele zeigen einige gängige Möglichkeiten zur Nutzung von Quellfiltern.

Example Einzelner Filter

Der folgende Filter repliziert alle Angestellten mit dem Wert `empid >= 100` in die Zieldatenbank.

```
{
  "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 Mehrere Filteroperatoren

Die folgenden Filter wenden mehrere Filteroperatoren auf eine einzelne Datenspalte an. Der folgende Filter repliziert alle Angestellten mit dem Wert (`empid <= 10`) ODER (`empid is between 50 and 75`) ODER (`empid >= 100`) in die Zieldatenbank.

```

{
  "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 Mehrere Filter

Die folgenden Filter wenden mehrere Filter auf zwei Spalten in einer Tabelle an. Der Filter repliziert alle Angestellten mit dem Wert (`empid <= 100`) UND (`dept = tech`) in die Zieldatenbank.

```
{
  "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 Filtern von NULL-Werten

Der folgende Filter zeigt, wie nach leeren Werten gefiltert wird. Er repliziert alle Angestellten mit dem Wert `dept = NULL` in die Zieldatenbank.

```
{
  "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 Filtern unter Verwendung von NOT-Operatoren

Einige der Operatoren können in negativer Form verwendet werden. Der folgende Filter repliziert alle Angestellten mit dem Wert (`empid is < 50`) OR (`empid is > 75`) in die Zieldatenbank.

```

{
  "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 Verwenden von kombinierten Filteroperatoren

Ab AWS DMS Version 3.5.0 können Sie inklusive Operatoren und negative Operatoren kombinieren.

Der folgende Filter repliziert alle Angestellten mit dem Wert (empid != 50) AND (dept is not NULL) in die Zieldatenbank.

```

{
  "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"
      }]
    }]
  }]
}

```

Beachten Sie bei Verwendung von null mit anderen Filteroperatoren Folgendes:

- Wenn Sie inklusive und negative Filterbedingungen sowie null-Filterbedingungen gemeinsam innerhalb desselben Filters verwenden, werden Datensätze mit NULL-Werten nicht repliziert.
- Wenn Sie negative Filterbedingungen und null-Filterbedingungen ohne inklusive Filterbedingungen innerhalb desselben Filters verwenden, werden keine Daten repliziert.
- Bei Verwendung negativer Filterbedingungen ohne explizite Festlegung einer null-Filterbedingung werden Datensätze mit NULL-Werten nicht repliziert.

Filtern nach Zeit und Datum

Wenn Sie Daten zum Importieren auswählen, können Sie ein Datum oder eine Uhrzeit als Teil Ihrer Filterkriterien bestimmen. AWS DMS verwendet das Datumsformat JJJJ-MM-TT und das Uhrzeitformat JJJJ-MM-TT HH:MM:SS zum Filtern. Die AWS DMS-Vergleichsfunktionen befolgen die SQLite-Konventionen. Weitere Informationen zu SQLite-Datentypen und -Datenvergleichen finden Sie unter [Datatypes In SQLite Version 3](#) in der SQLite-Dokumentation.

Der folgende Filter zeigt, wie Sie nach einem Datum filtern. Er repliziert alle Angestellten mit dem Wert `empstartdate >= January 1, 2002` in die Zieldatenbank.

Example Filter für einzelnes Datum

```
{
  "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"
      }]
    }]
  }]
}
```


}

Aktivieren und Verwenden von Vormigrationsbewertungen für eine Aufgabe

Bei einer Vormigrationsbewertung werden bestimmte Komponenten einer Datenbankmigrationsaufgabe beurteilt, um Probleme zu identifizieren, die eine erwartungsgemäße Ausführung einer Migrationsaufgabe verhindern könnten. Diese Bewertung gibt Ihnen die Möglichkeit, Probleme zu identifizieren und zu beheben, bevor Sie eine neue oder geänderte Aufgabe ausführen. Auf diese Weise können Sie Verzögerungen aufgrund fehlender Aufgaben oder bekannter Einschränkungen vermeiden.

AWS DMS bietet Zugriff auf zwei verschiedene Optionen für Bewertungen vor der Migration:

- **Bewertung des Datentyps:** Ein älterer Bericht, der einen begrenzten Umfang an Bewertungen bietet.
- **Bewertungslauf vor der Migration:** Enthält verschiedene Arten von Einzelprüfungen, einschließlich der Ergebnisse der Datentypbewertung.

Note

Wenn Sie sich für einen Bewertungslauf vor der Migration entscheiden, müssen Sie eine Datentypbewertung nicht separat auswählen.

Diese Optionen werden in den folgenden Themen beschrieben:

- [Angeben, Starten und Anzeigen von Vormigrationsbewertungsläufen](#): Ein Testlauf vor der Migration (empfohlen) spezifiziert eine oder mehrere Einzelprüfungen, die auf der Grundlage einer neuen oder vorhandenen Konfiguration der Migrationsaufgabe ausgeführt werden sollen. Bei jeder einzelnen Bewertung wird ein bestimmtes Element einer unterstützten Quell- und/oder Zieldatenbank anhand von Kriterien wie Migrationstyp, unterstützten Objekten, Indexkonfiguration und anderen Aufgabeneinstellungen bewertet, z. B. Tabellenzuordnungen, die die zu migrierenden Schemas und Tabellen identifizieren.

Bei einer individuellen Bewertung kann beispielsweise bewertet werden, welche Quelldatentypen oder Primärschlüsselformate migriert werden können oder nicht, möglicherweise basierend auf

der Engine-Version. AWS DMS Sie können die Ergebnisse des letzten Bewertungslaufs starten und anzeigen sowie die Ergebnisse aller vorherigen Bewertungsläufe für eine Aufgabe anzeigen, indem Sie entweder die AWS DMS Management Console oder die SDKs für den AWS CLI Zugriff auf die AWS DMS API verwenden. Sie können auch die Ergebnisse früherer Bewertungsläufe für eine Aufgabe in einem Amazon S3 S3-Bucket anzeigen, den Sie AWS DMS zum Speichern dieser Ergebnisse ausgewählt haben.

Note

Die Anzahl und die Arten der verfügbaren individuellen Bewertungen können im Laufe der Zeit zunehmen. Weitere Informationen zu regelmäßigen Aktualisierungen finden Sie unter [Angeben von individuellen Bewertungen](#).

- [Datentypbewertungen starten und anzeigen \(Legacy\)](#): Eine Datentypbewertung (Legacy) gibt die Ergebnisse einer einzelnen Art von Bewertung vor der Migration in einer einzigen JSON-Struktur zurück: die Datentypen, die in einer unterstützten relationalen Quelldatenbank-Instance möglicherweise nicht korrekt migriert wurden. Dieser Bericht gibt die Ergebnisse für alle problematischen Datentypen in allen Schemas und Tabellen in der Quelldatenbank zurück, die für die Migration ausgewählt wurden.

Schaffung der Voraussetzungen für Bewertungen vor der Migration

In diesem Abschnitt werden die Amazon S3- und IAM-Ressourcen beschrieben, die Sie für die Erstellung einer Bewertung vor der Migration benötigen.

Erstellen eines S3-Buckets

AWS DMS speichert Bewertungsberichte vor der Migration in einem S3-Bucket. Gehen Sie wie folgt vor, um den S3-Bucket zu erstellen:

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Bucket erstellen aus.
3. *Geben Sie auf der Seite **Bucket erstellen** einen weltweit eindeutigen Namen ein, der Ihren Anmeldenamen für den Bucket enthält, z. B. `dms-bucket-yoursignin`.*
4. Wählen Sie den für die AWS-Region DMS-Migrationsaufgabe.

5. Lassen Sie die übrigen Einstellungen unverändert und wählen Sie Create Bucket.

Erstellen von IAM-Ressourcen

DMS verwendet eine IAM-Rolle und -Richtlinie, um auf den S3-Bucket zuzugreifen und die Ergebnisse der Bewertung vor der Migration zu speichern.

Gehen Sie wie folgt vor, um die IAM-Richtlinie zu erstellen:

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter `https://console.aws.amazon.com/iam/`.](https://console.aws.amazon.com/iam/)
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte JSON aus.
5. Fügen Sie den folgenden JSON-Code in den Editor ein und ersetzen Sie dabei den Beispielcode. Ersetzen Sie *my-bucket* durch den Namen des Amazon S3 S3-Buckets, den Sie im vorherigen Abschnitt erstellt haben.

```
{
  "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. Wählen Sie Weiter: Tags und dann Weiter: Überprüfen.
7. Geben Sie unter Name* **DMSPremigrationAssessmentS3Policy** ein und wählen Sie dann Richtlinie erstellen aus.

Gehen Sie wie folgt vor, um die IAM-Rolle zu erstellen:

1. Wählen Sie in der IAM-Konsole im Navigationsbereich die Option Rollen aus.
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie auf der Seite Vertrauenswürdige Entität auswählen unter Typ der vertrauenswürdigen Entität die Option AWS -Service aus. Wählen Sie für Anwendungsfälle für andere AWS Dienste DMS aus.
4. Aktivieren Sie das Kontrollkästchen DMS und wählen Sie dann Weiter.
5. Wählen Sie auf der Seite „Berechtigungen hinzufügen“ die Option DMS S3Policy PremigrationAssessment aus. Wählen Sie Weiter aus.
6. Geben Sie auf der Seite Benennen, Prüfen und Erstellen **DMSPremigrationAssessmentS3Role** für Rollennamen ein und wählen Sie dann Rolle erstellen.
7. Geben Sie auf der Seite Rollen **DMSPremigrationAssessmentS3Role** für Rollennamen ein. Wählen Sie DMS S3Role. PremigrationAssessment
8. Wählen Sie auf der Seite DMS PremigrationAssessment S3Role den Tab Vertrauensbeziehungen aus. Wählen Sie Vertrauensrichtlinie bearbeiten aus.
9. Fügen Sie auf der Seite Vertrauensrichtlinie bearbeiten den folgenden JSON-Code in den Editor ein und ersetzen Sie den vorhandenen Text.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {

```

```
        "Service": "dms.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Diese Richtlinie erteilt DMS die `sts:AssumeRole` Erlaubnis, die Ergebnisse des Testlaufs vor der Migration in den S3-Bucket zu speichern.

10. Wählen Sie Richtlinie aktualisieren.

Angeben, Starten und Anzeigen von Vormigrationsbewertungsläufen

Eine Bewertung vor der Migration spezifiziert eine oder mehrere Einzelprüfungen, die auf der Grundlage einer neuen oder vorhandenen Konfiguration der Migrationsaufgabe ausgeführt werden sollen. Bei jeder individuellen Bewertung wird ein bestimmtes Element der Quell- oder Zieldatenbank bewertet, abhängig von Aspekten wie Migrationstyp, unterstützten Objekten, Indexkonfiguration und anderen Aufgabeneinstellungen, wie z. B. Tabellenzuweisungen, um die zu migrierenden Schemas und Tabellen zu identifizieren. Bei einer individuellen Bewertung könnte beispielsweise bewertet werden, welche Quelldatentypen oder Primärschlüsselformate migriert werden können und welche nicht.

Angeben von individuellen Bewertungen

Wenn Sie einen neuen Bewertungslauf erstellen, können Sie wählen, ob Sie einige oder alle Einzelprüfungen ausführen möchten, die für Ihre Aufgabenkonfiguration relevant sind.

AWS DMS unterstützt Bewertungsläufe vor der Migration für die folgenden relationalen Quell- und Zieldatenbank-Engines:

- [Bewertungen von Oracle](#)
- [SQL Server-Bewertungen](#)
- [MySQL-Bewertungen](#) (beinhaltet MariaDB und Amazon Aurora MySQL-kompatible Edition)
- [PostgreSQL-Bewertungen](#) (beinhaltet Amazon Aurora PostgreSQL-kompatible Edition)

Starten und Anzeigen von Vormigrationsbewertungsläufen

Sie können vor der Migration einen Bewertungslauf für eine neue oder bestehende Migrationsaufgabe mithilfe der AWS DMS Management Console, der und der AWS CLI API starten.

AWS DMS

So starten Sie einen Vormigrationsbewertungslauf für eine neue oder bestehende Aufgabe

1. Führen Sie auf der Seite Datenbankmigrationsaufgaben in der AWS DMS -Managementkonsole einen der folgenden Schritte aus:
 - Um eine neue Aufgabe zu erstellen und sie zu bewerten, wählen Sie Aufgabe erstellen. Die Seite Datenbankmigrationsaufgabe erstellen wird geöffnet:
 1. Geben Sie die erforderlichen Aufgabeneinstellungen zum Erstellen Ihrer Aufgabe, einschließlich der Tabellenzuweisung, ein.
 2. Im Abschnitt Bewertung vor der Migration ist das Kontrollkästchen Prüfung vor der Migration ausführen aktiviert. Diese Seite enthält die Optionen, mit denen Sie einen Testlauf für die neue Aufgabe angeben können.

Note

Wenn Sie eine neue Aufgabe erstellen, wird die Option zum automatischen Starten der Aufgabe bei ihrer Erstellung deaktiviert, wenn ein Vormigrationsbewertungslauf aktiviert wird. Sie können die Aufgabe nach Abschluss des Bewertungslaufs manuell starten.

- Um eine bestehende Aufgabe zu bewerten, wählen Sie auf der Seite mit den Aufgaben zur Datenbankmigration den Identifier für eine bestehende Aufgabe aus. Die Aufgabenseite für die ausgewählte bestehende Aufgabe wird geöffnet:
 1. Wählen Sie Aktionen und anschließend Vormigrationsbewertung erstellen aus. Die Seite Vormigrationsbewertung erstellen wird geöffnet. Sie enthält Optionen zum Angeben eines Bewertungslaufs für die bestehende Aufgabe.
- 2. Geben Sie einen eindeutigen Namen für Ihren Bewertungslauf ein, oder behalten Sie den Standardwert bei.
- 3. Wählen Sie die verfügbaren individuellen Bewertungen aus, die Sie in diesen Bewertungslauf einbeziehen möchten. Sie können die verfügbaren individuellen Bewertungen nur auf der

Grundlage Ihrer aktuellen Aufgabeneinstellungen auswählen. Standardmäßig sind alle verfügbaren individuellen Bewertungen aktiviert und ausgewählt.

- Suchen Sie in Ihrem Konto nach einem Amazon-S3-Bucket und einem Ordner zum Speichern Ihres Bewertungsergebnisberichts und wählen Sie diese aus. Informationen zum Einrichten von Ressourcen für Bewertungsläufe finden Sie unter [Schaffung der Voraussetzungen für Bewertungen vor der Migration](#).
- Wählen Sie eine IAM-Rolle mit vollständigem Kontozugriff auf Ihren ausgewählten Amazon S3-Bucket und Ordner aus oder geben Sie sie ein. Informationen zum Einrichten von Ressourcen für Bewertungsläufe finden Sie unter [Schaffung der Voraussetzungen für Bewertungen vor der Migration](#).
- Optional können Sie eine Einstellung auswählen, um den Bewertungsergebnisbericht in Ihrem Amazon-S3-Bucket zu verschlüsseln. Informationen zur S3-Bucket-Verschlüsselung finden Sie unter [Einstellung des standardmäßigen serverseitigen Verschlüsselungsverhaltens für Amazon S3 S3-Buckets](#).
- Wählen Sie Aufgabe erstellen für eine neue Aufgabe oder Erstellen für eine bestehende Aufgabe aus.

Die Seite Datenbankmigrationsaufgaben wird geöffnet. Auf dieser Seite ist Ihre neue oder geänderte Aufgabe mit dem Status Wird erstellt... aufgeführt und eine Bannermeldung weist darauf hin, dass Ihr Vormigrationsbewertungslauf beginnt, sobald die Aufgabe erstellt wurde.

AWS DMS bietet über die AWS DMS Management-Konsole, die oder die API Zugriff auf die neuesten und alle vorherigen Bewertungsläufe vor der AWS CLI Migration. AWS DMS

Um die Ergebnisse des Bewertungslaufs anzuzeigen

- Wählen Sie in der AWS DMS Management Console auf der Seite Datenbankmigrationsaufgaben den Identifier für Ihre bestehende Aufgabe aus. Die Aufgabenseite für die bestehende Aufgabe wird geöffnet.
- Wählen Sie auf der Seite für die bestehende Aufgabe die Registerkarte Vormigrationsbewertungen aus. Dadurch wird auf dieser Seite ein Abschnitt mit Bewertungen vor der Migration geöffnet, in dem die Ergebnisse der Bewertungsläufe, aufgelistet nach Namen, in umgekehrter chronologischer Reihenfolge angezeigt werden. Das neueste Ergebnis wird oben in der Liste angezeigt. Wählen Sie den Namen des Testlaufs, dessen Ergebnisse Sie anzeigen möchten.

Die Bewertungslaufergebnisse beginnen mit dem Namen des letzten Bewertungslaufs und einer Übersicht über dessen Status, gefolgt von einer Liste der angegebenen individuellen Bewertungen mit Statusangabe. Sie können nun die Informationen zum Status der individuellen Bewertungen untersuchen, indem Sie die betreffenden Namen in der Liste auswählen. Es sind Ergebnisse bis auf Tabellenspaltebene verfügbar.

Sowohl in der Statusübersicht für einen Bewertungslauf als auch für jede individuelle Bewertung wird ein Status-Wert angezeigt. Dieser Wert gibt den Gesamtstatus des Bewertungslaufs und einen ähnlichen Status für jede individuelle Bewertung an. Im Folgenden finden Sie eine Liste der Status-Werte für den Bewertungslauf:

- "cancelling" – Der Bewertungslauf wurde abgebrochen.
- "deleting" – Der Bewertungslauf wurde gelöscht.
- "failed" – Mindestens eine individuelle Bewertung wurde mit dem Status `failed` abgeschlossen.
- "error-provisioning" – Bei der Bereitstellung von Ressourcen (während des Status `provisioning`) ist ein interner Fehler aufgetreten.
- "error-executing" – Bei der Ausführung individueller Bewertungen (während des `running` Status) ist ein interner Fehler aufgetreten.
- "invalid state" – Der Bewertungslauf befindet sich in einem unbekanntem Status.
- "passed" – Alle individuellen Bewertungen wurden abgeschlossen und keine weist den Status `failed` auf.
- "provisioning" – Die für die Durchführung individueller Bewertungen erforderlichen Ressourcen werden bereitgestellt.
- "running" – Individuelle Bewertungen werden ausgeführt.
- "starting" – Der Bewertungslauf beginnt, aber es werden noch keine Ressourcen für individuelle Bewertungen bereitgestellt.
- "warning" – Mindestens eine individuelle Bewertung wurde mit dem Status `warning` abgeschlossen.

Im Folgenden finden Sie eine Liste der Status-Werte für die individuellen Bewertungen des Bewertungslaufs:

- "cancelled" – Die individuelle Bewertung wurde im Zuge des Abbruchs des Bewertungslaufs abgebrochen.

- "error" – Die individuelle Bewertung wurde nicht erfolgreich abgeschlossen.
- "failed" – Die individuelle Bewertung wurde erfolgreich abgeschlossen, das Validierungsergebnis lautet jedoch „fehlgeschlagen“. Weitere Informationen finden Sie in den Ergebnisdetails.
- "invalid state" – Die individuelle Bewertung befindet sich in einem unbekanntem Status.
- "passed" – Die individuelle Bewertung wurde mit einem erfolgreichen Validierungsergebnis abgeschlossen.
- "pending" – Die individuelle Bewertung wartet auf die Ausführung.
- "running" – Die individuelle Bewertung wird ausgeführt.
- "warning" – Die individuelle Bewertung wurde erfolgreich, jedoch mit einer Warnung als Validierungsergebnis, abgeschlossen. Weitere Informationen finden Sie in den Ergebnisdetails.

Sie können auch die JSON-Dateien für die Bewertungslaufergebnisse in Amazon S3 anzeigen.

So zeigen Sie die JSON-Dateien für den Bewertungslauf in Amazon S3 an

1. Wählen Sie in der AWS DMS Management Console den Amazon S3 S3-Bucket-Link aus, der in der Statusübersicht des Bewertungslaufs angezeigt wird. Daraufhin wird eine Liste der Bucket-Ordner und anderer Amazon-S3-Objekte angezeigt, die in dem Bucket gespeichert sind. Wenn Ihre Ergebnisse in einem Bucket-Ordner gespeichert sind, öffnen Sie den Ordner.
2. Die Ergebnisse Ihres Bewertungslaufs finden Sie in mehreren JSON-Dateien. In einer Datei `summary.json` sind die Gesamtergebnisse des Bewertungslaufs enthalten. Die übrigen Dateien sind jeweils nach einer individuellen Bewertung benannt, die für den Bewertungslauf angegeben wurde, beispielsweise `unsupported-data-types-in-source.json`. Diese Dateien enthalten jeweils die Ergebnisse der entsprechenden individuellen Bewertung aus dem ausgewählten Bewertungslauf.

Um die Ergebnisse von Bewertungsläufen vor der Migration für eine bestehende Migrationsaufgabe zu starten und anzuzeigen, können Sie die folgenden CLI-Befehle und AWS DMS API-Operationen ausführen:

- CLI: [describe-applicable-individual-assessments](#), API: [DescribeApplicableIndividualAssessments](#) – Stellt eine Liste individueller Bewertungen bereit, die Sie für einen neuen Vormigrationsbewertungslauf angeben können, unter Berücksichtigung von einem oder mehreren Aufgabenkonfigurationsparametern.

- CLI: [start-replication-task-assessment-run](#), API: [StartReplicationTaskAssessmentRun](#) – Startet einen neuen Vormigrationsbewertungslauf für eine oder mehrere individuelle Bewertungen einer bestehenden Migrationsaufgabe.
- CLI: [describe-replication-task-assessment-runs](#), API: [DescribeReplicationTaskAssessmentRuns](#) – Gibt basierend auf Filtereinstellungen eine paginierte Liste von Vormigrationsbewertungsläufen zurück.
- CLI: [describe-replication-task-individual-assessments](#), API: [DescribeReplicationTaskIndividualAssessments](#) – Gibt basierend auf Filtereinstellungen eine paginierte Liste individueller Bewertungen zurück.
- CLI: [cancel-replication-task-assessment-run](#), API: [CancelReplicationTaskAssessmentRun](#) – Bricht einen einzelnen Vormigrationsbewertungslauf ab, löscht diesen aber nicht.
- CLI: [delete-replication-task-assessment-run](#), API: [DeleteReplicationTaskAssessmentRun](#) – Löscht den Datensatz eines einzelnen Vormigrationsbewertungslaufs.

Individuelle Bewertungen

In diesem Abschnitt werden individuelle Bewertungen vor der Migration beschrieben.

Um mithilfe der AWS DMS API eine individuelle Bewertung vor der Migration zu erstellen, verwenden Sie den aufgelisteten API-Schlüssel als `IncludeOnly` Aktionsparameter.

[StartReplicationTaskAssessmentRun](#)

Themen

- [Bewertungen für alle Endpunkttypen](#)
- [Bewertungen von Oracle](#)
- [SQL Server-Bewertungen](#)
- [MySQL-Bewertungen](#)
- [MariaDB-Bewertungen](#)
- [PostgreSQL-Bewertungen](#)

Bewertungen für alle Endpunkttypen

In diesem Abschnitt werden individuelle Bewertungen vor der Migration für alle Endpunkttypen beschrieben.

Themen

- [Nicht unterstützte Datentypen](#)
- [Große Objekte \(LOBs\) werden verwendet, aber Ziel-LOB-Spalten können keine NULL-Werte annehmen](#)
- [Quelltabelle mit großen Objekten \(LOBs\), jedoch ohne Primärschlüssel oder eindeutige Einschränkungen](#)
- [Quelltabelle ohne Primärschlüssel nur für CDC- oder Volllast- und CDC-Aufgaben](#)
- [Zieltabelle ohne Primärschlüssel nur für CDC-Aufgaben](#)
- [Nicht unterstützte Quell-Primärschlüsseltypen — zusammengesetzte Primärschlüssel](#)

Nicht unterstützte Datentypen

API-Schlüssel: `unsupported-data-types-in-source`

Sucht im Quellendpunkt nach Datentypen, die DMS nicht unterstützt. Nicht alle Datentypen können zwischen Engines migriert werden.

Große Objekte (LOBs) werden verwendet, aber Ziel-LOB-Spalten können keine NULL-Werte annehmen

API-Schlüssel: `full-lob-not-nullable-at-target`

Überprüft, ob eine LOB-Spalte im Ziel NULL-Werte zulässt, wenn die Replikation den vollständigen LOB-Modus oder den Inline-LOB-Modus verwendet. DMS erfordert, dass eine LOB-Spalte Null ist, wenn diese LOB-Modi verwendet werden. Für diese Bewertung müssen die Quell- und Zieldatenbanken relational sein.

Quelltabelle mit großen Objekten (LOBs), jedoch ohne Primärschlüssel oder eindeutige Einschränkungen

API-Schlüssel: `table-with-lob-but-without-primary-key-or-unique-constraint`

Überprüft, ob Quelltabellen mit LOBs, aber ohne Primärschlüssel oder eindeutigen Schlüssel vorhanden sind. Eine Tabelle muss einen Primärschlüssel oder einen eindeutigen Schlüssel haben, damit DMS LOBs migrieren kann. Für diese Bewertung muss die Quelldatenbank relational sein.

Quelltabelle ohne Primärschlüssel nur für CDC- oder Volllast- und CDC-Aufgaben

API-Schlüssel: `table-with-no-primary-key-or-unique-constraint`

Überprüft das Vorhandensein eines Primärschlüssels oder eines eindeutigen Schlüssels in Quelltabellen für eine Migration mit vollständiger Last und Change Data Capture (CDC) oder eine reine CDC-Migration. Das Fehlen eines Primärschlüssels oder eines eindeutigen Schlüssels kann zu Leistungsproblemen während der CDC-Migration führen. Für diese Bewertung muss die Quelldatenbank relational sein und der Migrationstyp muss CDC enthalten.

Zieltabelle ohne Primärschlüssel nur für CDC-Aufgaben


API-Schlüssel: `target-table-has-unique-key-or-primary-key-for-cdc`

Überprüft, ob ein Primärschlüssel oder ein eindeutiger Schlüssel in bereits erstellten Zieltabellen für eine reine CDC-Migration vorhanden ist. Das Fehlen eines Primärschlüssels oder eines eindeutigen Schlüssels kann dazu führen, dass vollständige Tabellen im Ziel gescannt werden, wenn DMS Aktualisierungen und Löschungen vornimmt. Dies kann zu Leistungsproblemen während der CDC-Migration führen. Für diese Bewertung muss die Zieldatenbank relational sein und der Migrationstyp muss CDC enthalten.

Nicht unterstützte Quell-Primärschlüsseltypen — zusammengesetzte Primärschlüssel

API-Schlüssel: `unsupported-source-pk-type-for-elasticsearch-target`

Prüft bei der Migration zu Amazon OpenSearch Service, ob zusammengesetzte Primärschlüssel in Quelltabellen vorhanden sind. Der Primärschlüssel der Quelltable muss aus einer einzelnen Spalte bestehen. Für diese Bewertung muss es sich bei der Quelldatenbank um eine relationale Datenbank und bei der Zieldatenbank um DynamoDB handeln.

 Note

DMS unterstützt die Migration einer Quelldatenbank zu einem OpenSearch Serviceziel, bei dem der Quell-Primärschlüssel aus mehreren Spalten besteht.

Bewertungen von Oracle

In diesem Abschnitt werden individuelle Vormigrationsbewertungen für Migrationsaufgaben beschrieben, bei denen ein Oracle-Quellendpunkt verwendet wird.

Note

Für die Verwendung der Vormigrationsbewertungen in diesem Abschnitt müssen Sie die folgenden Berechtigungen zu `dms_user` hinzufügen:

```
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;
```

Weitere Informationen zu Berechtigungen bei Verwendung von Oracle als Quelle finden Sie unter [Erforderliche Benutzerkontenberechtigungen für eine selbstverwaltete Oracle-Quelle für AWS DMS](#).

Themen

- [Überprüfung der zusätzlichen Protokollierung auf Datenbankebene](#)
- [Validierung, ob der erforderliche DB-Link für Standby erstellt wird](#)
- [Oracle-Validierung, ob der LOB-Datentyp vorhanden und Binary Reader konfiguriert ist](#)
- [Überprüfung, ob es sich bei der Datenbank um CDB handelt](#)
- [Überprüfen der Oracle-Datenbankausgabe](#)
- [Validieren der Oracle-CDC-Methode für DMS](#)
- [Validieren der Oracle-RAC-Konfiguration für DMS](#)

Überprüfung der zusätzlichen Protokollierung auf Datenbankebene

API-Schlüssel: `oracle-supplemental-db-level`

Bei dieser Vormigrationsbewertung wird überprüft, ob eine minimale zusätzliche Protokollierung auf Datenbankebene aktiviert ist. Sie müssen die zusätzliche Protokollierung aktivieren, um eine Oracle-Datenbank als Migrationsquelle verwenden zu können.

Verwenden Sie die folgende Abfrage, um die zusätzliche Protokollierung zu aktivieren:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
```

Weitere Informationen finden Sie unter [Einrichten der zusätzlichen Protokollierung](#).

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Validierung, ob der erforderliche DB-Link für Standby erstellt wird

API-Schlüssel: `oracle-validate-standby-dblink`

Diese Bewertung vor der Migration validiert, ob Dblink für die Oracle-Standby-Datenbankquelle erstellt wurde. `AWSDMS_DBLINK` ist eine Voraussetzung für die Verwendung einer Standby-Datenbank als Quelle. Wenn Sie Oracle Standby als Quelle verwenden, prüft AWS DMS standardmäßig keine offenen Transaktionen.

Weitere Informationen finden Sie unter [Arbeiten mit einer selbstverwalteten Oracle-Datenbank als Quelle für AWS DMS](#).

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Oracle-Validierung, ob der LOB-Datentyp vorhanden und Binary Reader konfiguriert ist

API-Schlüssel: `oracle-binary-lob-source-validation`

Mit dieser Bewertung vor der Migration LogMiner wird überprüft, ob Oracle für einen Oracle-Datenbank-Endpunkt der Version 12c oder höher verwendet wird. AWS DMS unterstützt Oracle

nicht LogMiner für Migrationen von LOB-Spalten aus Oracle-Datenbanken Version 12c. Bei dieser Bewertung wird auch geprüft, ob LOB-Spalten vorhanden sind, und es werden entsprechende Empfehlungen gegeben.

Um Ihre Migration so zu konfigurieren, dass Oracle nicht verwendet wird LogMiner, fügen Sie Ihrem Quellendpunkt die folgende Konfiguration hinzu:

```
useLogMinerReader=N;useBfile=Y;
```


Weitere Informationen finden Sie unter [Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC](#).

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Überprüfung, ob es sich bei der Datenbank um CDB handelt

API-Schlüssel: `oracle-validate-cdb`

Bei dieser Vormigrationsbewertung wird überprüft, ob es sich bei der Datenbank um eine Container-Datenbank handelt. AWS DMS unterstützt die Multi-Tenant-Container-Stammdatenbank (CDB \$ROOT) nicht.

 Note

Diese Bewertung ist nur für Oracle-Versionen 12.1.0.1 oder höher erforderlich. Diese Bewertung ist nicht für Oracle-Versionen vor 12.1.0.1 anwendbar.

Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von Oracle als Quelle für AWS DMS](#).

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Überprüfen der Oracle-Datenbankausgabe

API-Schlüssel: `oracle-check-cdc-support-express-edition`

Bei dieser Vormigrationsbewertung wird geprüft, ob es sich bei der Oracle-Quelldatenbank um die Express Edition handelt. AWS DMS unterstützt CDC nicht für Oracle Express Edition (Oracle Database XE) Version 18.0 und höher.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Validieren der Oracle-CDC-Methode für DMS

API-Schlüssel: `oracle-recommendation-cdc-method`

Diese Bewertung vor der Migration validiert die Redo-Log-Generierung für die letzten sieben Tage und gibt eine Empfehlung, ob AWS DMS Binary Reader oder Oracle LogMiner für CDC verwendet werden soll.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen dazu, welche CDC-Methode Sie verwenden sollten, finden Sie unter [Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC](#).

Validieren der Oracle-RAC-Konfiguration für DMS

API-Schlüssel: `oracle-check-rac`

Bei dieser Vormigrationsbewertung wird geprüft, ob es sich bei der Oracle-Datenbank um einen Real-Application-Cluster handelt. Real-Application-Cluster-Datenbanken müssen korrekt konfiguriert sein. Wenn die Datenbank auf RAC basiert, empfehlen wir, AWS DMS Binary Reader for CDC anstelle von Oracle zu verwenden. LogMiner

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC](#).

SQL Server-Bewertungen

In diesem Abschnitt werden individuelle Vormigrationsbewertungen für Migrationsaufgaben beschrieben, bei denen ein Microsoft-SQL-Server-Quellendpunkt verwendet wird.

Themen

- [Prüfung, ob das Wiederherstellungsmodell für die Datenbank einfach ist](#)
- [Prüfung, ob Tabellen im Aufgabenumfang berechnete Spalten enthalten](#)
- [Prüfung, ob Tabellen im Aufgabenumfang Columnstore-Indizes aufweisen](#)
- [Prüfung, ob speicheroptimierte Tabellen Teil des Aufgabenumfangs sind](#)
- [Prüfung, ob temporale Tabellen Teil des Aufgabenumfangs sind](#)
- [Prüfung, ob verzögerte Haltbarkeit auf Datenbankebene aktiviert ist](#)
- [Überprüfung, ob die beschleunigte Datenwiederherstellung auf Datenbankebene aktiviert ist](#)
- [Prüfung, ob die Tabellenzuweisung mehr als 10 000 Tabellen mit Primärschlüsseln enthält](#)

Prüfung, ob das Wiederherstellungsmodell für die Datenbank einfach ist

API-Schlüssel: `sqlserver-check-for-recovery-model`

Diese Bewertung vor der Migration validiert das Wiederherstellungsmodell für Quellendpunkte. AWS DMS erfordert, dass das Wiederherstellungsmodell auf `Bulk logged` oder `Full` für die laufende Replikation eingestellt ist.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Voraussetzungen für die Verwendung der laufenden Replikation \(CDC\) von einer SQL-Server-Quelle aus](#).

Prüfung, ob Tabellen im Aufgabenumfang berechnete Spalten enthalten

API-Schlüssel: `sqlserver-check-for-computed-fields`

Bei dieser Bewertung vor der Migration wird geprüft, ob berechnete Spalten vorhanden sind. AWS DMS unterstützt nicht das Replizieren von Änderungen aus berechneten SQL Server-Spalten.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#).

Prüfung, ob Tabellen im Aufgabenumfang Columnstore-Indizes aufweisen

API-Schlüssel: `sqlserver-check-for-columnstore-indexes`

Bei dieser Bewertung vor der Migration wird geprüft, ob Tabellen mit Columnstore-Indizes vorhanden sind. AWS DMS unterstützt nicht das Replizieren von Änderungen aus SQL Server-Tabellen mit Columnstore-Indizes.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#).

Prüfung, ob speicheroptimierte Tabellen Teil des Aufgabenumfangs sind

API-Schlüssel: `sqlserver-check-for-memory-optimized-tables`

Bei dieser Bewertung vor der Migration wird geprüft, ob speicheroptimierte Tabellen vorhanden sind. AWS DMS unterstützt nicht das Replizieren von Änderungen aus speicheroptimierten Tabellen.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#).

Prüfung, ob temporale Tabellen Teil des Aufgabenumfangs sind

API-Schlüssel: `sqlserver-check-for-temporal-tables`

Bei dieser Bewertung vor der Migration wird geprüft, ob Temporaltabellen vorhanden sind. AWS DMS unterstützt das Replizieren von Änderungen aus zeitlichen Tabellen nicht.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#).

Prüfung, ob verzögerte Haltbarkeit auf Datenbankebene aktiviert ist

API-Schlüssel: `sqlserver-check-for-delayed-durability`

Bei dieser Bewertung vor der Migration wird geprüft, ob eine verzögerte Haltbarkeit vorliegt. AWS DMS unterstützt nicht die Replikation von Änderungen aus Transaktionen mit verzögerter Haltbarkeit.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#).

Überprüfung, ob die beschleunigte Datenwiederherstellung auf Datenbankebene aktiviert ist

API-Schlüssel: `sqlserver-check-for-accelerated-data-recovery`

Bei dieser Bewertung vor der Migration wird geprüft, ob eine beschleunigte Datenwiederherstellung vorliegt. AWS DMS unterstützt nicht das Replizieren von Änderungen aus Datenbanken mit beschleunigter Datenwiederherstellung.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen finden Sie unter [Einschränkungen bei der Verwendung von SQL Server als Quelle für AWS DMS](#).

Prüfung, ob die Tabellenzuweisung mehr als 10 000 Tabellen mit Primärschlüsseln enthält

API-Schlüssel: `sqlserver-large-number-of-tables`

Bei dieser Vormigrationsbewertung wird geprüft, ob mehr als 10 000 Tabellen mit Primärschlüsseln vorhanden sind. Bei Datenbanken, die mit MS-Replikation konfiguriert sind, können Aufgaben fehlschlagen, wenn zu viele Tabellen mit Primärschlüsseln vorhanden sind.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen zum Konfigurieren einer MS-Replikation finden Sie unter [Erfassen von Datenänderungen für selbstverwaltete SQL-Server-Quellen \(On-Premises oder in Amazon EC2\)](#).

MySQL-Bewertungen

In diesem Abschnitt werden individuelle Bewertungen vor der Migration für Migrationsaufgaben beschrieben, die einen MySQL-Quellendpunkt verwenden.

Themen

- [Überprüfen Sie, ob eine Tabelle eine andere Speicher-Engine als InnoDB verwendet](#)
- [Überprüfen Sie, ob die automatische Inkrementierung für alle für die Migration verwendeten Tabellen aktiviert ist](#)
- [Überprüfen Sie, ob das Datenbank-Binlog-Image so eingestellt ist, dass es FULL DMS CDC unterstützt](#)
- [Überprüfen Sie, ob es sich bei der Quelldatenbank um eine MySQL Read-Replica handelt](#)
- [Überprüfen Sie, ob eine Tabelle Partitionen hat, und empfehlen Sie, die Einstellungen `target_table_prep_mode` für Aufgaben bei Volllast zu verwenden](#)
- [Überprüfen Sie, ob DMS die Datenbankversion unterstützt](#)
- [Überprüfen Sie, ob die Zieldatenbank so konfiguriert ist, dass sie `local_infile` auf 1 gesetzt ist](#)
- [Überprüfen Sie, ob die Zieldatenbank Tabellen mit Fremdschlüsseln enthält](#)
- [Überprüfen Sie, ob für Quelltabellen im Aufgabenbereich kaskadierende Einschränkungen gelten](#)
- [Überprüfen Sie, ob die Timeout-Werte für eine MySQL-Quelle oder ein MySQL-Ziel geeignet sind](#)

Überprüfen Sie, ob eine Tabelle eine andere Speicher-Engine als InnoDB verwendet

API-Schlüssel: `mysql-check-table-storage-engine`

Diese Bewertung vor der Migration bestätigt, ob es sich bei der für eine Tabelle in der MySQL-Quelldatenbank verwendeten Speicher-Engine um eine andere Engine als InnoDB handelt. DMS erstellt standardmäßig Zieltabellen mit der InnoDB-Speicher-Engine. Wenn Sie eine andere Speicher-Engine als InnoDB verwenden müssen, müssen Sie die Tabelle in der Zieldatenbank manuell erstellen und Ihre DMS-Aufgabe so konfigurieren, dass sie `TRUNCATE_BEFORE_LOAD` oder

DO_NOTHING als Full-Load-Task-Einstellung verwendet. Weitere Informationen zu den Einstellungen für Volllast-Aufgaben finden Sie unter [Aufgabeneinstellungen für vollständiges Laden](#)

Weitere Hinweise zu Einschränkungen von MySQL-Endpunkten finden Sie unter [Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS](#).

Überprüfen Sie, ob die automatische Inkrementierung für alle für die Migration verwendeten Tabellen aktiviert ist

API-Schlüssel: `mysql-check-auto-increment`

Bei dieser Bewertung vor der Migration wird überprüft, ob für die Quelltabellen, die in der Aufgabe verwendet werden, die automatische Erhöhung aktiviert ist. DMS migriert das AUTO_INCREMENT-Attribut einer Spalte nicht in eine Zieldatenbank.

Weitere Hinweise zu Einschränkungen von MySQL-Endpunkten finden Sie unter [Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS](#). Informationen zum Umgang mit Identitätsspalten in MySQL finden Sie unter [Umgang mit IDENTITY-Spalten in AWS DMS: Teil 2](#).

Überprüfen Sie, ob das Datenbank-Binlog-Image so eingestellt ist, dass es **FULL** DMS CDC unterstützt

API-Schlüssel: `mysql-check-binlog-image`

Bei dieser Bewertung vor der Migration wird geprüft, ob das Binlog-Image der Quelldatenbank auf eingestellt ist. FULL In MySQL bestimmt die `binlog_row_image` Variable, wie ein binäres Protokollereignis geschrieben wird, wenn das ROW Format verwendet wird. Um die Kompatibilität mit DMS sicherzustellen und CDC zu unterstützen, setzen Sie die `binlog_row_image` Variable auf FULL. Diese Einstellung stellt sicher, dass DMS während der Migration genügend Informationen erhält, um die vollständige Data Manipulation Language (DML) für die Zieldatenbank zu erstellen.

Gehen Sie wie folgt vor, um das Binlog-Image auf FULL einzustellen:

- Für Amazon RDS ist dieser Wert FULL standardmäßig.
- Für Datenbanken, die lokal oder auf Amazon EC2 gehostet werden, legen Sie den `binlog_row_image` Wert in `my.ini` (Microsoft Windows) oder `my.cnf` (UNIX) fest.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Überprüfen Sie, ob es sich bei der Quelldatenbank um eine MySQL Read-Replica handelt

API-Schlüssel: `mysql-check-database-role`

Bei dieser Bewertung vor der Migration wird überprüft, ob es sich bei der Quelldatenbank um eine Read Replica handelt. Um die CDC-Unterstützung für DMS zu aktivieren, wenn eine Verbindung zu einer Read Replica besteht, setzen Sie den Parameter auf `log_slave_updates True`. Weitere Hinweise zur Verwendung einer selbstverwalteten MySQL-Datenbank finden Sie unter [Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#).

Gehen Sie wie folgt vor `True`, um den `log_slave_updates` Wert auf festzulegen:

- Verwenden Sie für Amazon RDS die Parametergruppe der Datenbank. Informationen zur Verwendung von RDS-Datenbankparametergruppen finden Sie unter [Arbeiten mit Parametergruppen](#) im Amazon RDS-Benutzerhandbuch.
- Für Datenbanken, die lokal oder auf Amazon EC2 gehostet werden, legen Sie den `log_slave_updates` Wert in `my.ini` (Microsoft Windows) oder `my.cnf` (UNIX) fest.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Überprüfen Sie, ob eine Tabelle Partitionen hat, und empfehlen Sie, die Einstellungen **`target_table_prep_mode`** für Aufgaben bei Volllast zu verwenden

API-Schlüssel: `mysql-check-table-partition`

Bei dieser Bewertung vor der Migration wird geprüft, ob Tabellen mit Partitionen in der Quelldatenbank vorhanden sind. DMS erstellt Tabellen ohne Partitionen auf dem MySQL-Ziel. Um partitionierte Tabellen in eine partitionierte Tabelle auf dem Ziel zu migrieren, müssen Sie wie folgt vorgehen:

- Erstellen Sie die partitionierten Tabellen in der MySQL-Zieldatenbank vorab.
- Konfigurieren Sie Ihre DMS-Aufgabe so, dass sie `TRUNCATE_BEFORE_LOAD` oder `DO_NOTHING` als Einstellung für die Volllastaufgabe verwendet.

Weitere Hinweise zu Einschränkungen von MySQL-Endpunkten finden Sie unter [Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS](#).

Überprüfen Sie, ob DMS die Datenbankversion unterstützt

API-Schlüssel: `mysql-check-supported-version`

Bei dieser Bewertung vor der Migration wird überprüft, ob die Quelldatenbankversion mit DMS kompatibel ist. CDC wird mit Amazon RDS MySQL-Versionen 5.5 oder niedriger oder mit MySQL-Versionen über 8.0.x nicht unterstützt. CDC wird nur für die MySQL-Versionen 5.6, 5.7 oder 8.0 unterstützt. Weitere Hinweise zu unterstützten MySQL-Versionen finden Sie unter [Quellendpunkte für die Datenmigration](#).

Überprüfen Sie, ob die Zieldatenbank so konfiguriert ist, dass sie **local_infile** auf 1 gesetzt ist

API-Schlüssel: `mysql-check-target-localinfile-set`

Bei dieser Bewertung vor der Migration wird geprüft, ob der `local_infile` Parameter in der Zieldatenbank auf 1 gesetzt ist. DMS erfordert, dass der Parameter 'local_infile' bei Volllast in Ihrer Zieldatenbank auf 1 gesetzt wird. Weitere Informationen finden Sie unter [Migrieren von MySQL auf MySQL mit AWS DMS](#).

Diese Bewertung ist nur für eine Volllast- oder Volllast- und CDC-Aufgabe gültig.

Überprüfen Sie, ob die Zieldatenbank Tabellen mit Fremdschlüsseln enthält

API-Schlüssel: `mysql-check-fk-target`

Bei dieser Bewertung vor der Migration wird geprüft, ob eine Volllast- oder Full- und CDC-Aufgabe, die zu einer MySQL-Datenbank migriert, Tabellen mit Fremdschlüsseln enthält. Die Standardeinstellung in DMS besteht darin, Tabellen in alphabetischer Reihenfolge zu laden. Tabellen mit Fremdschlüsseln und Einschränkungen der referenziellen Integrität können dazu führen, dass das Laden fehlschlägt, da die übergeordneten und untergeordneten Tabellen möglicherweise nicht gleichzeitig geladen werden.

Weitere Informationen zur referentiellen Integrität in DMS finden Sie im Thema Arbeiten mit Indizes, Triggern und Einschränkungen der referentiellen Integrität. [Verbessern der Leistung einer AWS DMS-Migration](#)

Überprüfen Sie, ob für Quelltabellen im Aufgabenbereich kaskadierende Einschränkungen gelten

API-Schlüssel: `mysql-check-cascade-constraints`

Bei dieser Bewertung vor der Migration wird geprüft, ob eine der MySQL-Quelltabellen über kaskadierende Einschränkungen verfügt. Kaskadierende Einschränkungen werden von DMS-

Aufgaben nicht migriert oder repliziert, da MySQL die Änderungen für diese Ereignisse nicht im Binlog aufzeichnet. Diese Einschränkungen werden zwar AWS DMS nicht unterstützt, Sie können jedoch Workarounds für relationale Datenbankziele verwenden.

Informationen zur Unterstützung von Cascade-Einschränkungen und anderen Einschränkungen finden Sie [Indizes, Fremdschlüssel oder kaskadierende Aktualisierungen oder Löschungen wurden nicht migriert](#) im Thema Problembehandlung bei Migrationsaufgaben. AWS DMS

Überprüfen Sie, ob die Timeout-Werte für eine MySQL-Quelle oder ein MySQL-Ziel geeignet sind

API-Schlüssel: `mysql-check-network-parameter`

Bei dieser Bewertung vor der Migration wird geprüft `net_read_timeout`, `net_wait_timeout` ob der MySQL-Endpunkt einer Aufgabe die `wait_timeout` Einstellungen auf mindestens 300 Sekunden gesetzt hat. Dies ist erforderlich, um Verbindungsabbrüche während der Migration zu verhindern.

Weitere Informationen finden Sie unter [Verbindungen mit einer MySQL-Ziel-Instance werden während einer Aufgabe getrennt](#).

MariaDB-Bewertungen

In diesem Abschnitt werden individuelle Bewertungen vor der Migration für Migrationsaufgaben beschrieben, die einen MariaDB-Quellendpunkt verwenden.

Um mithilfe der AWS DMS API eine individuelle Bewertung vor der Migration zu erstellen, verwenden Sie den aufgelisteten API-Schlüssel für den `Include` Parameter der Aktion.

[StartReplicationTaskAssessmentRun](#)

Themen

- [Überprüfen Sie, ob eine Tabelle eine andere Speicher-Engine als InnoDB verwendet](#)
- [Überprüfen Sie, ob die automatische Inkrementierung für alle für die Migration verwendeten Tabellen aktiviert ist](#)
- [Überprüfen Sie, ob das Datenbank-Binlog-Format so eingestellt ist, dass es DMS ROW CDC unterstützt](#)
- [Überprüfen Sie, ob das Datenbank-Binlog-Image so eingestellt ist, dass es DMS CDC FULL unterstützt](#)
- [Überprüfen Sie, ob die Quelldatenbank eine MariaDB-Read-Replica ist](#)

- [Überprüfen Sie, ob eine Tabelle Partitionen hat, und empfehlen Sie TRUNCATE_BEFORE_LOAD oder, um die Einstellungen DO_NOTHING für Vollladeaufgaben zu verwenden](#)
- [Überprüfen Sie, ob DMS die Datenbankversion unterstützt](#)
- [Überprüfen Sie, ob die Zieldatenbank so konfiguriert ist, dass sie auf 1 gesetzt local_infile ist](#)
- [Überprüfen Sie, ob die Zieldatenbank Tabellen mit Fremdschlüsseln enthält](#)
- [Überprüfen Sie, ob für Quelltabellen im Aufgabenbereich kaskadierende Einschränkungen gelten](#)
- [Überprüfen Sie, ob Quelltabellen im Aufgabenbereich generierte Spalten haben](#)
- [Überprüfen Sie, ob die Timeout-Werte für eine MariaDB-Quelle geeignet sind](#)
- [Überprüfen Sie, ob die Timeout-Werte für ein MariaDB-Ziel geeignet sind](#)

Überprüfen Sie, ob eine Tabelle eine andere Speicher-Engine als InnoDB verwendet

API-Schlüssel: mariadb-check-table-storage-engine

Diese Bewertung vor der Migration bestätigt, ob es sich bei der Speicher-Engine, die für eine Tabelle in der Quell-MariaDB-Datenbank verwendet wird, um eine andere Engine als InnoDB handelt.

DMS erstellt standardmäßig Zieltabellen mit der InnoDB-Speicher-Engine. Wenn Sie eine andere Speicher-Engine als InnoDB verwenden müssen, müssen Sie die Tabelle in der Zieldatenbank manuell erstellen und Ihre DMS-Aufgabe so konfigurieren, dass sie TRUNCATE_BEFORE_LOAD oder DO_NOTHING als Full-Load-Task-Einstellung verwendet. Weitere Informationen zu den Einstellungen für Volllast-Aufgaben finden Sie unter [Aufgabeneinstellungen für vollständiges Laden](#)

Weitere Informationen zu den Einschränkungen von MariaDB-Endpunkten finden Sie unter [Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS](#)

Überprüfen Sie, ob die automatische Inkrementierung für alle für die Migration verwendeten Tabellen aktiviert ist

API-Schlüssel: mariadb-check-auto-increment

Bei dieser Bewertung vor der Migration wird überprüft, ob für die Quelltabellen, die in der Aufgabe verwendet werden, die automatische Erhöhung aktiviert ist. DMS migriert das AUTO_INCREMENT-Attribut einer Spalte nicht in eine Zieldatenbank.

Weitere Informationen zu den Einschränkungen von MariaDB-Endpunkten finden Sie unter [Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS](#)

Informationen zum Umgang mit Identitätsspalten in MariaDB finden Sie unter [Handle IDENTITY-Spalten in AWS DMS: Teil 2](#).

Überprüfen Sie, ob das Datenbank-Binlog-Format so eingestellt ist, dass es DMS **ROW CDC** unterstützt

API-Schlüssel: `mariadb-check-binlog-format`

Bei dieser Bewertung vor der Migration wird überprüft, ob das Binlog-Format der Quelldatenbank so eingestellt ist, dass es DMS Change Data Row Capture (CDC) unterstützt.

Gehen Sie wie folgt vor, um das Binlog-Format auf einzustellen: ROW

- Verwenden Sie für Amazon RDS die Parametergruppe der Datenbank. Informationen zur Verwendung einer RDS-Parametergruppe finden Sie unter [Konfiguration der MySQL-Binärprotokollierung](#) im Amazon RDS-Benutzerhandbuch.
- Für Datenbanken, die lokal oder auf Amazon EC2 gehostet werden, legen Sie den `binlog_format` Wert in `my.ini` (Microsoft Windows) oder `my.cnf` (UNIX) fest.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen zu selbst gehosteten MariaDB-Servern finden Sie unter [Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#)

Überprüfen Sie, ob das Datenbank-Binlog-Image so eingestellt ist, dass es DMS CDC **FULL** unterstützt

API-Schlüssel: `mariadb-check-binlog-image`

Bei dieser Bewertung vor der Migration wird geprüft, ob das Binlog-Image der Quelldatenbank auf eingestellt ist. FULL In MariaDB bestimmt die `binlog_row_image` Variable, wie ein binäres Protokollereignis geschrieben wird, wenn das ROW Format verwendet wird. Um die Kompatibilität mit DMS sicherzustellen und CDC zu unterstützen, setzen Sie die Variable auf `binlog_row_image FULL`. Diese Einstellung stellt sicher, dass DMS während der Migration genügend Informationen erhält, um die vollständige Data Manipulation Language (DML) für die Zieldatenbank zu erstellen.

Gehen Sie wie folgt vor, um das Binlog-Image auf FULL einzustellen:

- Für Amazon RDS ist dieser Wert FULL standardmäßig.
- Für Datenbanken, die lokal oder auf Amazon EC2 gehostet werden, legen Sie den `binlog_row_image` Wert in `my.ini` (Microsoft Windows) oder `my.cnf` (UNIX) fest.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Weitere Informationen zu selbst gehosteten MariaDB-Servern finden Sie unter [Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#)

Überprüfen Sie, ob die Quelldatenbank eine MariaDB-Read-Replica ist

API-Schlüssel: `mariadb-check-database-role`

Bei dieser Bewertung vor der Migration wird überprüft, ob es sich bei der Quelldatenbank um eine Read Replica handelt. Um die CDC-Unterstützung für DMS zu aktivieren, wenn eine Verbindung zu einer Read Replica besteht, setzen Sie den Parameter auf `log_slave_updates True`. Weitere Hinweise zur Verwendung einer selbstverwalteten MySQL-Datenbank finden Sie unter [Verwendung einer selbstverwalteten MySQL-kompatiblen Datenbank als Quelle für AWS DMS](#).

Gehen Sie wie folgt vor `True`, um den `log_slave_updates` Wert auf festzulegen:

- Verwenden Sie für Amazon RDS die Parametergruppe der Datenbank. Informationen zur Verwendung von RDS-Datenbankparametergruppen finden Sie unter [Arbeiten mit Parametergruppen](#) im Amazon RDS-Benutzerhandbuch.
- Für Datenbanken, die lokal oder auf Amazon EC2 gehostet werden, legen Sie den `log_slave_updates` Wert in `my.ini` (Microsoft Windows) oder `my.cnf` (UNIX) fest.

Diese Bewertung ist nur für eine Migration mit vollständigem Ladevorgang und CDC oder eine reine CDC-Migration gültig. Diese Bewertung ist nicht für eine Migration nur mit vollständigem Ladevorgang gültig.

Überprüfen Sie, ob eine Tabelle Partitionen hat, und empfehlen Sie **TRUNCATE_BEFORE_LOAD** oder, um die Einstellungen **DO_NOTHING** für Vollladeaufgaben zu verwenden

API-Schlüssel: `mariadb-check-table-partition`

Bei dieser Bewertung vor der Migration wird geprüft, ob Tabellen mit Partitionen in der Quelldatenbank vorhanden sind. DMS erstellt Tabellen ohne Partitionen auf dem MariaDB-Ziel. Um partitionierte Tabellen in eine partitionierte Tabelle auf dem Ziel zu migrieren, müssen Sie wie folgt vorgehen:

- Erstellen Sie die partitionierten Tabellen in der MariaDB-Zieldatenbank vorab.

- Konfigurieren Sie Ihre DMS-Aufgabe so, dass sie TRUNCATE_BEFORE_LOAD oder DO_NOTHING als Einstellung für die Volllastaufgabe verwendet.

Weitere Informationen zu den Einschränkungen von MariaDB-Endpunkten finden Sie unter [Einschränkungen bei der Verwendung einer MySQL-Datenbank als Quelle für AWS DMS](#)

Überprüfen Sie, ob DMS die Datenbankversion unterstützt

API-Schlüssel: mariadb-check-supported-version

Bei dieser Bewertung vor der Migration wird überprüft, ob die Quelldatenbankversion mit DMS kompatibel ist. CDC wird mit Amazon RDS MariaDB-Versionen 10.4 oder niedriger oder mit MySQL-Versionen über 10.11 nicht unterstützt. Weitere Informationen zu unterstützten MariaDB-Versionen finden Sie unter [Quellendpunkte für die Datenmigration](#)

Überprüfen Sie, ob die Zieldatenbank so konfiguriert ist, dass sie auf 1 gesetzt **local_infile** ist

API-Schlüssel: mariadb-check-target-localinfile-set

Bei dieser Bewertung vor der Migration wird geprüft, ob der `local_infile` Parameter in der Zieldatenbank auf 1 gesetzt ist. DMS erfordert, dass der Parameter 'local_infile' bei Volllast in Ihrer Zieldatenbank auf 1 gesetzt wird. Weitere Informationen finden Sie unter [Migrieren von MySQL auf MySQL mit AWS DMS](#).

Diese Bewertung ist nur für eine Volllastaufgabe gültig.

Überprüfen Sie, ob die Zieldatenbank Tabellen mit Fremdschlüsseln enthält

API-Schlüssel: mariadb-check-fk-target

Bei dieser Bewertung vor der Migration wird geprüft, ob eine Volllast- oder Full- und CDC-Aufgabe, die zu einer MariaDB-Datenbank migriert, Tabellen mit Fremdschlüsseln enthält. Die Standardeinstellung in DMS besteht darin, Tabellen in alphabetischer Reihenfolge zu laden. Tabellen mit Fremdschlüsseln und Einschränkungen der referenziellen Integrität können dazu führen, dass das Laden fehlschlägt, da die übergeordneten und untergeordneten Tabellen möglicherweise nicht gleichzeitig geladen werden.

Weitere Informationen zur referentiellen Integrität in DMS finden Sie im Thema Arbeiten mit Indizes, Triggern und Einschränkungen der referentiellen Integrität. [Verbessern der Leistung einer AWS DMS-Migration](#)

Überprüfen Sie, ob für Quelltabellen im Aufgabenbereich kaskadierende Einschränkungen gelten

API-Schlüssel: `mariadb-check-cascade-constraints`

Bei dieser Bewertung vor der Migration wird geprüft, ob eine der MariaDB-Quelltabellen kaskadierende Einschränkungen aufweist. Kaskadierende Einschränkungen werden von DMS-Aufgaben nicht migriert oder repliziert, da MariaDB die Änderungen für diese Ereignisse nicht im Binlog aufzeichnet. Unterstützt diese Einschränkungen zwar AWS DMS nicht, Sie können jedoch Problemumgehungen für relationale Datenbankziele verwenden.

Informationen zur Unterstützung von Cascade-Einschränkungen und anderen Einschränkungen finden Sie [Indizes, Fremdschlüssel oder kaskadierende Aktualisierungen oder Löschungen wurden nicht migriert](#) im Thema Problembehandlung bei Migrationsaufgaben. AWS DMS

Überprüfen Sie, ob Quelltabellen im Aufgabenbereich generierte Spalten haben

API-Schlüssel: `mariadb-check-generated-columns`

Bei dieser Bewertung vor der Migration wird geprüft, ob eine der MariaDB-Quelltabellen generierte Spalten enthält. DMS-Aufgaben migrieren oder replizieren keine generierten Spalten.

Informationen zur Migration generierter Spalten finden Sie unter [???](#)

Überprüfen Sie, ob die Timeout-Werte für eine MariaDB-Quelle geeignet sind

API-Schlüssel: `mariadb-check-source-network-parameter`

Bei dieser Bewertung vor der Migration wird geprüft, ob der MariaDB-Quellendpunkt einer Aufgabe die `net_read_timeout` `wait_timeout` Einstellungen `net_wait_timeout` und auf mindestens 300 Sekunden gesetzt hat. Dies ist erforderlich, um Verbindungsabbrüche während der Migration zu verhindern.

Weitere Informationen finden Sie unter [Verbindungen mit einer MySQL-Ziel-Instance werden während einer Aufgabe getrennt](#).

Überprüfen Sie, ob die Timeout-Werte für ein MariaDB-Ziel geeignet sind

API-Schlüssel: `mariadb-check-target-network-parameter`

Bei dieser Bewertung vor der Migration wird geprüft, ob der MariaDB-Zielendpunkt einer Aufgabe die `net_read_timeout` `wait_timeout` Einstellungen `net_wait_timeout` und mindestens 300 Sekunden hat. Dies ist erforderlich, um Verbindungsabbrüche während der Migration zu verhindern.

Weitere Informationen finden Sie unter [Verbindungen mit einer MySQL-Ziel-Instance werden während einer Aufgabe getrennt](#).

PostgreSQL-Bewertungen

In diesem Abschnitt werden individuelle Bewertungen vor der Migration für Migrationsaufgaben beschrieben, die einen PostgreSQL-Quellendpunkt verwenden.

Themen

- [Überprüfen Sie, ob die Version der Quelldatenbank von DMS für die Migration unterstützt wird](#)
- [Überprüfen Sie den `logical_decoding_work_mem` Parameter in der Quelldatenbank](#)
- [Überprüfen Sie, ob die Quelldatenbank Transaktionen mit langer Laufzeit enthält](#)
- [Überprüfen Sie den Quelldatenbankparameter `max_slot_wal_keep_size`](#)
- [Überprüfen Sie, ob der Quelldatenbankparameter so eingestellt ist, dass er CDC unterstützt.](#)
- [Überprüfen Sie, ob die Quelldatenbank für konfiguriert ist `PGLOGICAL`](#)
- [Überprüfen Sie, ob der Primärschlüssel der Quelltable vom LOB-Datentyp ist](#)
- [Überprüfen Sie, ob die Quelltable einen Primärschlüssel hat](#)
- [Überprüfen Sie, ob vorbereitete Transaktionen in der Quelldatenbank vorhanden sind](#)
- [Überprüfen Sie, ob `wal_sender_timeout` ein für die Unterstützung von DMS CDC erforderlicher Mindestwert festgelegt ist](#)
- [Überprüfen Sie, ob in `wal_level` der Quelldatenbank auf logisch gesetzt ist](#)

Überprüfen Sie, ob die Version der Quelldatenbank von DMS für die Migration unterstützt wird

API-Schlüssel: `postgres-check-dbversion`

Bei dieser Bewertung vor der Migration wird überprüft, ob die Quelldatenbankversion kompatibel ist mit. AWS DMS

Überprüfen Sie den **`logical_decoding_work_mem`** Parameter in der Quelldatenbank

API-Schlüssel: `postgres-check-for-logical-decoding-work-mem`

Bei dieser Bewertung vor der Migration wird empfohlen, den `logical_decoding_work_mem` Parameter in der Quelldatenbank zu optimieren. In einer Datenbank mit hohem Transaktionsvolumen,

in der Sie möglicherweise lange laufende Transaktionen oder viele Untertransaktionen haben, kann dies zu einem erhöhten Speicherverbrauch bei der logischen Dekodierung und der Notwendigkeit führen, Daten auf die Festplatte zu übertragen. Dies führt zu einer hohen Latenz der DMS-Quelle während der Replikation. In solchen Szenarien müssen Sie möglicherweise Anpassungen vornehmen. `logical_decoding_work_mem` Dieser Parameter wird in PostgreSQL-Versionen 13 und höher unterstützt.

Überprüfen Sie, ob die Quelldatenbank Transaktionen mit langer Laufzeit enthält

API-Schlüssel: `postgres-check-longrunningtxn`

Bei dieser Bewertung vor der Migration wird überprüft, ob die Quelldatenbank Transaktionen mit langer Laufzeit enthält, die länger als 10 Minuten dauerten. Das Starten der Aufgabe schlägt möglicherweise fehl, da DMS beim Starten der Aufgabe standardmäßig nach offenen Transaktionen sucht.

Überprüfen Sie den Quelldatenbankparameter `max_slot_wal_keep_size`

API-Schlüssel: `postgres-check-maxslot-wal-keep-size`

Bei dieser Bewertung vor der Migration wird der für konfigurierte Wert überprüft. `max_slot_wal_keep_size` Wenn dieser Wert auf einen anderen Wert eingestellt `max_slot_wal_keep_size` ist, schlägt die DMS-Aufgabe möglicherweise fehl, da die erforderlichen WAL-Dateien entfernt werden.

Überprüfen Sie, ob der Quelldatenbankparameter so eingestellt `postgres-check-maxwalsenders` ist, dass er CDC unterstützt.

API-Schlüssel: `postgres-check-maxwalsenders`

Bei dieser Bewertung vor der Migration wird der Wert überprüft, für den in der Quelldatenbank `max_wal_senders` konfiguriert wurde. DMS muss `max_wal_senders` auf einen Wert größer als 1 gesetzt sein, um Change Data Capture (CDC) zu unterstützen.

Überprüfen Sie, ob die Quelldatenbank für konfiguriert ist `PGLOGICAL`

API-Schlüssel: `postgres-check-pglogical`

Bei dieser Bewertung vor der Migration wird überprüft, ob der `shared_preload_libraries` Wert auf Unterstützung `PGLOGICAL` für `pglogical` CDC festgelegt ist. Beachten Sie, dass Sie diese

Bewertung ignorieren können, wenn Sie planen, die Testdekodierung für die logische Replikation zu verwenden.

Überprüfen Sie, ob der Primärschlüssel der Quelltable vom LOB-Datentyp ist

API-Schlüssel: `postgres-check-pk-lob`

Bei dieser Bewertung vor der Migration wird überprüft, ob der Primärschlüssel einer Tabelle vom Datentyp Large Object (LOB) ist. DMS unterstützt keine Replikation, wenn die Quelltable eine LOB-Spalte als Primärschlüssel hat.

Überprüfen Sie, ob die Quelltable einen Primärschlüssel hat

API-Schlüssel: `postgres-check-pk`

Bei dieser Bewertung vor der Migration wird überprüft, ob Primärschlüssel für die im Aufgabenbereich verwendeten Tabellen vorhanden sind. DMS unterstützt keine Replikation für Tabellen ohne Primärschlüssel, es sei denn, die Replikatidentität ist in der Quelltable `full` auf festgelegt.

Überprüfen Sie, ob vorbereitete Transaktionen in der Quelldatenbank vorhanden sind

API-Schlüssel: `postgres-check-preparedtxn`

Bei dieser Bewertung vor der Migration wird überprüft, ob in der Quelldatenbank vorbereitete Transaktionen vorhanden sind. Die Erstellung des Replikationssteckplatzes reagiert möglicherweise nicht mehr, wenn in der Quelldatenbank vorbereitete Transaktionen vorhanden sind.

Überprüfen Sie, ob `wal_sender_timeout` ein für die Unterstützung von DMS CDC erforderlicher Mindestwert festgelegt ist

API-Schlüssel: `postgres-check-walsenderstimeout`

Bei dieser Bewertung vor der Migration `wal_sender_timeout` wird überprüft, ob der Wert auf mindestens 10000 Millisekunden (10 Sekunden) festgelegt ist. Eine DMS-Aufgabe mit CDC benötigt mindestens 10000 Millisekunden (10 Sekunden) und schlägt fehl, wenn der Wert unter 10000 liegt.

Überprüfen Sie, ob in `wal_level` der Quelldatenbank auf logisch gesetzt ist

API-Schlüssel: `postgres-check-wallevel`

Bei dieser Bewertung vor der Migration wird überprüft, ob der Wert auf `wal_level` logisch gesetzt ist. Damit DMS CDC funktioniert, muss dieser Parameter in der Quelldatenbank aktiviert sein.

Datentypbewertungen starten und anzeigen (Legacy)

Note

In diesem Abschnitt werden ältere Inhalte beschrieben. Es wird empfohlen, die Testläufe vor der Migration zu verwenden, die weiter oben unter beschrieben wurden. [Angeben, Starten und Anzeigen von Vormigrationsbewertungsläufen](#)

Datentypbewertungen sind in der Konsole nicht verfügbar. Sie können Datentypbewertungen nur mit der API oder CLI ausführen, und Sie können die Ergebnisse einer Datentypbewertung nur im S3-Bucket der Aufgabe anzeigen.

Eine Datentypbewertung identifiziert Datentypen in einer Quelldatenbank, die möglicherweise nicht korrekt migriert werden, weil das Ziel sie nicht unterstützt. AWS DMS liest während dieser Bewertung die Quelldatenbankschemas für eine Migrationsaufgabe und erstellt eine Liste der Spaltendatentypen. Anschließend wird diese Liste mit einer vordefinierten Liste von Datentypen verglichen, die von AWS DMS unterstützt werden. Wenn Ihre Migrationsaufgabe Datentypen enthält, die nicht unterstützt werden, wird ein Bericht AWS DMS erstellt, in dem Sie nachsehen können, ob Ihre Migrationsaufgabe Datentypen enthält, die nicht unterstützt werden. AWS DMS erstellt keinen Bericht, wenn Ihre Migrationsaufgabe keine Datentypen enthält, die nicht unterstützt werden.

AWS DMS unterstützt die Erstellung von Berichten zur Bewertung von Datentypen für die folgenden relationalen Datenbanken:

- Oracle
- SQL Server
- PostgreSQL
- MySQL
- MariaDB
- Amazon Aurora

Sie können mithilfe der CLI und der SDKs einen Bericht zur Bewertung des Datentyps starten und anzeigen, um auf die AWS DMS API zuzugreifen:

- In der CLI wird zum Starten einer Datentypbewertung der Befehl [start-replication-task-assessment](#) und zum Anzeigen des neuesten Berichts zur Datentypbewertung im JSON-Format der Befehl [describe-replication-task-assessment-results](#) verwendet.

- Die AWS DMS API verwendet den [StartReplicationTaskAssessment](#)Vorgang, um eine Datentypbewertung zu starten, und verwendet den [DescribeReplicationTaskAssessmentResults](#)Vorgang, um den neuesten Datentypbewertungsbericht im JSON-Format anzuzeigen.

Der Bericht zur Datentypbewertung enthält in einer einzelnen JSON-Datei eine Zusammenfassung mit einer Auflistung der nicht unterstützten Datentypen und ihrer jeweiligen Spaltenanzahl. Er enthält eine Liste der Datenstrukturen für jeden nicht unterstützten Datentyp, einschließlich der Schemas, Tabellen und Spalten mit dem nicht unterstützten Datentyp. Sie können den Bericht dazu verwenden, die Quelldatentypen zu ändern und die Chancen für eine erfolgreiche Migration zu verbessern.

Es gibt zwei Ebenen von nicht unterstützten Datentypen. Datentypen, die in dem Bericht als "nicht unterstützt" angezeigt werden, können nicht migriert werden. Datentypen, die in dem Bericht als teilweise unterstützt angezeigt werden, sind eventuell in einen anderen Datentyp konvertierbar, werden möglicherweise aber nicht wie erwartet migriert.

Das folgende Beispiel zeigt, wie ein Bericht zur Datentypbewertung aussehen könnte.

```
{
  "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 speichert die neuesten und alle vorherigen Datentypbewertungen in einem Amazon S3 S3-Bucket, der von AWS DMS in Ihrem Konto erstellt wurde. Der Name des Amazon-S3-Buckets hat das folgende Format, wobei *customerID* Ihre Kunden-ID und *CustomerDNS* eine interne Kennung ist.

```
dms-customerID-customerDNS
```

Note

Standardmäßig können Sie bis zu 100 Amazon-S3-Buckets in jedem Ihrer AWS -Konten erstellen. Da in Ihrem Konto ein Bucket AWS DMS erstellt wird, stellen Sie sicher, dass es Ihr Bucket-Limit nicht überschreitet. Andernfalls schlägt die Datentypbewertung fehl.

Alle Berichte zur Datentypbewertung für eine bestimmte Migrationsaufgabe werden in einem Bucket-Ordner gespeichert, der mit der Aufgaben-ID benannt ist. Der Dateiname jedes Berichts entspricht

dem Datum der Datentypbewertung im Format yyyy-mm-dd-hh -mm. Sie können frühere Berichte zu Datentypbewertungen in der Amazon-S3-Managementkonsole anzeigen und vergleichen.

AWS DMS erstellt außerdem eine AWS Identity and Access Management (IAM-) Rolle, um den Zugriff auf den S3-Bucket zu ermöglichen, der für diese Berichte erstellt wurde. Der Rollename lautet `dms-access-for-tasks`. Die Rolle verwendet die Richtlinie `AmazonDMSRedshiftS3Role`. Wenn bei der Ausführung ein `ResourceNotFoundFault` Fehler auftritt `StartReplicationTaskAssessment`, finden Sie [ResourceNotFoundFault](#) im Abschnitt Fehlerbehebung Informationen zur manuellen Erstellung der `dms-access-for-tasks` Rolle.

Problembehandlung bei Bewertungsläufen

Im Folgenden finden Sie Themen zur Behebung von Problemen bei der Ausführung von Bewertungsberichten mit AWS Database Migration Service. Diese Themen können Ihnen helfen, häufig auftretende Probleme zu lösen.

Themen

- [ResourceNotFoundFault beim Laufen StartReplicationTaskAssessment](#)

ResourceNotFoundFault beim Laufen StartReplicationTaskAssessment

Beim Ausführen der [StartReplicationTaskAssessment](#) Aktion kann die folgende Ausnahme auftreten.

```
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
```

Wenn Sie auf diese Ausnahme stoßen, erstellen Sie die `dms-access-for-tasks` Rolle, indem Sie wie folgt vorgehen:

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Roles (Rolle) aus.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie auf der Seite Vertrauenswürdige Entität auswählen für Vertrauenswürdigen Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
5. Fügen Sie den folgenden JSON-Code in den Editor ein und ersetzen Sie den vorhandenen Text.

```
{
```

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

Die vorherige Richtlinie gewährt die `sts:AssumeRole` Genehmigung für AWS DMS. Wenn Sie die `AmazonDMSRedShifts3Role`-Richtlinie hinzufügen, kann DMS den S3-Bucket in Ihrem Konto erstellen und die Ergebnisse der Datentypbewertung in diesen S3-Bucket übernehmen.

6. Wählen Sie Weiter aus.
7. Suchen Sie auf der Seite „Berechtigungen hinzufügen“ nach der Richtlinie `AmazonDMSRedshiftS3Role` und fügen Sie sie hinzu. Wählen Sie Weiter aus.
8. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Namen für die Rolle ein. `dms-access-for-tasks` Wählen Sie Rolle erstellen aus.

Angeben zusätzlicher Daten für Aufgabeneinstellungen

Wenn Sie eine Replikationsaufgabe für einige AWS DMS-Endpunkte erstellen oder ändern, benötigt die Aufgabe möglicherweise zusätzliche Informationen, um die Migration durchzuführen. Sie können diese zusätzlichen Informationen über eine Option in der DMS-Konsole angeben. Sie können sie auch mit dem `TaskData`-Parameter für die DMS-API-Operation `CreateReplicationTask` oder `ModifyReplicationTask` angeben.

Wenn Ihr Zielendpunkt Amazon Neptune ist, müssen Sie zusätzlich zur Tabellenzuweisung Zuweisungsdaten angeben. Diese zusätzlichen Zuweisungsdaten geben an, wie relationale Quelldaten in die Zieldiagrammdaten konvertiert werden, die von einer Neptune-Datenbank verwendet werden können. In diesem Fall können Sie eines von zwei möglichen Formaten verwenden. Weitere Informationen finden Sie unter [Angeben von Graph-Zuordnungsregeln mit Gremlin und R2RML für Amazon Neptune als Ziel](#).

Überwachen von AWS-DMS-Aufgaben

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Verfügbarkeit und Performance von AWS DMS und Ihren AWS-Lösungen. Sie sollten Überwachungsdaten von allen Teilen Ihrer AWS-Lösung erfassen, damit Sie einen Multi-Point-Fehler leichter debuggen können, wenn ein solcher auftritt. AWS bietet verschiedene Tools zur Überwachung von AWS DMS-Aufgaben und -Ressourcen und zur Reaktion auf potenzielle Vorfälle.

AWS DMS-Ereignisse und Benachrichtigungen

AWS DMS verwendet Amazon Simple Notification Service (Amazon SNS), um Benachrichtigungen bereitzustellen, wenn ein AWS DMS-Ereignis eintritt, z. B. das Erstellen oder Löschen einer Replikations-Instance. AWS DMS gruppiert Ereignisse in Kategorien, die Sie abonnieren können, so dass Sie benachrichtigt werden, wenn ein Ereignis in einer bestimmten Kategorie eintritt. Wenn Sie beispielsweise die Kategorie "Erstellung" für eine bestimmte Replikations-Instance abonnieren, werden Sie benachrichtigt, sobald ein erstellungsbezogenes Ereignis auftritt, das sich auf Ihre Replikations-Instance auswirkt. Sie können mit diesen Benachrichtigungen in einem beliebigen Format arbeiten, das von Amazon SNS für eine AWS-Region unterstützt wird, z. B. als E-Mail-Nachricht, Textnachricht oder Aufruf an einen HTTP-Endpunkt. Weitere Informationen finden Sie unter [Arbeiten mit Amazon-SNS-Ereignissen und Benachrichtigungen in AWS Database Migration Service](#).

Aufgabenstatus

Sie können den Fortschritt einer Aufgabe überwachen, indem Sie den Aufgabenstatus überprüfen und die Steuerungstabelle der Aufgabe überwachen. Der Aufgabenstatus gibt den Zustand einer AWS DMS-Aufgabe und die zugehörigen Ressourcen an. Er enthält Hinweise dazu, ob die Aufgabe erstellt, gestartet, ausgeführt oder gestoppt wird. Er enthält auch den aktuellen Status der Tabellen, die die Aufgabe migriert, z. B. wenn eine vollständige Last einer Tabelle begonnen hat oder gerade ausgeführt wird, sowie Details wie die Anzahl der aufgetretenen Einfügungen, Löschungen und Aktualisierungen für die Tabelle. Weitere Informationen zum Überwachen des Zustands von Aufgaben- und Aufgabenressourcen finden Sie unter [Aufgabenstatus](#) und [Tabellenstatus während der Aufgaben](#). Weitere Informationen zu Steuerungstabellen finden Sie unter [Kontrolltabellen-Aufgabeneinstellungen](#).

Amazon-CloudWatch-Alarme und -Protokolle

Mithilfe von Amazon-CloudWatch-Alarmen überwachen Sie eine oder mehrere Aufgaben-Metriken über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten

Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon-SNS-Thema gesendet. CloudWatch-Alarmläufe rufen keine Aktionen auf, nur weil sie einen besonderen Status haben. Vielmehr muss der Zustand geändert und für eine bestimmte Anzahl von Perioden beibehalten werden. AWS DMS verwendet CloudWatch auch, um Aufgabeninformationen während des Migrationsprozesses zu protokollieren. Sie können die AWS CLI- oder die AWS DMS-API verwenden, um Informationen zu den Aufgabenprotokollen anzuzeigen. Weitere Informationen zur Verwendung von CloudWatch mit AWS DMS finden Sie unter [Überwachen von Replikationsaufgaben mit Amazon CloudWatch](#). Weitere Informationen zur Überwachung von AWS DMS-Metriken finden Sie unter [AWS Database Migration Service-Metriken](#). Weitere Hinweise zum Verwenden von AWS DMS-Aufgabenprotokollen finden Sie unter [Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen](#).

Time-Travel-Protokolle

Um Replikationsaufgaben zu protokollieren und zu debuggen, können Sie AWS DMS Time Travel verwenden. Bei diesem Konzept verwenden Sie Amazon S3, um Protokolle zu speichern und sie mit Ihren Verschlüsselungsschlüsseln zu verschlüsseln. Sie können Ihre S3-Protokolle mithilfe von Datums- und Uhrzeitfiltern abrufen und die Protokolle dann nach Bedarf anzeigen, herunterladen und verschleiern. Auf diese Weise können Sie „in die Vergangenheit reisen“, um Datenbankaktivitäten zu untersuchen.

Sie können Time Travel mit DMS-unterstützten PostgreSQL-Quellendpunkten und DMS-unterstützten PostgreSQL- und MySQL-Zielendpunkten verwenden. Sie können Time Travel nur für Volllast- und CDC-Aufgaben oder nur für CDC-Aufgaben aktivieren. Um Time Travel zu aktivieren oder vorhandene Time-Travel-Einstellungen zu ändern, stellen Sie sicher, dass Ihre Aufgabe angehalten wurde.

Weitere Informationen zu Time-Travel-Protokollen finden Sie unter [Time-Travel-Aufgabeneinstellungen](#). Bewährte Methoden für die Verwendung von Time-Travel-Protokollen finden Sie unter [Problembehandlung bei Replikationsaufgaben mit Time Travel](#).

AWS CloudTrail-Protokolle

AWS DMS ist in AWS CloudTrail integriert, einen Service, der die Aktionen eines Benutzers, einer IAM-Rolle oder eines AWS-Service in AWS DMS protokolliert. CloudTrail erfasst alle API-Aufrufe für AWS DMS als Ereignisse, einschließlich Aufrufen von der AWS DMS-Konsole und von Code-Aufrufen an die AWS DMS-API-Vorgänge. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon S3-Bucket, einschließlich Ereignisse für AWS DMS aktivieren. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole trotzdem in Event history (Ereignisverlauf)

anzeigen. Mit den von CloudTrail erfassten Informationen können Sie die an AWS DMS gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen. Weitere Informationen finden Sie unter [Protokollierung von AWS DMS-API-Aufrufen mit AWS CloudTrail](#).

Datenbankprotokolle

Sie können Datenbankprotokolle für Ihre Aufgabenendpunkte mithilfe der AWS Management Console-, AWS CLI- oder der API für Ihren AWS-Datenbankservice anzeigen, herunterladen und überwachen. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Datenbankservice unter [AWS-Dokumentation](#).

Weitere Informationen finden Sie unter den folgenden Themen.

Themen

- [Aufgabenstatus](#)
- [Tabellenstatus während der Aufgaben](#)
- [Überwachen von Replikationsaufgaben mit Amazon CloudWatch](#)
- [AWS Database Migration Service-Metriken](#)
- [Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen](#)
- [Protokollierung von AWS DMS-API-Aufrufen mit AWS CloudTrail](#)
- [AWS DMS-Kontext-Protokollierung](#)

Aufgabenstatus

Der Aufgabenstatus zeigt den Zustand der Aufgabe an. Die folgende Tabelle zeigt die möglichen Status einer Aufgabe:

Aufgabenstatus	Beschreibung
Erstellen	AWS DMS erstellt die Aufgabe.
In Ausführung	Die Aufgabe führt die angegebenen Migrationsvorgänge aus.
Angehalten	Die Aufgabe wird angehalten.

Aufgabenstatus	Beschreibung
Wird angehalten	Die Aufgabe wird angehalten. Dies ist in der Regel ein Hinweis auf ein Eingreifen in die Aufgabe durch den Benutzer.
Wird gelöscht	Die Aufgabe wird gelöscht, in der Regel von einer Anforderung für einen Benutzereingriff.
Fehlgeschlagen	Die Aufgabe ist fehlgeschlagen. Weitere Informationen finden Sie in den Aufgabenprotokolldateien.
Fehler	Der Auftrag wurde aufgrund eines Fehlers angehalten. Eine kurze Beschreibung des Aufgabenfehlers finden Sie im Abschnitt „Letzte Fehlermeldung“ auf der Registerkarte Übersicht.
Ausführung mit Fehlern	Die Aufgabe wird mit einem Fehlerstatus ausgeführt. Dies zeigt normalerweise an, dass eine oder mehrere der Tabellen in der Aufgabe nicht migriert werden konnten. Die Aufgabe fährt fort, andere Tabellen gemäß den Auswahlregeln zu laden.
Wird gestartet	Die Aufgabe stellt eine Verbindung mit der Replikations-Instance und den Quell- und Zielendpunkten her. Alle Filter und Umwandlungen werden angewendet.
Bereit	Die Aufgabe kann ausgeführt werden. Dieser Status folgt in der Regel dem Status "Creating".
Ändern	Die Aufgabe wird geändert; in der Regel aufgrund einer Benutzeraktion, durch die die Aufgabeneinstellungen geändert wurden.

Aufgabenstatus	Beschreibung
Wird verschoben	Die Aufgabe wird derzeit auf eine andere Replikations-Instance verschoben. Die Replikation bleibt in diesem Status, bis die Verschiebung abgeschlossen ist. Das Löschen der Aufgabe ist der einzige Vorgang, der für die Replikationsaufgabe zulässig ist, während sie verschoben wird.
Fehler beim Verschieben	Die Aufgabenverschiebung ist aus irgendeinem Grund fehlgeschlagen, z. B. weil nicht genügend Speicherplatz auf der Ziel-Replikations-Instance vorhanden war. Wenn sich eine Replikationsaufgabe in diesem Status befindet, kann sie gestartet, geändert, verschoben oder gelöscht werden.
Testen	Die für diese Aufgabe angegebene Datenbankmigration wird als Reaktion auf die Ausführung des Vorgangs StartReplicationTaskAssessmentRun oder StartReplicationTaskAssessment getestet.

Die Aufgabenstatusleiste bietet eine Schätzung des Fortschritts der Aufgabe. Die Qualität dieser Schätzung hängt von der Qualität der Tabellenstatistik der Quelldatenbank ab. Je besser die Tabellenstatistik, desto genauer die Schätzung. Wenn eine Aufgabe nur eine Tabelle ohne geschätzte Zeilenstatistik enthält, können wir den prozentualen Fortschritt nicht berechnen. In diesem Fall kann anhand des Aufgabenstatus und der Angabe der geladenen Zeilen überprüft werden, ob die Aufgabe tatsächlich ausgeführt wird und Fortschritte macht.

Beachten Sie, dass die Spalte für „Letzte Aktualisierung“ der DMS-Konsole nur den Zeitpunkt anzeigt, an dem AWS DMS zuletzt den Tabellenstatistik-Datensatz für eine Tabelle aktualisiert hat. Es wird nicht der Zeitpunkt der letzten Tabellenaktualisierung angegeben.

Sie können nicht nur die DMS-Konsole verwenden, sondern auch eine Beschreibung der aktuellen Replikationsaufgaben, einschließlich des Aufgabenstatus, ausgeben, indem Sie den Befehl `aws dms describe-replication-tasks` in der [AWS CLI](#) verwenden, wie im folgenden Beispiel gezeigt.

```
{
  "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
      }
    }
  ]
}

```

Tabellenstatus während der Aufgaben

Während der Migration aktualisiert die AWS-DMS-Konsole Informationen zum Status Ihrer Tabellen. Die folgende Tabelle zeigt die möglichen Statuswerte:

DMS > Database migration tasks > dms-gs-task

dms-gs-task

Summary

Status: ⊘ Running with errors | Type: Full load, ongoing replication

Overview details | **Table statistics** | CloudWatch metrics | Mapping rules | Premigration assessments | Tags

Table statistics (157)
Total rows include loaded source table rows from Inserts, Deletes, Updates, DDLs, and Full load rows.

Find schema

<input type="checkbox"/>	Schema name	Table	Load state	Elapsed load time
<input type="checkbox"/>	mysql	user	Table error	< 1 s
<input type="checkbox"/>	mysql	server_cost	Table completed	< 1 s
<input type="checkbox"/>	mysql	tables_priv	Table completed	< 1 s
<input type="checkbox"/>	mysql	gtid_executed	Table completed	< 1 s
<input type="checkbox"/>	mysql	replication_asynchronous_connection_failover	Table completed	< 1 s

Status	Beschreibung
Table does not exist (Tabelle nicht vorhanden)	AWS DMS kann die Tabelle auf dem Quellendpunkt nicht finden.
Before load (Vor dem Laden)	Der vollständige Ladevorgang wurde aktiviert, hat aber noch nicht begonnen.
Full load	Der vollständige Ladevorgang wird ausgeführt.
Table completed	Der vollständige Ladevorgang ist abgeschlossen.
Table cancelled	Das Laden der Tabelle wurde abgebrochen.
Table error	Beim Laden der Tabelle ist ein Fehler aufgetreten.

Überwachen von Replikationsaufgaben mit Amazon CloudWatch

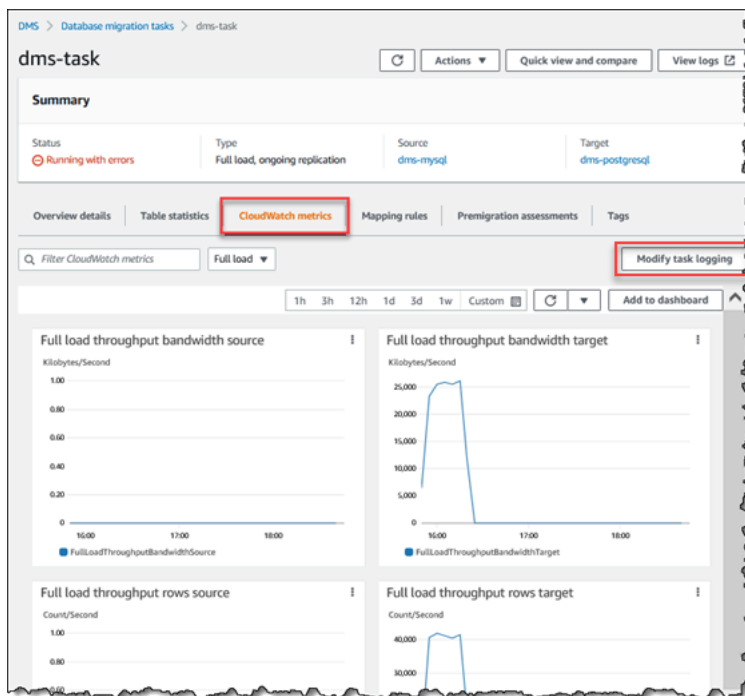
Sie können Amazon CloudWatch-Alarme oder -Ereignisse verwenden, um Ihre Migration noch genauer zu überwachen. Weitere Informationen zu Amazon CloudWatch finden Sie unter [Was sind Amazon CloudWatch, Amazon CloudWatch Events und Amazon CloudWatch Logs?](#) im Amazon-

CloudWatch-Benutzerhandbuch. Beachten Sie, dass für die Nutzung von Amazon CloudWatch Gebühren anfallen.

Wenn Ihre Replikationsaufgabe keine CloudWatch-Protokolle erstellt, finden Sie weitere Informationen unter [AWS DMS erstellt keine CloudWatch Protokolle](#) in der Anleitung zur Fehlerbehebung.

Die AWS DMS-Konsole zeigt einfache CloudWatch-Statistiken für jede Aufgabe an, darunter Aufgabenstatus, abgeschlossener Prozentsatz, verstrichene Zeit und Tabellenstatistiken, wie im Folgenden dargestellt. Wählen Sie die Replikationsaufgabe und dann die Registerkarte CloudWatch-Metriken aus.

Um die Einstellungen für das CloudWatch-Aufgabenprotokoll anzuzeigen und zu ändern, wählen Sie Aufgabenprotokollierung ändern. Weitere Informationen finden Sie unter [Einstellungen für das Protokollieren von Aufgaben](#).



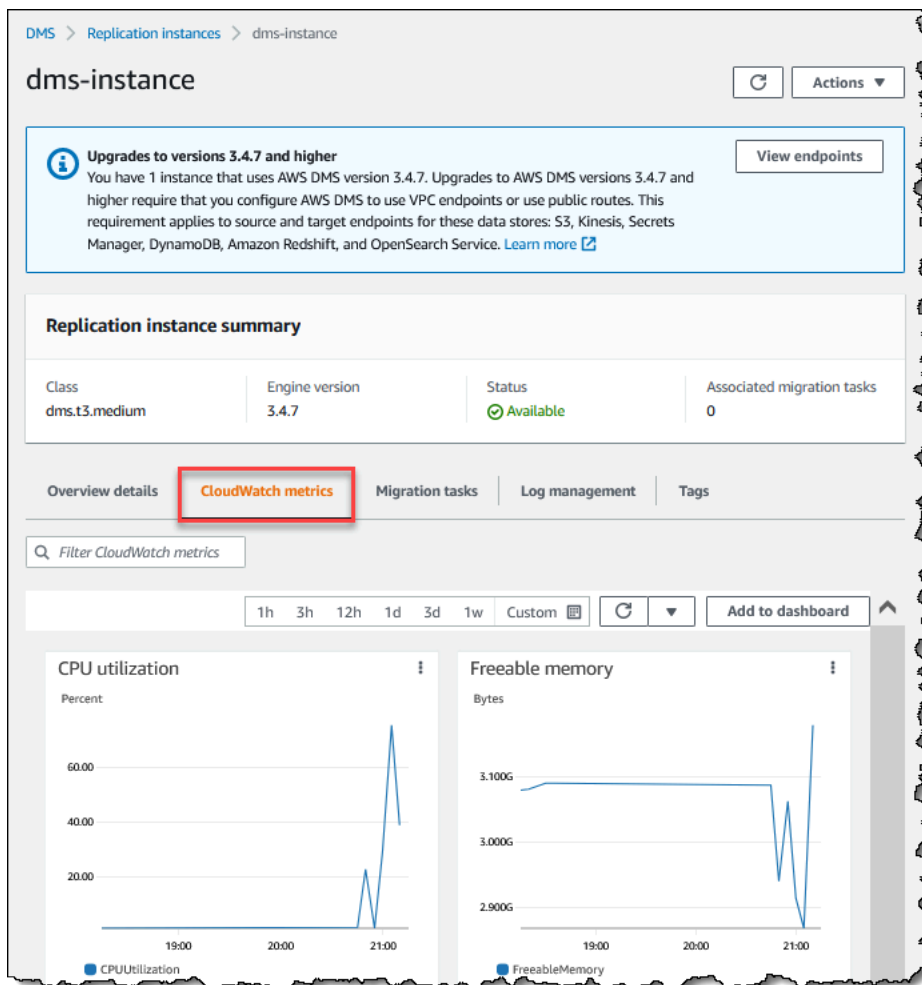
Die AWS-DMS-Konsole zeigt für jede Tabelle eine Leistungsstatistik an, darunter die Anzahl der Einfüge-, Lösch- und Aktualisierungsvorgänge. Wählen Sie dazu die Registerkarte Tabellenstatistik aus.

Table statistics (173)
Total rows include loaded source table rows from Inserts, Deletes, Updates, DDLs, and Full load rows.

Find schema

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

Wenn Sie zudem eine Replikations-Instance auf der Seite Replikations-Instance auswählen, können Sie Leistungskennzahlen für die Instance anzeigen, wenn Sie die Registerkarte CloudWatch-Metriken auswählen.



AWS Database Migration Service-Metriken

AWS DMS stellt folgende Statistiken bereit:

- Host-Metriken – Leistungs- und Nutzungsstatistiken für den Replikations-Host, bereitgestellt von Amazon CloudWatch. Eine vollständige Liste der verfügbaren Metriken finden Sie unter [Metriken zur Replikations-Instance](#).
- Metriken für die Replikationsaufgabe – Statistiken für Replikationsaufgaben, z. B. eingehende und festgeschriebene Änderungen, und die Latenz zwischen Replikations-Host sowie Quell- und Zieldatenbank. Eine vollständige Liste der verfügbaren Metriken finden Sie unter [Metriken für die Replikationsaufgabe](#).
- Tabellen-Metriken – Statistik für Tabellen, die aktuell migriert werden, darunter die Anzahl der Einfüge-, Lösch- und Aktualisierungsvorgänge sowie abgeschlossene DDL-Anweisungen.

Die Aufgabenmetriken werden in Statistiken zwischen Replikationshost und Quellendpunkt und zwischen Replikationshost und Zielpunkt untergliedert. Sie können die Gesamtstatistik für eine Aufgabe anzeigen, indem Sie zwei verwandte Statistiken zusammen hinzufügen. Sie können beispielsweise die Gesamtlatenzzeit oder die Replikatzögerung für eine Aufgabe bestimmen, indem Sie die Werte `CDCLatencySource` und `CDCLatencyTarget` kombinieren.

Die Metrikerwerte für Aufgaben können durch aktuelle Aktivitäten in Ihrer Quelldatenbank beeinflusst werden. Wenn z. B. eine Transaktion begonnen hat, aber noch nicht festgeschrieben wurde, erhöht sich die Metrik `CDCLatencySource`, bis die Transaktion festgeschrieben ist.

Bei der Replikations-Instance erfordert die Metrik `FreeableMemory` eine Klärung. Der freisetzbare Arbeitsspeicher gibt keinen Hinweis auf den tatsächlich verfügbaren Speicher. Es handelt sich dabei um den derzeit verwendeten Arbeitsspeicher, der freigegeben und für andere Zwecke eingesetzt werden. Es ist eine Kombination aus aktuell verwendeten Puffern und dem Zwischenspeicher auf der Replikations-Instance.

Während die `FreeableMemory`-Metrik nicht den tatsächlich verfügbaren Arbeitsspeicher wiedergibt, gibt die Kombination der Metriken `FreeableMemory` und `SwapUsage` an, ob die Replikations-Instance überlastet ist.

Überwachen Sie diese beiden Metriken auf die folgenden Bedingungen:

- Die Metrik `FreeableMemory` nähert sich Null.
- Die Metrik `SwapUsage` steigt oder schwankt.

Ist eine dieser Bedingungen erfüllt, ist dies ein Hinweis darauf, dass Sie eine größere Replikations-Instance verwenden sollten. Sie sollten auch die Reduzierung der Anzahl und Art der Aufgaben in

Betracht ziehen, die auf der Replikations-Instance ausgeführt werden. Full Load-Aufgaben benötigen mehr Speicher als Aufgaben, die nur Änderungen replizieren.

Um den tatsächlichen Speicherbedarf für eine AWS DMS-Migrationsaufgabe grob abzuschätzen, können Sie die folgenden Parameter verwenden.

LOB-Spalten

Eine durchschnittliche Anzahl von LOB-Spalten in jeder Tabelle in Ihrem Migrationsbereich.

Maximum number of tables to load in parallel (Maximale Anzahl der parallel zu ladenden Tabellen)

Die maximale Anzahl von Tabellen, die in einer Aufgabe parallel von AWS DMS geladen werden.

Der Standardwert ist 8.

LOB-Block-Größe

Die Größe der LOB-Blöcke in Kilobyte, die AWS DMS zum Replizieren von Daten in die Zieldatenbank verwendet.

Commit-Rate bei Volllast

Die maximale Anzahl von Datensätzen, die AWS DMS parallel übertragen kann.

Der Standardwert lautet 10.000.

LOB-Größe

Die maximale Größe eines einzelnen LOB in Kilobyte.

Massen-Array-Größe

Die maximale Anzahl von Zeilen, die von Ihrem Endpunkttreiber abgerufen oder verarbeitet werden. Dieser Wert hängt von den Treibereinstellungen ab.

Der Standardwert lautet 1.000.

Nachdem Sie diese Werte ermittelt haben, können Sie mit einer der folgenden Methoden abschätzen, wie viel Arbeitsspeicher für Ihre Migrationsaufgabe benötigt wird. Diese Methoden hängen von der Option ab, die Sie in Ihrer Migrationsaufgabe für die LOB-Spalteneinstellungen wählen.

- Verwenden Sie für den vollständigen LOB-Modus die folgende Formel.

$$\text{Required memory} = (\text{LOB columns}) * (\text{Maximum number of tables to load in parallel}) * (\text{LOB chunk size}) * (\text{Commit rate during full load})$$

Nehmen Sie ein Beispiel, bei dem Ihre Quelltabellen durchschnittlich 2 LOB-Spalten enthalten und die Größe der LOB-Blöcke 64 KB beträgt. Wenn Sie die Standardwerte für `Maximum number of tables to load in parallel` `Commit rate during full load` und verwenden, ist die Menge an benötigtem Speicher für Ihre Aufgabe wie folgt.

$$\text{Required memory} = 2 * 8 * 64 * 10,000 = 10,240,000 \text{ KB}$$

Note

Um den Wert der Commit-Rate bei Volllast zu reduzieren, öffnen Sie die AWS DMS-Konsole, wählen Sie Datenbankmigrationsaufgaben und erstellen oder ändern Sie eine Aufgabe. Erweitern Sie Erweiterte Einstellungen und geben Sie Ihren Wert für Commit-Rate bei Volllast ein.

- Verwenden Sie für den Eingeschränkten LOB-Modus die folgende Formel.

$$\text{Required memory} = (\text{LOB columns}) * (\text{Maximum number of tables to load in parallel}) * (\text{LOB size}) * (\text{Bulk array size})$$

Nehmen Sie ein Beispiel, bei dem Ihre Quelltabellen durchschnittlich 2 LOB-Spalten enthalten und die maximale Größe der einzelnen LOB-Blöcke 4 096 KB beträgt. Wenn Sie die Standardwerte für `Maximum number of tables to load in parallel` und `Bulk array size` verwenden, ist die Menge an benötigtem Speicher für Ihre Aufgabe wie folgt.

$$\text{Required memory} = 2 * 8 * 4,096 * 1,000 = 65,536,000 \text{ KB}$$

Damit AWS DMS Konvertierungen optimal durchführen kann, muss die CPU zum Zeitpunkt der Konvertierungen verfügbar sein. Eine Überlastung der CPU und ein Mangel an CPU-Ressourcen können zu langsamen Migrationen führen. AWS DMS kann CPU-intensiv sein, insbesondere bei heterogenen Migrationen und Replikationen wie der Migration von Oracle zu PostgreSQL. Die C4-Replikations-Instance-Klasse kann eine gute Wahl für solche Situationen sein. Weitere Informationen finden Sie unter [Auswahl der richtigen AWS DMS-Replikationsinstanz für Ihre Migration](#).

Metriken zur Replikations-Instance

Die Überwachung der Replikations-Instance beinhaltet Amazon-CloudWatch-Metriken für die folgenden Statistiken:

Metrik	Beschreibung
AvailableMemory	<p>Eine Schätzung, wie viel Speicher für den Start neuer Anwendungen ohne Auslagern zur Verfügung steht. Weitere Informationen finden Sie unter MemAvailable im /proc/memInfo -Abschnitt der Linux-Handbuchseiten.</p> <p>Einheiten: Byte</p>
CPUAllocated	<p>Der Prozentsatz der CPU, der der Aufgabe maximal zugewiesen ist (0 bedeutet kein Limit).</p> <p>AWS DMS berechnet diese Metrik anhand der kombinierten Dimensionen von ReplicationInstanceIdentifier und ReplicationTaskIdentifier in der CloudWatch-Konsole. Verwenden Sie die ReplicationInstanceIdentifier, ReplicationTaskIdentifier -Kategorie, um diese Metrik anzuzeigen.</p> <p>Einheiten: Prozent</p>
CPUUtilization	<p>Prozentsatz der zugeordneten vCPUs (virtuellen CPUs), die gegenwärtig auf der Instance verwendet werden.</p> <p>Einheiten: Prozent</p>
DiskQueueDepth	<p>Anzahl der offenstehenden Lese-/Schreibanforderungen (E/As), die auf Festplattenzugriff warten.</p> <p>Einheiten: Anzahl</p>
FreeStorageSpace	<p>Verfügbarer Speicherplatz</p> <p>Einheiten: Byte</p>
FreeMemory	<p>Die Menge an physischem Speicher, die für Anwendungen, den Seitencache und für die eigenen Datenstrukturen des Kerns verfügbar ist. Weitere Informationen finden Sie unter MemFree im /proc/memInfo -Abschnitt der Linux-Handbuchseiten.</p> <p>Einheiten: Byte</p>

Metrik	Beschreibung
FreeableMemory	<p>Verfügbarer Arbeitsspeicher.</p> <p>Einheiten: Byte</p>
MemoryAllocated	<p>Die maximale Speicherzuweisung für die Aufgabe (0 bedeutet keine Einschränkungen).</p> <p>AWS DMS berechnet diese Metrik anhand der kombinierten Dimensionen von <code>ReplicationInstanceIdentifier</code> und <code>ReplicationTaskIdentifier</code> in der CloudWatch-Konsole. Verwenden Sie die <code>ReplicationInstanceIdentifier</code>, <code>ReplicationTaskIdentifier</code>-Kategorie, um diese Metrik anzuzeigen.</p> <p>Einheiten: MiB</p>
WriteIOPS	<p>Durchschnittliche Anzahl von Festplatten-I/O-Schreibvorgänge pro Sekunde.</p> <p>Einheiten: Anzahl/Sekunde</p>
ReadIOPS	<p>Durchschnittliche Anzahl der Festplatten-I/O-Lesevorgänge pro Sekunde.</p> <p>Einheiten: Anzahl/Sekunde</p>
WriteThroughput	<p>Die durchschnittliche Anzahl von Bytes, die pro Sekunde auf den Datenträger geschrieben werden.</p> <p>Einheiten: Byte/Sekunde</p>
ReadThroughput	<p>Die durchschnittliche Anzahl Byte, die pro Sekunde vom Datenträger gelesen werden.</p> <p>Einheiten: Byte/Sekunde</p>
WriteLatency	<p>Die durchschnittliche Dauer für einen Festplatten-E/A(Ausgabe)-Vorgang.</p> <p>Einheiten: Millisekunden</p>

Metrik	Beschreibung
ReadLatency	Die durchschnittliche Dauer für einen Festplatten-E/A(Eingabe)-Vorgang. Einheiten: Millisekunden
SwapUsage	Die Menge des für die Replikations-Instance verwendeten Auslagerungsbereichs Einheiten: Byte
NetworkTransmitThroughput	Der ausgehende Netzwerkverkehr (Transmit) auf der Replikations-Instance, einschließlich Kundendatenbankverkehr und AWS DMS-Datenverkehr, der für Überwachung und Replikation verwendet wird. Einheiten: Byte/Sekunde
NetworkReceiveThroughput	Der eingehende Netzwerkverkehr (Receive) auf der Replikations-Instance, einschließlich Kundendatenbankverkehr und AWS DMS-Datenverkehr, der für Überwachung und Replikation verwendet wird. Einheiten: Byte/Sekunde

Metriken für die Replikationsaufgabe

Die Überwachung von Replikationsaufgaben beinhaltet Metriken für die folgenden Statistiken:

Metrik	Beschreibung
FullLoadThroughput BandwidthTarget	Für das Ziel übertragene ausgehende Daten von Volllast in KB pro Sekunde.
FullLoadThroughput RowsTarget	Ausgehende Änderungen von einem vollständigen Ladevorgang für das Ziel in Zeilen pro Sekunde
CDCIncomingChanges	Die Gesamtanzahl der Änderungsereignisse zu einem Zeitpunkt, die darauf warten, auf das Ziel angewendet zu werden. Beachten Sie, dass dies nicht dasselbe ist wie eine Messung der Transaktionsänderu

Metrik	Beschreibung
	ngsrate am Quellendpunkt. Eine hohe Zahl für diese Metrik weist in der Regel darauf hin, dass AWS DMS nicht in der Lage ist, erfasste Änderungen zeitgerecht anzuwenden, was zu einer hohen Ziellatenz führt.
CDCChange sMemorySource	Anzahl der Zeilen, die sich in einem Arbeitsspeicher angesammelt haben und auf Festschreibung von der Quelle warten Sie können diese Metrik zusammen mit CDCChangesDiskSource anzeigen.
CDCChange sMemoryTarget	Anzahl der Zeilen, die sich in einem Arbeitsspeicher angesammelt haben und auf Festschreibung im Ziel warten Sie können diese Metrik zusammen mit CDCChangesDiskTarget anzeigen.
CDCChangesDiskSou rce	Anzahl der Zeilen, die sich auf einem Datenträger angesammelt haben und auf Festschreibung von der Quelle warten Sie können diese Metrik zusammen mit CDCChangesMemorySource anzeigen.
CDCChangesDiskTarg et	Anzahl der Zeilen, die sich auf einem Datenträger angesammelt haben und auf Festschreibung im Ziel warten Sie können diese Metrik zusammen mit CDCChangesMemoryTarget anzeigen.
CDCThroughputBandw idthTarget	Für das Ziel übertragene ausgehende Daten in KB pro Sekunde. CDCThroughputBandwidth zeichnet ausgehende Daten auf, die an Stichprobenpunkten übertragen werden. Wenn kein Netzwerkverkehr für die Aufgabe gefunden wird, ist der Wert Null. Da CDC keine Langläufer-Transaktionen ausgibt, wird der Netzwerkdatenverkehr u. U. nicht aufgezeichnet.
CDCThroughputRowsS ource	Eingehende Aufgabenänderungen von der Quelle in Zeilen pro Sekunde
CDCThroughputRowsT arget	Ausgehende Aufgabenänderungen für das Ziel in Zeilen pro Sekunde

Metrik	Beschreibung
CDCLatencySource	<p>Die Anzahl der Sekunden zwischen dem zuletzt erfassten Ereignis vom Quellendpunkt und dem aktuellen Systemzeitstempel der AWS DMS-Instance. CDCLatencySource steht für die Latenz zwischen Quelle und Replikations-Instance. Ein hoher CDCLatencySource-Wert bedeutet, dass der Prozess der Erfassung von Änderungen aus der Quelle verzögert wird. Um die Latenz bei einer laufenden Replikation zu ermitteln, können Sie sich diese Metrik zusammen mit CDCLatencyTarget anzeigen lassen. Wenn sowohl CDCLatencySource als auch CDCLatencyTarget hoch sind, untersuchen Sie zuerst CDCLatencySource.</p> <p>CDCSourceLatency kann 0 sein, wenn keine Replikationsverzögerung zwischen der Quelle und der Replikations-Instance besteht. CDCSourceLatency kann auch den Wert Null haben, wenn die Replikation versucht, das nächste Ereignis im Transaktionsprotokoll der Quelle zu lesen, und es keine neuen Ereignisse im Vergleich zum letzten Lesen aus der Quelle gibt. Wenn dies geschieht, setzt die Aufgabe CDCSourceLatency auf 0 zurück.</p>

Metrik	Beschreibung
CDCLatencyTarget	<p>Die Anzahl der Sekunden zwischen dem ersten Ereigniszeitstempel, für den auf dem Ziel ein Commit ausgeführt werden soll, und dem aktuellen Zeitstempel der AWS DMS-Instance. Die Ziellatenz ist der Unterschied zwischen der Serverzeit der Replikations-Instance und der ältesten unbestätigten Ereignis-ID, die an eine Zielkomponente weitergeleitet wurde. Mit anderen Worten, die Ziellatenz ist der Zeitstempelunterschied zwischen der Replikations-Instance und dem ältesten Ereignis, das angewendet, aber vom TRG-Endpunkt nicht bestätigt wurde (99 %). Wenn CDCLatencyTarget hoch ist, bedeutet dies, dass der Prozess der Anwendung von Änderungsereignissen auf das Ziel verzögert ist. Um die Latenz bei einer laufenden Replikation zu ermitteln, können Sie sich diese Metrik zusammen mit CDCLatencySource anzeigen lassen. Wenn CDCLatencyTarget hoch, aber CDCLatencySource nicht hoch ist, untersuchen Sie Folgendes:</p> <ul style="list-style-type: none">• Das Ziel enthält keine Primärschlüssel oder Indizes• In der Ziel- oder Replikations-Instance treten Ressourcenengpässe auf• Netzwerkprobleme treten zwischen der Replikations-Instance und dem Ziel auf
CPUUtilization	<p>Der Prozentsatz der CPU, der von einer Aufgabe über mehrere Kerne hinweg genutzt wird. Die Semantik der CPUUtilization von Aufgaben unterscheidet sich geringfügig von der CPUUtilization bei der Replikation. Wenn 1 vCPU vollständig genutzt wird, bedeutet dies 100 %. Wenn jedoch mehrere vCPUs genutzt werden, kann der Wert über 100 % liegen.</p> <p>Einheiten: Prozent</p>
SwapUsage	<p>Größe des von der Aufgabe belegten Auslagerungsspeichers.</p> <p>Einheiten: Byte</p>

Metrik	Beschreibung
MemoryUsage	<p>Die Kontrollgruppe (Cgroup) <code>memory.usage_in_bytes</code>, die von einer Aufgabe genutzt wird. DMS verwendet cgroups, um die Nutzung von Systemressourcen wie Speicher und CPU zu kontrollieren. Diese Metrik gibt die Speichernutzung einer Aufgabe in Megabyte innerhalb der für diese Aufgabe zugewiesenen cgroup an. Die cgroup-Grenzwerte basieren auf den Ressourcen, die für Ihre DMS-Replikations-Instance-Klasse verfügbar sind. <code>memory.usage_in_bytes</code> besteht aus den Komponenten Resident Set Size (RSS), Cache und Swap-Speicher. Das Betriebssystem kann bei Bedarf Cache-Speicher zurückfordern. Wir empfehlen, dass Sie auch die Replikations-Instance-Metrik <code>AvailableMemory</code> überwachen.</p> <p>AWS DMS berechnet diese Metrik anhand der kombinierten Dimensionen von <code>ReplicationInstanceIdentifier</code> und <code>ReplicationTaskIdentifier</code> in der CloudWatch-Konsole. Verwenden Sie die <code>ReplicationInstanceIdentifier</code>, <code>ReplicationTaskIdentifier</code>-Kategorie, um diese Metrik anzuzeigen.</p>

Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen

Sie können Amazon CloudWatch verwenden, um Aufgabeninformationen während eines AWS DMS-Migrationsprozesses zu protokollieren. Sie aktivieren die Protokollierung während der Auswahl der Aufgabeneinstellungen. Weitere Informationen finden Sie unter [Einstellungen für das Protokollieren von Aufgaben](#).

Gehen Sie wie folgt vor, um Protokolle einer Aufgabe anzuzeigen, die ausgeführt wurde:

1. Öffnen Sie die AWS DMS-Konsole. Wählen Sie im Navigationsbereich die Option `Database migration tasks` (Datenbankmigrationsaufgaben) aus. Das Dialogfeld „Database migration tasks (Datenbankmigrationsaufgaben)“ wird angezeigt.
2. Wählen Sie den Namen der Aufgabe aus. Das Dialogfeld „Overview details (Übersichtsdetails)“ wird angezeigt.
3. Suchen Sie den Abschnitt `Migration task logs` (Migrationsaufgabenprotokolle) und wählen Sie `View CloudWatch Logs` (CloudWatch-Protokolle anzeigen) aus.

Zusätzlich können Sie die AWS CLI oder die AWS DMS-API verwenden, um Informationen zu Aufgabenprotokollen anzuzeigen. Verwenden Sie hierzu den Befehl `describe-replication-instance-task-logs` AWS CLI oder die AWS DMS-API-Aktion `DescribeReplicationInstanceTaskLogs`.

Der folgende AWS CLI-Befehl zeigt beispielsweise die Metadaten des Aufgabenprotokolls im JSON-Format an.

```
$ aws dms describe-replication-instance-task-logs \  
  --replication-instance-arn arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAIY
```

Eine Beispielantwort des Befehls lautet wie folgt.

```
{  
  "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"  
}
```

In dieser Antwort gibt es ein einzelnes Aufgabenprotokoll (`mysql-to-ddb`), das mit der Replikations-Instance verknüpft ist. Die Größe dieser Protokolldatei beträgt 3.726.124 Bytes.

Sie können die von `describe-replication-instance-task-logs` zurückgegebenen Informationen verwenden, um Probleme mit Aufgabenprotokollen zu diagnostizieren und zu beheben. Wenn Sie z. B. die detaillierte Debug-Protokollierung für eine Aufgabe aktivieren, wird die Größe der Aufgabenprotokolldatei rasch zunehmen – möglicherweise wird der gesamte verfügbare Speicherplatz auf der Replikations-Instance benötigt und der Status der Instance ändert sich zu `storage-full`. Durch Beschreiben der Aufgabenprotokolle können Sie feststellen, welche nicht mehr benötigt werden. Diese können Sie dann löschen und so Speicherplatz freigeben.

Um die Aufgabenprotokolle einer Aufgabe zu löschen, setzen Sie die Aufgabeneinstellung `DeleteTaskLogs` auf `true`. Beispielsweise löscht das folgende JSON die Aufgabenprotokolle, wenn eine Aufgabe mit dem AWS CLI `modify-replication-task`-Befehl oder der AWS DMS-API `ModifyReplicationTask` modifiziert wird.

```
{
  "Logging": {
    "DeleteTaskLogs": true
  }
}
```

Protokollierung von AWS DMS-API-Aufrufen mit AWS CloudTrail

AWS DMS ist in AWS CloudTrail integriert, einen Service, der die Aktionen eines Benutzers, einer Rolle oder eines AWS-Service in AWS DMS protokolliert. CloudTrail erfasst alle API-Aufrufe für AWS DMS als Ereignisse, einschließlich Aufrufen von der AWS DMS-Konsole und von Code-Aufrufen an die AWS DMS-API-Vorgänge. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon S3-Bucket, einschließlich Ereignisse für AWS DMS aktivieren. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole trotzdem in Event history (Ereignisverlauf) anzeigen. Mit den von CloudTrail erfassten Informationen können Sie die an AWS DMS gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

AWS DMS-Informationen in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Kontos für Sie aktiviert. Die in AWS DMS auftretenden Aktivitäten werden als CloudTrail-Ereignis zusammen mit anderen AWS-Serviceereignissen in Event History (Ereignisverlauf) aufgezeichnet. Sie können die neuesten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail-API-Ereignisverlauf](#).

Zur kontinuierlichen Aufzeichnung von Ereignissen in Ihrem AWS-Konto, einschließlich Ereignissen für AWS DMS, erstellen Sie einen Trail. Ein Trail ermöglicht es CloudTrail, Protokolldateien in einem

Amazon-S3-Bucket bereitzustellen. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen AWS-Regionen in der AWS-Partition und stellt die Protokolldateien in dem Amazon-S3-Bucket bereit, den Sie angeben. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- [Übersicht zum Erstellen eines Trails](#)
- [In CloudTrail unterstützte Services und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail-Protokolldateien aus mehreren AWS-Regionen und Empfangen von CloudTrail-Protokolldateien aus mehreren Konten](#)

Alle AWS DMS-Aktionen werden von CloudTrail protokolliert und sind in der [AWS Database Migration Service-API-Referenz](#) dokumentiert. Zum Beispiel generieren Aufrufe der Aktionen `CreateReplicationInstance`, `TestConnection` und `StartReplicationTask` Einträge in den CloudTrail-Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anfrage mit Root- oder IAM-Benutzer-Anmeldeinformationen von ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer gesendet wurde.
- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter [CloudTrail-Element `userIdentity`](#).

Grundlagen zu AWS DMS-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der die Aktion `RebootReplicationInstance` demonstriert.

```
{
  "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 DMS-Kontext-Protokollierung

AWS DMS verwendet die Kontextprotokollierung, um Ihnen Informationen über eine laufende Migration zu geben. Die Kontextprotokollierung schreibt Informationen wie die folgenden in das CloudWatch-Protokoll der Aufgabe:

- Informationen zur Verbindung der Aufgabe mit der Quell- und der Zieldatenbank.
- Verhalten der Replikationsaufgabe. Sie können die Aufgabenprotokolle verwenden, um Replikationsprobleme zu diagnostizieren.
- SQL-Anweisungen ohne Daten, die AWS DMS für Quell- und Zieldatenbanken ausführt. Sie können die SQL-Protokolle verwenden, um unerwartetes Migrationsverhalten zu diagnostizieren.
- Stream-Positionsdetails für jedes CDC-Ereignis.

Die Kontextprotokollierung ist nur in AWS DMS Version 3.5.0 oder höher verfügbar.

AWS DMS aktiviert standardmäßig die Kontextprotokollierung. Um die Kontextprotokollierung zu steuern, setzen Sie die `EnableLogContext`-Aufgabeneinstellung auf `true` oder `false`, indem Sie die Aufgabe in der Konsole ändern.

AWS DMS schreibt alle drei Minuten Kontextprotokollinformationen in die Replikationsaufgabe des CloudWatch-Protokolls. Stellen Sie sicher, dass Ihre Replikations-Instance über ausreichend Speicherplatz für ihr Anwendungsprotokoll verfügt. Weitere Hinweise zur Verwaltung von Aufgabenprotokollen finden Sie unter [Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen](#).

Themen

- [Objekttypen](#)
- [Beispiele für die Protokollierung](#)
- [Einschränkungen](#)

Objekttypen

AWS DMS erzeugt Kontextprotokollierung in CloudWatch für die folgenden Objekttypen.

Objekttyp	Beschreibung
TABLE_NAME	Diese Protokolleinträge enthalten Informationen zu Tabellen, die in den Geltungsbereich der aktuellen Aufgabenzuordnungsregel fallen. Sie können diese Einträge verwenden, um die Tabellenergebnisse für einen bestimmten Zeitraum während der Migration zu untersuchen.
SCHEMA_NAME	Diese Protokolleinträge enthalten Informationen zu Schemata, die von der aktuellen Aufgabenzuordnungsregel verwendet werden. Anhand dieser Einträge können Sie ermitteln, welches Schema AWS DMS für einen bestimmten Zeitraum während der Migration verwendet.
TRANSACTION_ID	Diese Einträge enthalten die Transaktions-ID für jede DML/DDL-Änderung, die aus der Quelldatenbank erfasst wurde. Sie können diese Protokolleinträge verwenden, um festzustellen, welche Änderungen während einer bestimmten Transaktion vorgenommen wurden.
CONNECTION_ID	Diese Einträge enthalten die Verbindungs-ID. Sie können diese Protokolleinträge verwenden, um zu ermitteln, welche Verbindung für jeden Migrationsschritt AWS DMS verwendet.
STATEMENT	Diese Einträge enthalten den SQL-Code, der zum Abrufen, Verarbeiten und Anwenden der einzelnen Migrationsänderungen verwendet wird.
STREAM_POSITION	Diese Einträge enthalten die Position in der Transaktionsprotokolldatei für jede Migrationssaktion in der Quelldatenbank. Das Format

Objekttyp	Beschreibung
	für diese Einträge variiert je nach Typ der Quelldatenbank-Engine. Sie können diese Informationen auch verwenden, um eine Startposition für einen Wiederherstellungs-Checkpoint zu bestimmen, wenn Sie die reine CDC-Replikation konfigurieren.

Beispiele für die Protokollierung

Dieser Abschnitt enthält Beispiele für Protokolldatensätze, die Sie zur Überwachung der Replikation und zur Diagnose von Replikationsproblemen verwenden können.

Verbindungsprotokollbeispiele

Dieser Abschnitt enthält Protokollbeispiele, die Verbindungs-IDs enthalten.

```
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)
```

Beispiele für Protokolle zum Verhalten von Aufgaben

Dieser Abschnitt enthält Protokollbeispiele zum Verhalten von Protokollen bei Replikationsaufgaben. Sie können diese Informationen verwenden, um Replikationsprobleme zu diagnostizieren, z. B. wenn sich eine Aufgabe im Status IDLE befindet.

Die folgenden SOURCE_CAPTURE-Protokolle weisen darauf hin, dass keine Ereignisse verfügbar sind, die aus der Protokolldatei der Quelldatenbank gelesen werden können. Sie enthalten TARGET_APPLY-Datensätze, die darauf hinweisen, dass keine Ereignisse von AWS DMS-CDC-

Komponenten empfangen wurden, die auf die Zieldatenbank angewendet werden könnten. Diese Ereignisse enthalten auch zuvor angewendete ereignisbezogene Kontextdetails.

```
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
(?,?,?,?/?),
```

SQL-Anweisungsprotokollbeispiele

Dieser Abschnitt enthält Protokollbeispiele zu SQL-Anweisungen, die in Quell- und Zieldatenbanken ausgeführt werden. Die SQL-Anweisungen, die Sie in den Protokollen sehen, zeigen nur die SQL-Anweisung, nicht die Daten. Das folgende TARGET_APPLY-Protokoll zeigt eine INSERT-Anweisung, die auf dem Ziel ausgeführt wurde.

```
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
(?,?,?,?/?),
```

Einschränkungen

Die folgenden Einschränkungen gelten für die AWS DMS-Kontextprotokollierung:

- Zwar AWS DMS erstellt eine minimale Protokollierung für alle Endpunkttypen, eine umfangreiche modulspezifische Kontextprotokollierung ist jedoch nur für die folgenden Endpunkttypen verfügbar. Wir empfehlen, die Kontextprotokollierung zu aktivieren, wenn Sie diese Endpunkttypen verwenden.
 - MySQL
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server
 - MongoDB/Amazon DocumentDB

- Amazon S3

Arbeiten mit Amazon-Event-Bridge-Ereignissen und Benachrichtigungen in AWS Database Migration Service

Sie können Amazon EventBridge zum Bereitstellen von Benachrichtigungen verwenden, wenn ein AWS DMS-Ereignis auftritt, z. B. beim Erstellen oder Löschen einer Replikations-Instance. EventBridge empfängt Ereignisse und leitet Benachrichtigungen über ein Ereignis weiter, wie in den Ereignisregeln definiert. Sie können mit diesen Benachrichtigungen in jeder Form arbeiten, die von Amazon EventBridge für eine AWS-Region unterstützt wird. Weitere Informationen zur Verwendung von Amazon EventBridge finden Sie unter [Was ist Amazon EventBridge?](#) im Benutzerhandbuch für Amazon EventBridge.

Note

Die Arbeit mit Amazon-EventBridge-Ereignissen wird in AWS DMS Version 3.4.5 und höher unterstützt.

EventBridge empfängt ein Ereignis, einen Indikator für eine Änderung in der AWS DMS-Umgebung, und wendet eine Regel an, um das Ereignis an einen Benachrichtigungsmechanismus weiterzuleiten. Regeln ordnen Ereignisse anhand der Struktur des Ereignisses, des so genannten Ereignismusters, den Benachrichtigungsmechanismen zu.

AWS DMS stuft Ereignisse in Kategorien ein, auf die Sie eine Ereignisregel anwenden können. Dadurch werden Sie benachrichtigt, wenn ein Ereignis in dieser Kategorie auftritt. Nehmen wir beispielsweise an, Sie wenden eine EventBridge-Ereignisregel auf die Kategorie „Erstellung“ für eine bestimmte Replikations-Instance an. Sie werden dann benachrichtigt, wenn ein Ereignis im Zusammenhang mit einer Erstellung eintritt, das sich auf Ihre Replikations-Instance auswirkt. Wenn Sie die Kategorie „Konfigurationsänderung“ für eine Replikations-Instance abonnieren, werden Sie benachrichtigt, wenn die Konfiguration der Replikations-Instance geändert wird. Eine Liste der von AWS DMS bereitgestellten Ereigniskategorien finden Sie im Folgenden unter AWS DMS-Ereigniskategorien und Ereignismeldungen.

Note

Stellen Sie sicher, dass Sie die Zugriffsrichtlinien Ihrer Amazon- SNS-Themen aktualisieren, um die Veröffentlichung von events.amazonaws.com zu ermöglichen. Weitere Informationen

finden Sie unter [Verwenden ressourcenbasierter Richtlinien für Amazon EventBridge](#) im Amazon-EventBridge-Benutzerhandbuch.

Weitere Informationen zum Verschieben von Ereignis-Abonnements zu Amazon EventBridge finden Sie unter [Migrieren aktiver Ereignis-Abonnements von DMS zu Amazon EventBridge](#).

Weitere Informationen zum Verwenden von SMS-Nachrichten mit Amazon SNS finden Sie unter [Senden und Empfangen von SMS-Benachrichtigungen mit Amazon SNS](#).

Verwenden von Amazon-EventBridge-Ereignisregeln für AWS DMS

Amazon EventBridge sendet Ereignisbenachrichtigungen an die Adressen, die Sie angeben, wenn Sie eine EventBridge-Ereignisregel erstellen. Möglicherweise möchten Sie mehrere verschiedene Regeln erstellen. Sie könnten beispielsweise eine Regel erstellen, die alle Ereignisbenachrichtigungen empfängt, und eine andere, die nur kritische Ereignisse für Ihre DMS-Produktionsressourcen enthält. Sie können in EventBridge auch Ereignisbenachrichtigungen aktivieren oder deaktivieren.

Erstellen von Amazon-EventBridge-Regeln, die auf AWS DMS-Ereignisse reagieren

- Führen Sie die unter [Erstellen von Amazon-EventBridge-Regeln, die auf Ereignisse reagieren](#) im Amazon-EventBridge-Benutzerhandbuch beschriebenen Schritte aus und erstellen Sie eine Regel für AWS DMS-Ereignisse:
 - a. Geben Sie eine Benachrichtigungsaktion an, die ausgeführt werden soll, wenn EventBridge ein Ereignis erhält, das dem Ereignismuster in der Regel entspricht. Wenn ein Ereignis übereinstimmt, sendet EventBridge das Ereignis an das angegebene Ziel und löst die in der Regel definierte Aktion aus.
 - b. Wählen Sie für Service provider (Serviceanbieter) die Option AWS aus.
 - c. Wählen Sie unter Servicename die Option Database Migration Service (SMS) aus.

Sie erhalten dann Ereignisbenachrichtigungen.

Das folgende JSON-Beispiel zeigt ein EventBridge-Ereignismodell für einen AWS DMS-Service.

```
{
  "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:AAAABBBB0CCCCDDDDDEEEEEE1FFFF2GGG3FFFFFFF3"
],
"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"
}
}
```

Die Liste der Kategorien und Ereignisse, über die Sie benachrichtigt werden können, finden Sie im folgenden Abschnitt.

AWS DMS-Ereigniskategorien und -Ereignismeldungen

AWS DMS generiert eine erhebliche Anzahl von Ereignissen in Kategorien, die Sie identifizieren können. Jede Kategorie gilt für eine Replikations-Instance oder für einen Replikationsaufgaben-Quellentyp.

Themen

- [ReplicationInstance-Ereignismeldungen](#)
- [ReplicationTask-Ereignismeldungen](#)
- [Replikations-Ereignismeldungen](#)

ReplicationInstance-Ereignismeldungen

In der folgenden Tabelle sind die möglichen Kategorien und Ereignisse für den Quelltyp ReplicationInstance angegeben.

Kategorie	Ereignis-ID	Beschreibung
Erstellung	DMS-EVENT-0067	Eine Replikations-Instance wird erstellt.
Löschung	DMS-EVENT-0066	Die Replikations-Instance wird gelöscht.
Konfigurationsänderung	DMS-EVENT-0012	Die Replikations-Instance-Klasse für diese Replikations-Instance wird geändert.
Konfigurationsänderung	DMS-EVENT-0018	Der Speicher für die Replikations-Instance wird erweitert.
Konfigurationsänderung	DMS-EVENT-0024	Die Replikations-Instance wird auf eine Multi-AZ-Konfiguration umgestellt.
Konfigurationsänderung	DMS-EVENT-0030	Die Replikations-Instance wird auf eine Single-AZ-Konfiguration umgestellt.
Wartung	DMS-EVENT-0026	Die Offline-Wartung der Replikations-Instance wird durchgeführt. Die Replikations-Instance steht derzeit nicht zur Verfügung.
Erstellung	DMS-EVENT-0005	Eine Replikations-Instance wurde erstellt.
Löschung	DMS-EVENT-0003	Die Replikations-Instance wurde gelöscht.
Konfigurationsänderung	DMS-EVENT-0014	Die Replikations-Instance-Klasse für diese Replikations-Instance wurde geändert.

Kategorie	Ereignis-ID	Beschreibung
Konfigurationsänderung	DMS-EVENT-0017	Der Speicher für die Replikations-Instanz wurde erweitert.
Konfigurationsänderung	DMS-EVENT-0025	Die Umstellung der Replikations-Instanz auf eine Multi-AZ-Konfiguration ist abgeschlossen.
Konfigurationsänderung	DMS-EVENT-0029	Die Umstellung der Replikations-Instanz auf eine Single-AZ-Konfiguration ist abgeschlossen.
Wartung	DMS-EVENT-0047	Die Managementsoftware auf der Replikations-Instanz wurde aktualisiert.
Wartung	DMS-EVENT-0027	Die Offline-Wartung der Replikations-Instanz ist abgeschlossen. Die Replikations-Instanz steht nun zur Verfügung.
Wartung	DMS-EVENT-0068	Die Replikations-Instanz befindet sich in einem Zustand, der nicht aktualisiert werden kann.
Failover	DMS-EVENT-0034	Wenn Sie zu häufig einen Failover anfordern, wird dieses Ereignis anstelle regelmäßiger Failover-Ereignisse ausgelöst.
Fehler	DMS-EVENT-0031	Die Replikations-Instanz wurde in den Status %s versetzt.
Fehler	DMS-EVENT-0036	Die Replikations-Instanz ist aufgrund eines nicht kompatiblen Netzwerks fehlgeschlagen.

Kategorie	Ereignis-ID	Beschreibung
Fehler	DMS-EVENT-0037	Wenn der Service nicht auf den zum Verschlüsseln der Datenmenge verwendeten KMS-Schlüssel zugreifen kann.
Fehler		Die Replikations-Instance verwendete inkompatible Parameter
Failover		Beim Warten auf einen sicheren Status, der den vom Benutzer angeforderten Failover einleitet, wurde eine Zeitüberschreitung festgestellt
Failover	DMS-EVENT-0013	Failover für eine Multi-AZ-Replikations-Instance wurde gestartet.
Failover	DMS-EVENT-0049	Failover für eine Multi-AZ-Replikations-Instance wurde abgeschlossen.
Failover	DMS-EVENT-0050	Die Multi-AZ-Aktivierung wurde gestartet.
Failover	DMS-EVENT-0051	Die Multi-AZ-Aktivierung ist abgeschlossen.
StateChange		Allgemeine Protokolle und Protokolle für langsame Abfragen wurden automatisch als %s rotiert

Kategorie	Ereignis-ID	Beschreibung
StateChange		AWS DMS kann nicht auf den KMS-Verschlüsselungsschlüssel für die Anwendungs-Instance %s zugreifen. Dies liegt wahrscheinlich daran, dass der Schlüssel deaktiviert ist oder AWS DMS nicht darauf zugreifen kann. Wenn dies so weitergeht, wird die Anwendung in den Zustand versetzt, dass nicht auf sie zugegriffen werden kann. Weitere Informationen finden Sie im Abschnitt zur Fehlerbehebung in der AWS DMS-Dokumentation.
StateChange		AWS DMS kann jetzt erfolgreich auf den KMS-Verschlüsselungsschlüssel für die Anwendungs-Instance %s zugreifen.
StateChange		Amazon DMS konnte nicht auf den KMS-Verschlüsselungsschlüssel für die Anwendungs-Instance %s zugreifen. Diese Anwendung wird in den Zustand versetzt, dass nicht auf sie zugegriffen werden kann. Weitere Informationen finden Sie im Abschnitt zur Fehlerbehebung in der Amazon-DMS-Dokumentation.
StateChange		Neustart der App auf HM als Teil der Erstellung der Replikations-Instance
StateChange		Herunterfahren der App auf HM als Teil des Löschens der Replikations-Instance

Kategorie	Ereignis-ID	Beschreibung
Failover	DMS-EVENT-0015	Multi-AZ-Failover zu Standby ist abgeschlossen.
LowStorage	DMS-EVENT-0007	Der verfügbare Speicherplatz für die Replikations-Instance ist gering.
LowStorage		Die zugewiesenen Inodes sind erschöpft. Skalieren Sie den Speicherplatz, um das Problem zu lösen.

ReplicationTask-Ereignismeldungen

In der folgenden Tabelle sind die möglichen Kategorien und Ereignisse für den Quelltyp Replikationsaufgabe angegeben.

Kategorie	Ereignis-ID	Beschreibung
Fehler	DMS-EVENT-0078	Eine Replikationsaufgabe ist fehlgeschlagen.
Fehler	DMS-EVENT-0082	Ein Aufruf zur Bereinigung der Aufgabendaten ist fehlgeschlagen.
Statusänderung	DMS-EVENT-0081	Das Neuladen der Tabellendetails wurde angefordert.
Statusänderung		Die Replikationsaufgabe wurde kopiert.
Statusänderung		Das Kopieren der Replikationsaufgabe ist fehlgeschlagen.
Statusänderung		Die Replikationsaufgabe wurde verschoben.

Kategorie	Ereignis-ID	Beschreibung
Statusänderung		Das Verschieben der Replikationsaufgabe ist fehlgeschlagen.
Statusänderung		Die Erstellung der Zielaufgabe ist fehlgeschlagen.
Statusänderung		Der Bewertungslauf für die Replikationsaufgabe wurde gestartet.
Statusänderung		Der Bewertungslauf für die Replikationsaufgabe wurde erfolgreich abgeschlossen.
Statusänderung		Der Bewertungslauf für die Replikationsaufgabe wurde mit einem Fehlschlag abgeschlossen.
StateChange		Der Bewertungslauf für die Replikationsaufgabe wurde mit einer Warnung abgeschlossen.
StateChange		Der Bewertungslauf für die Replikationsaufgabe wurde mit einem Fehler abgeschlossen.
StateChange		Der Bewertungslauf %s für die Replikationsaufgabe wurde abgebrochen.
StateChange		Der Bewertungslauf %s für die Replikationsaufgabe wurde gelöscht.
StateChange		Der Bewertungslauf für die Replikationsaufgabe konnten keine Ressourcen bereitstellen.
StateChange		Replikationsaufgabe ist fehlgeschlagen.

Kategorie	Ereignis-ID	Beschreibung
Erstellung		Die Replikationsaufgabe wurde erstellt.
ConfigurationChange		Eine Replikationsaufgabe wurde geändert.
Fehler		Eine Replikationsaufgabe ist fehlgeschlagen.
StateChange	DMS-EVENT-0091	Der Lesevorgang wurde unterbrochen, das Limit für die Auslagerungsdateien wurde erreicht.
StateChange	DMS-EVENT-0092	Der Lesevorgang wurde unterbrochen, das Limit für die Festplattennutzung wurde erreicht.
StateChange	DMS-EVENT-0093	Der Lesevorgang wurde unterbrochen, das Limit für die Festplattennutzung wurde erreicht.
StateChange	DMS-EVENT-0093	Der Lesevorgang wurde fortgesetzt.
StateChange	DMS-EVENT-0069	Die Replikationsaufgabe wurde mit taskType: %s, startType: %s gestartet
StateChange	DMS-EVENT-0079	Die Replikationsaufgabe wurde angehalten.
Löschung	DMS-EVENT-0073	Die Replikationsaufgabe wurde gelöscht.

Replikations-Ereignismeldungen

In der folgenden Tabelle sind die möglichen Kategorien und Ereignisse für den Quelltyp Replikation angegeben.

Kategorie	Beschreibung
Statusänderung	Ereignis zur Hochskalierung der DMS-Replikation.
Statusänderung	Ereignis zur Herunterskalierung der DMS-Replikation.
Statusänderung	Das DMS-Replikationsskalierungsereignis wurde abgeschlossen.
Statusänderung	Die DMS-Replikation wurde erstellt.
Statusänderung	Die DMS-Replikation wird initialisiert.
Statusänderung	Die DMS-Replikation bereitet die Ressourcen für die Metadaten erfassung vor.
Statusänderung	Die mit der DMS-Replikation verbundenen Verbindungen werden getestet.
Statusänderung	Die DMS-Replikation ruft Metadaten ab
Statusänderung	Die DMS-Replikation berechnet die Kapazität
Statusänderung	Die DMS-Replikation stellt ihre Kapazität bereit
Statusänderung	Die DMS-Replikation wurde bereitgestellt.
Statusänderung	Die DMS-Replikation wurde gestartet
Statusänderung	Die DMS-Replikation wird ausgeführt.
Statusänderung	Die DMS-Replikation wird angehalten.
Statusänderung	Die Replikation wurde angehalten.
Statusänderung	Die DMS-Replikation wird geändert.
Statusänderung	Die DMS-Replikation wird gelöscht.
Statusänderung	Die DMS-Replikation nimmt die Bereitstellung ihrer Kapazität zurück
Statusänderung	Die Bereitstellung der DMS-Replikation wurde zurückgenommen.

Kategorie	Beschreibung
Fehler	Die DMS-Replikation ist fehlgeschlagen.

Arbeiten mit Amazon-SNS-Ereignissen und Benachrichtigungen in AWS Database Migration Service

Ab der Veröffentlichung von AWS DMS 3.4.5 und späteren Versionen empfehlen wir, Amazon EventBridge zu verwenden, um Benachrichtigungen bereitzustellen, wenn ein AWS DMS-Ereignis eintritt. Weitere Informationen zur Verwendung von EventBridge-Ereignissen mit AWS DMS finden Sie unter [Arbeiten mit Amazon-Event-Bridge-Ereignissen und Benachrichtigungen in AWS Database Migration Service](#).

verschieben von Event-Abonnements zu Amazon EventBridge

Sie können den folgenden AWS CLI-Befehl verwenden, um bis zu 10 aktive Ereignis-Abonnements von DMS zu Amazon EventBridge zu migrieren.

```
update-subscriptions-to-event-bridge [--force-move | --no-force-move]
```

Standardmäßig migriert AWS DMS aktive Event-Abonnements nur, wenn Ihre Replikations-Instance die aktuelle Version AWS DMS 3.4.5 und höher hat. Zum Übergehen dieses Verhaltens kann die Option `--force-move` verwendet werden. Einige Ereignistypen sind jedoch möglicherweise nicht mit Amazon EventBridge verfügbar, wenn Ihre Replikations-Instances nicht aktualisiert wurden.

Um den `update-subscriptions-to-event-bridge`-CLI-Befehl ausführen zu können, muss ein AWS Identity and Access Management (IAM)-Benutzer über die folgenden Richtlinienberechtigungen verfügen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "events:PutTargets",
        "events:EnableRule",
        "events:PutRule"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    }  
  ]  
}
```

Weitere Informationen zum Verschieben von Abonnements zu EventBridge finden Sie unter [UpdateSubscriptionsToEventBridge](#) in der AWS Database Migration Service-API-Referenz.

Arbeiten mit Amazon-SNS-Ereignissen und Benachrichtigungen

Die AWS DMS-Versionen 3.4.5 und früher unterstützen die Arbeit mit Ereignissen und Benachrichtigungen, wie im Folgenden beschrieben.

AWS Database Migration Service (AWS DMS) kann Amazon Simple Notification Service (Amazon SNS) verwenden, um Benachrichtigungen bereitzustellen, wenn ein AWS-DMS-Ereignis eintritt, z. B. die Erstellung oder Löschung einer Replikations-Instance. Sie können mit diesen Benachrichtigungen in einem beliebigen Format arbeiten, das von Amazon SNS für eine AWS-Region unterstützt wird, z. B. als E-Mail-Nachricht, Textnachricht oder Aufruf an einen HTTP-Endpunkt.

AWS DMS stuft Ereignisse in Kategorien ein, die Sie abonnieren können. Dadurch werden Sie benachrichtigt, wenn ein Ereignis in dieser Kategorie auftritt. Wenn Sie beispielsweise die Kategorie "Erstellung" für eine bestimmte Replikations-Instance abonnieren, werden Sie benachrichtigt, sobald ein erstellungsbezogenes Ereignis auftritt, das sich auf Ihre Replikations-Instance auswirkt. Wenn Sie eine Konfigurationsänderungs-Kategorie für eine Replikations-Instance abonnieren, werden Sie benachrichtigt, wenn die Konfiguration der Replikations-Instance geändert wird. Außerdem erhalten Sie eine Benachrichtigung, wenn ein Abonnement für Ereignisbenachrichtigungen geändert wird. Eine Liste der von AWS DMS bereitgestellten Ereigniskategorien finden Sie im Folgenden unter [AWS-DMS-Ereigniskategorien und -Ereignismeldungen für SNS-Benachrichtigungen](#).

AWS DMS sendet Ereignisbenachrichtigungen an die Adressen, die Sie beim Erstellen eines Ereignisabonnements angeben. Sie sollten mehrere verschiedene Abonnements erstellen, beispielsweise ein Abonnement, das alle Ereignisbenachrichtigungen empfängt, und ein anderes Abonnement, das nur kritische Ereignisse für Ihre DMS-Produktionsressourcen enthält. Sie können Benachrichtigungen einfach ohne Löschen eines Abonnements deaktivieren, indem Sie die Option Aktiviert in der AWS-Konsole abwählen, oder indem Sie für den Parameter `Enabled` mithilfe der AWS-API den Wert auf `false` festlegen.

Note

AWS-DMS-Ereignisbenachrichtigungen mithilfe von SMS-Nachrichten sind derzeit für AWS-Ressourcen in allen AWS-Regionen verfügbar, in denen Amazon SNS unterstützt wird. Eine Liste der AWS-Regionen und Länder, in denen Amazon SNS SMS-Nachrichten unterstützt, finden Sie unter [Unterstützte Regionen und Länder](#).

Weitere Informationen zum Verwenden von SMS-Nachrichten mit SNS finden Sie unter [Sending and receiving SMS notifications using Amazon SNS](#) (Senden und Empfangen von SMS-Benachrichtigungen mit Amazon SNS).

AWS DMS-Ereignisbenachrichtigungen unterscheiden sich von CloudTrail-Ereignissen in CloudWatch oder EventBridge. CloudTrail-Ereignisbenachrichtigungen können durch jeden API-Aufruf generiert werden. DMS sendet nur dann eine Benachrichtigung, wenn ein DMS-Ereignis eintritt.

AWS DMS verwendet eine Abonnement-ID zum Identifizieren des jeweiligen Abonnements. Sie können mehrere AWS-DMS-Ereignisabonnements in demselben Amazon-SNS-Thema veröffentlichen. Wenn Sie Ereignisbenachrichtigungen nutzen, fallen Amazon-SNS-Gebühren an. Weitere Informationen zur Fakturierung von Amazon SNS finden Sie unter [Amazon SNS – Preise](#).

Gehen Sie zum Abonnieren von AWS-DMS-Ereignissen wie folgt vor:

1. Erstellen Sie ein Amazon-SNS-Thema. Im Thema geben Sie an, welche Art von Benachrichtigung Sie empfangen möchten und an welche Adresse oder Nummer die Benachrichtigung gesendet werden soll.
2. Erstellen Sie mit der AWS Management Console-, der AWS CLI- oder der AWS-DMS-API ein Abonnement für AWS-DMS-Ereignisbenachrichtigungen.
3. AWS DMS sendet eine Bestätigungs-E-Mail oder SMS-Nachricht an die Adressen, die Sie mit Ihrem Abonnement übermittelt haben. Um Ihr Abonnement zu bestätigen, klicken Sie in der Bestätigungs-E-Mail oder SMS-Nachricht auf den Link.
4. Wenn Sie das Abonnement bestätigt haben, wird der Status Ihres Abonnements im Abschnitt Ereignisabonnements in der AWS-DMS-Konsole aktualisiert.
5. Sie erhalten dann Ereignisbenachrichtigungen.

Die Liste der Kategorien und Ereignisse, über die Sie benachrichtigt werden können, finden Sie im folgenden Abschnitt. Weitere Informationen zum Abonnieren von und Arbeiten mit AWS-DMS-

Ereignisabonnements finden Sie unter [Abonnieren von AWS-DMS-Ereignisbenachrichtigungen mithilfe von SNS](#).

AWS-DMS-Ereigniskategorien und -Ereignismeldungen für SNS-Benachrichtigungen

Important

Ab der Veröffentlichung von AWS DMS 3.4.5 und späteren Versionen empfehlen wir, Amazon EventBridge zu verwenden, um Benachrichtigungen bereitzustellen, wenn ein AWS DMS-Ereignis eintritt. Weitere Informationen zur Verwendung von EventBridge-Ereignissen mit AWS DMS finden Sie unter [Arbeiten mit Amazon-Event-Bridge-Ereignissen und Benachrichtigungen in AWS Database Migration Service](#).

AWS DMS generiert eine beträchtliche Anzahl von Ereignissen in Kategorien, die Sie mithilfe der AWS-DMS -Konsole oder der AWS-DMS-API abonnieren können. Jede Kategorie gilt für einen Quelltyp. Derzeit unterstützt AWS DMS die Quelltypen „Replikations-Instance“ und „Replikationsaufgabe“.

In der folgenden Tabelle sind die möglichen Kategorien und Ereignisse für den Quelltyp "Replikations-Instance" angegeben.

Kategorie	DMS-Ereignis-ID	Beschreibung
Konfigurationsänderung	DMS-EVENT-0012	Die Replikations-Instance-Klasse für diese Replikations-Instance wird geändert.
Konfigurationsänderung	DMS-EVENT-0014	Die Replikations-Instance-Klasse für diese Replikations-Instance wurde geändert.
Konfigurationsänderung	DMS-EVENT-0018	Der Speicher für die Replikations-Instance wird erweitert.
Konfigurationsänderung	DMS-EVENT-0017	Der Speicher für die Replikations-Instance wurde erweitert.

Kategorie	DMS-Ereignis-ID	Beschreibung
Konfigurationsänderung	DMS-EVENT-0024	Die Replikations-Instance wird auf eine Multi-AZ-Konfiguration umgestellt.
Konfigurationsänderung	DMS-EVENT-0025	Die Umstellung der Replikations-Instance auf eine Multi-AZ-Konfiguration ist abgeschlossen.
Konfigurationsänderung	DMS-EVENT-0030	Die Replikations-Instance wird auf eine Single-AZ-Konfiguration umgestellt.
Konfigurationsänderung	DMS-EVENT-0029	Die Umstellung der Replikations-Instance auf eine Single-AZ-Konfiguration ist abgeschlossen.
Erstellung	DMS-EVENT-0067	Eine Replikations-Instance wird erstellt.
Erstellung	DMS-EVENT-0005	Eine Replikations-Instance wird erstellt.
Löschung	DMS-EVENT-0066	Die Replikations-Instance wird gelöscht.
Löschung	DMS-EVENT-0003	Die Replikations-Instance wird gelöscht.
Wartung	DMS-EVENT-0047	Die Managementsoftware auf der Replikations-Instance wurde aktualisiert.
Wartung	DMS-EVENT-0026	Die Offline-Wartung der Replikations-Instance wird durchgeführt. Die Replikations-Instance steht derzeit nicht zur Verfügung.
Wartung	DMS-EVENT-0027	Die Offline-Wartung der Replikations-Instance ist abgeschlossen. Die Replikations-Instance steht nun zur Verfügung.
Wartung	DMS-EVENT-0068	Eine Replikations-Instance befindet sich in einem Zustand, der nicht aktualisiert werden kann.
LowStorage	DMS-EVENT-0007	Die Replikations-Instance hat mehr als 90 % ihres zugewiesenen Speichers verbraucht. Sie können den Speicherplatz für eine Replikations-Instance mit der Metrik „Freier Speicherplatz“ überwachen.

Kategorie	DMS-Ereignis-ID	Beschreibung
Failover	DMS-EVENT-0013	Failover für eine Multi-AZ-Replikations-Instance wurde gestartet.
Failover	DMS-EVENT-0049	Failover für eine Multi-AZ-Replikations-Instance ist abgeschlossen.
Failover	DMS-EVENT-0015	Multi-AZ-Failover zu Standby ist abgeschlossen.
Failover	DMS-EVENT-0050	Die Multi-AZ-Aktivierung wurde gestartet.
Failover	DMS-EVENT-0051	Die Multi-AZ-Aktivierung wurde abgeschlossen.
Failover	DMS-EVENT-0034	Wenn Sie zu häufig einen Failover anfordern, wird dieses Ereignis anstelle regelmäßiger Failover-Ereignisse ausgelöst.
Fehler	DMS-EVENT-0031	Bei der Replikations-Instance ist ein Speicherfehler aufgetreten.
Fehler	DMS-EVENT-0036	Die Replikations-Instance ist aufgrund eines nicht kompatiblen Netzwerks fehlgeschlagen.
Fehler	DMS-EVENT-0037	Der Service kann nicht auf den AWS KMS-Schlüssel zugreifen, der zur Verschlüsselung des Daten-Volumens verwendet wurde.

In der folgenden Tabelle sind die möglichen Kategorien und Ereignisse für den Quelltyp „Replikationsaufgabe“ angegeben.

Kategorie	DMS-Ereignis-ID	Beschreibung
Statusänderung	DMS-EVENT-0069	Die Replikationsaufgabe wurde gestartet.
Statusänderung	DMS-EVENT-0081	Das Neuladen der Tabellendetails wurde angefordert.
Statusänderung	DMS-EVENT-0079	Die Replikationsaufgabe wurde angehalten.

Kategorie	DMS-Ereignis-ID	Beschreibung
Statusänderung	DMS-EVENT-0091	Der Lesevorgang wurde unterbrochen, das Limit für die Auslagerungsdateien wurde erreicht.
Statusänderung	DMS-EVENT-0092	Der Lesevorgang wurde unterbrochen, das Limit für die Festplattennutzung wurde erreicht.
Statusänderung	DMS-EVENT-0093	Der Lesevorgang wurde fortgesetzt.
Fehler	DMS-EVENT-0078	Die Replikationsaufgabe ist fehlgeschlagen.
Fehler	DMS-EVENT-0082	Ein Aufruf zum Löschen der Aufgabe konnte die Aufgabendaten nicht bereinigen.
Konfigurationsänderung	DMS-EVENT-0080	Die Replikationsaufgabe wird geändert.
Löschung	DMS-EVENT-0073	Die Replikationsaufgabe wird gelöscht.
Erstellung	DMS-EVENT-0074	Die Replikationsaufgabe wird erstellt.

Das folgende Beispiel zeigt ein AWS DMS-Ereignisabonnement mit der Kategorie „Statusänderung“.

```

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

```

Abonnieren von AWS-DMS-Ereignisbenachrichtigungen mithilfe von SNS

Important

Ab der Veröffentlichung von AWS DMS 3.4.5 und späteren Versionen empfehlen wir, Amazon EventBridge zu verwenden, um Benachrichtigungen bereitzustellen, wenn ein AWS DMS-Ereignis eintritt. Weitere Informationen zur Verwendung von EventBridge-Ereignissen mit AWS DMS finden Sie unter [Arbeiten mit Amazon-Event-Bridge-Ereignissen und Benachrichtigungen in AWS Database Migration Service](#).

Sie können ein Abonnement für AWS DMS-Ereignisbenachrichtigungen erstellen, sodass Sie beim Auftreten eines AWS DMS-Ereignisses benachrichtigt werden. Am einfachsten lässt sich ein Abonnement mit der AWS DMS-Konsole erstellen. In einem Benachrichtigungsabonnement entscheiden Sie, wie und wohin Benachrichtigungen gesendet werden sollen. Sie geben den Quelltyp an, zu dem Sie Benachrichtigungen erhalten möchten. Derzeit werden die Quelltypen „Replikations-Instance“ und „Replikationsaufgabe“ von AWS DMS unterstützt. Je nach ausgewähltem Quelltyp wählen Sie die Ereigniskategorien und identifizieren die Quellen, für die Sie Ereignisbenachrichtigungen erhalten möchten.

Verwendung von AWS Management Console

Important

Ab der Veröffentlichung von AWS DMS 3.4.5 und späteren Versionen empfehlen wir, Amazon EventBridge zu verwenden, um Benachrichtigungen bereitzustellen, wenn ein AWS DMS-Ereignis eintritt. Weitere Informationen zur Verwendung von EventBridge-Ereignissen mit AWS DMS finden Sie unter [Arbeiten mit Amazon-Event-Bridge-Ereignissen und Benachrichtigungen in AWS Database Migration Service](#).

So abonnieren Sie AWS-DMS-Ereignisbenachrichtigungen mit Amazon SNS mithilfe der Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als IAM-Benutzer angemeldet sind, müssen Sie über die entsprechenden Berechtigungen für den Zugriff auf AWS DMS verfügen.

2. Wählen Sie im Navigationsbereich Ereignisabonnements aus.
3. Wählen Sie auf der Seite Ereignisabonnements die Option Ereignisabonnement erstellen aus.
4. Gehen Sie auf der Seite Ereignisabonnement erstellen wie folgt vor:
 - a. Geben Sie unter Details für Name einen Namen für das Abonnement für Ereignisbenachrichtigungen ein.
 - b. Wählen Sie Aktiviert, um das Abonnement zu aktivieren. Wenn Sie das Abonnement erstellen möchten, jedoch noch keine Benachrichtigungen gesendet haben, wählen Sie nicht Aktiviert aus.
 - c. Wählen Sie unter Ziel entweder Vorhandene Themen, Neues E-Mail-Thema erstellen oder Neues SMS-Thema erstellen aus, um Benachrichtigungen zu senden. Entweder müssen Sie ein vorhandenes Amazon-SNS-Thema haben, an das Sie Benachrichtigungen senden möchten, oder Sie müssen das Thema selbst erstellen. Wenn Sie ein Thema erstellen, können Sie eine E-Mail-Adresse eingeben, an die die Benachrichtigungen gesendet werden sollen.
 - d. Wählen Sie unter Ereignisquelle für Quelltyp einen Quelltyp aus. Die einzigen Optionen sind replication-instance und replication-task.
 - e. Je nach ausgewähltem Quelltyp wählen Sie die Ereigniskategorien und Quellen aus, für die Sie Ereignisbenachrichtigungen erhalten möchten.

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. Wählen Sie Ereignis-Abonnement erstellen.

In der AWS-DMS-Konsole wird die Erstellung des Abonnements angezeigt.

Note

Sie können auch mithilfe der AWS DMS-API und der CLI Abonnements für Amazon-SNS-Ereignisbenachrichtigungen erstellen. Weitere Informationen finden Sie unter [CreateEventSubscription](#) in der AWS DMSAPI-Referenz und [create-event-subscription](#) in der AWS DMS.CLI-Referenzdokumentation.

Überprüfen der Zugriffsrichtlinie für Ihr SNS-Thema

Ihre SNS-Zugriffsrichtlinie erfordert Berechtigungen, die es AWS DMS ermöglichen, Ereignisse zu Ihrem SNS-Thema zu veröffentlichen. Sie können Ihre Zugriffsrichtlinie wie in den folgenden Verfahren beschrieben validieren und aktualisieren.

So validieren Sie Ihre Zugriffsrichtlinie

1. Öffnen Sie die Amazon-SNS-Konsole.
2. Wählen Sie im Navigationsbereich Themen und dann das Thema aus, zu dem Sie DMS-Benachrichtigungen erhalten möchten.
3. Wählen Sie die Registerkarte Zugriffsrichtlinie aus.

Sie können Ihre Richtlinie aktualisieren, wenn Ihre SNS-Zugriffsrichtlinie die Veröffentlichung von Ereignissen zu Ihrem SNS-Thema durch AWS DMS nicht zulässt.

So aktualisieren Sie Ihre Zugriffsrichtlinie

1. Wählen Sie auf Ihrer Themenseite im Bereich Details die Option Bearbeiten aus.
2. Erweitern Sie den Abschnitt Zugriffsrichtlinie und fügen Sie das folgende Richtliniendokument in den JSON-Editor ein.

```
{
  "Sid": "dms-allow-publish",
  "Effect": "Allow",
  "Principal": {
```

```
    "Service": "dms.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "your-SNS-topic-ARN"
}
```

Wir empfehlen Ihnen, den Zugriff auf Ihr SNS-Thema weiter einzuschränken, indem Sie die `aws:SourceArn`-Bedingung angeben, d. h. den DMS-EventSubscription-ARN, der Ereignisse zum Thema veröffentlicht.

```
...
"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. Wählen Sie **Save Changes**.

Datenvalidierung mit AWS DMS

Themen

- [Replikationsaufgaben-Statistiken](#)
- [Statistiken zu Replikationsaufgaben mit Amazon CloudWatch](#)
- [Erneutes Validieren von Tabellen während einer Aufgabe](#)
- [Ändern von Validierungsregeln mithilfe des JSON-Editors](#)
- [Reine Validierungsaufgaben](#)
- [Fehlerbehebung](#)
- [Redshift-Validierungsleistung](#)
- [Einschränkungen](#)
- [Zieldatenvalidierung in Amazon S3](#)

AWS DMS unterstützt die Datenvalidierung, um sicherzustellen, dass Ihre Daten korrekt von der Quelle zum Ziel migriert wurden. Wenn diese Option aktiviert ist, beginnt die Validierung unmittelbar nach einer Volllastaufgabe für eine Tabelle. Bei der Validierung werden die inkrementellen Änderungen für eine CDC-fähige Aufgabe verglichen, sobald sie auftreten.

Während der Datenvalidierung vergleicht AWS DMS jede Zeile in der Quelle mit der entsprechenden Zeile im Ziel und stellt sicher, dass die Zeilen die gleichen Daten enthalten. Alle Abweichungen werden gemeldet. Zu diesem Zweck führt AWS DMS entsprechende Abfragen durch, um die Daten abzurufen. Beachten Sie, dass diese Abfragen zusätzliche Ressourcen an der Quelle und am Ziel sowie zusätzliche Netzwerkressourcen verbrauchen.

Bei einer reinen CDC-Aufgabe mit aktivierter Validierung werden alle bereits vorhandenen Daten in einer Tabelle überprüft, bevor mit der Validierung neuer Daten begonnen wird.

Die Datenvalidierung funktioniert mit den folgenden Quelldatenbanken, wenn AWS DMS sie als Quellendpunkte unterstützt:

- Oracle
- PostgreSQL-kompatible Datenbank (PostgreSQL, Aurora PostgreSQL oder Aurora Serverless für PostgreSQL)
- MySQL-kompatible Datenbank (MySQL, MariaDB, Aurora MySQL oder Aurora Serverless für MySQL)

- Microsoft SQL Server
- IBM Db2 (LUW)

Die Datenvalidierung funktioniert mit den folgenden Zieldatenbanken, wenn AWS DMS sie als Zielpunkte unterstützt:

- Oracle
- PostgreSQL-kompatible Datenbank (PostgreSQL, Aurora PostgreSQL oder Aurora Serverless für PostgreSQL)
- MySQL-kompatible Datenbank (MySQL, MariaDB, Aurora MySQL oder Aurora Serverless für MySQL)
- Microsoft SQL Server
- IBM Db2 (LUW)
- Amazon Redshift
- Amazon S3. Informationen zur Validierung von Amazon-S3-Zieldaten finden Sie unter [Zieldatenvalidierung in Amazon S3](#).

Weitere Informationen zu den unterstützten Endpunkten finden Sie unter [Arbeiten mit AWS-DMS-Endpunkten](#).

Die Datenvalidierung erfordert zusätzliche Zeit, die über die Dauer hinausgeht, die für die Migration selbst erforderlich ist. Die zusätzlich benötigte Zeit hängt davon ab, wie viele Daten migriert wurden.

Weitere Informationen zu diesen Einstellungen finden Sie unter [Aufgabeneinstellungen zur Datenvalidierung](#).

Ein Beispiel für ValidationSettings-Aufgabeneinstellungen in einer JSON-Datei finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Replikationsaufgaben-Statistiken

Wenn die Datenvalidierung aktiviert ist, stellt AWS DMS die folgenden Statistiken auf Tabellenebene bereit:

- ValidationState – Der Validierungsstatus der Tabelle. Folgende Parameterwerte sind möglich:
 - Not enabled – Für die Tabelle in der Migrationsaufgabe ist keine Validierung aktiviert.

- Pending records – Einige Datensätze in der Tabelle müssen noch validiert werden.
- Nicht übereinstimmende Datensätze – Bei einigen Datensätzen in der Tabelle gibt es eine Abweichung zwischen Quelle und Ziel. Ein Konflikt kann aus unterschiedlichen Gründen auftreten. Weitere Informationen finden Sie in der Tabelle `awsdms_control.awsdms_validation_failures_v1` auf dem Ziel-Endpunkt.
- Suspended records (Ausgesetzte Datensätze) – Einige Datensätze in der Tabelle können nicht validiert werden.
- No primary key (Kein Primärschlüssel) – Die Tabelle kann aufgrund eines fehlenden Primärschlüssels nicht validiert werden.
- Table error (Tabellenfehler) – Die Tabelle wurde aufgrund ihres Fehlerstatus nicht validiert, zudem wurden einige Daten nicht migriert.
- Validiert – Alle Zeilen der Tabelle wurden validiert. Wenn die Tabelle aktualisiert wurde, ändert sich der Status möglicherweise in "Validated".
- Error (Fehler) – Die Tabelle kann aufgrund eines unerwarteten Fehlers nicht validiert werden.
- Ausstehende Validierung – Die Validierung der Tabelle steht noch aus.
- Tabelle wird vorbereitet – Die in der Migrationsaufgabe aktivierte Tabelle wird für die Validierung vorbereitet.
- Ausstehende erneute Validierung – Die Validierung aller Zeilen in der Tabelle steht noch aus, nachdem die Tabelle aktualisiert wurde.
- ValidationPending (Validierung ausstehende) – Die Anzahl der Datensätze, die in die Zieldatenbank migriert aber noch nicht validiert wurden.
- ValidationSuspended – Die Anzahl der Datensätze, die AWS DMS nicht vergleichen kann. Wenn beispielsweise ein Datensatz an der Quelle ständig aktualisiert wird, kann AWS DMS Quelle und Ziel nicht vergleichen.
- ValidationFailed – Die Anzahl der Datensätze, die die Datenvalidierung nicht bestanden haben.

Ein Beispiel für `ValidationSettings`-Aufgabeneinstellungen in einer JSON-Datei finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Sie können die Informationen zur Datenvalidierung mit der Konsole, der AWS CLI oder der AWS DMS-API anzeigen.

- Sie können in der Konsole eine Aufgabe zur Validierung auswählen, wenn Sie die Aufgabe erstellen oder ändern. Zum Anzeigen des Validierungsberichts in der Konsole wählen Sie die Aufgabe auf der Seite Tasks aus und klicken dann im Detailbereich auf Table statistics.
- Setzen Sie bei Verwendung der CLI den Parameter EnableValidation auf true, wenn Sie eine Aufgabe erstellen oder ändern, um mit der Datenvalidierung zu beginnen. Im folgenden Beispiel wird eine Aufgabe erstellt und die Datenvalidierung aktiviert.

```
create-replication-task
  --replication-task-settings '{"ValidationSettings":{"EnableValidation":true}}'
  --replication-instance-arn arn:aws:dms:us-east-1:5731014:
    rep:36KWVMB7Q
  --source-endpoint-arn arn:aws:dms:us-east-1:5731014:
    endpoint:CSZAEFQURFYMM
  --target-endpoint-arn arn:aws:dms:us-east-1:5731014:
    endpoint:CGPP7MF6WT4JQ
  --migration-type full-load-and-cdc
  --table-mappings '{"rules": [{"rule-type": "selection", "rule-id": "1",
    "rule-name": "1", "object-locator": {"schema-name": "data_types", "table-name":
    "%"}},
    {"rule-action": "include"}]}'
```

Verwenden Sie den Befehl `describe-table-statistics`, um den Datenvalidierungsbericht im JSON-Format zu erhalten. Mit dem folgenden Befehl wird der Datenvalidierungsbericht angezeigt.

```
aws dms describe-table-statistics --replication-task-arn arn:aws:dms:us-
east-1:5731014:
rep:36KWVMB7Q
```

Der Bericht sieht in etwa wie folgt aus.

```
{
  "ReplicationTaskArn": "arn:aws:dms:us-west-2:5731014:task:VFPFTYKK2RYSI",
  "TableStatistics": [
    {
      "ValidationPendingRecords": 2,
      "Inserts": 25,
      "ValidationState": "Pending records",
      "ValidationSuspendedRecords": 0,
      "LastUpdateTime": 1510181065.349,
      "FullLoadErrorRows": 0,
```



```
    "FullLoadCondtnlChkFailedRows": 0,  
    "Ddls": 0,  
    "TableName": "t_binary",  
    "ValidationFailedRecords": 0,  
    "Updates": 0,  
    "FullLoadRows": 10,  
    "TableState": "Table completed",  
    "SchemaName": "d_types_s_sqlserver",  
    "Deletes": 0  
  }  
}
```

- Wenn Sie die AWS DMS-API verwenden, erstellen Sie mithilfe der Aktion `CreateReplicationTask` eine Aufgabe und setzen Sie den Parameter `EnableValidation` auf `true`, um die über die Aufgabe migrierten Daten zu validieren. Nutzen Sie die `DescribeTableStatistics`-Aktion, um den Datenvalidierungsbericht im JSON-Format zu erhalten.

Statistiken zu Replikationsaufgaben mit Amazon CloudWatch

Wenn Amazon CloudWatch aktiviert ist, stellt AWS DMS die folgenden Statistiken für Replikationsaufgaben bereit:

- `ValidationSucceededRecordCount` – Anzahl der Zeilen, die von AWS DMS pro Minute validiert wurden.
- `ValidationAttemptedRecordCount` – Anzahl von Zeilen, für die pro Minute eine Validierung versucht wurde.
- `ValidationFailedOverallCount` – Anzahl von Zeilen, bei denen die Validierung fehlschlug.
- `ValidationSuspendedOverallCount` – Anzahl von Zeilen, bei denen die Validierung ausgesetzt wurde.
- `ValidationPendingOverallCount` – Anzahl von Zeilen, bei denen eine Validierung noch ansteht.
- `ValidationBulkQuerySourceLatency` – AWS DMS kann die Datenvalidierung gebündelt durchführen. Dies ist insbesondere bei bestimmten Szenarien während Volllastaufgaben oder der laufenden Replikation relevant, wenn viele Änderungen vorliegen. Diese Metrik gibt die Latenz an, die zum Lesen eines Massendatensatzes aus dem Quellendpunkt erforderlich ist.
- `ValidationBulkQueryTargetLatency` – AWS DMS kann die Datenvalidierung gebündelt durchführen. Dies ist insbesondere bei bestimmten Szenarien während Volllastaufgaben oder der laufenden

Replikation relevant, wenn viele Änderungen vorliegen. Diese Metrik gibt die Latenz an, die zum Lesen eines Massendatensatzes am Zielendpunkt erforderlich ist.

- **ValidationItemQuerySourceLatency** – Während der laufenden Replikation können mit der Datenvalidierung fortlaufende Änderungen identifiziert und validiert werden. Diese Metrik gibt die Latenz an, die zum Lesen solcher Änderungen aus der Quelle erforderlich ist. Die Validierung kann mehr Abfragen als basierend auf der Anzahl von Änderungen erforderlich ausführen, wenn während der Validierung Fehler aufgetreten sind.
- **ValidationItemQueryTargetLatency** – Während der laufenden Replikation können mit der Datenvalidierung laufende Änderungen identifiziert und zeilenweise validiert werden. Diese Metrik gibt die Latenz zum Lesen solcher Änderungen am Ziel an. Die Validierung kann mehr Abfragen als basierend auf der Anzahl von Änderungen erforderlich ausführen, wenn während der Validierung Fehler aufgetreten sind.

Um Informationen zur Datenvalidierung aus CloudWatch-fähigen Statistiken zu erfassen, wählen Sie CloudWatch Logs aktivieren aus, wenn Sie eine Aufgabe mithilfe der Konsole erstellen oder ändern. Gehen Sie dann folgendermaßen vor, um die Informationen zur Datenvalidierung anzuzeigen und sicherzustellen, dass Ihre Daten korrekt von der Quelle zum Ziel migriert wurden.

1. Wählen Sie die Aufgabe auf der Seite Datenbankmigrationsaufgaben aus.
2. Wählen Sie die Registerkarte CloudWatch-Metriken aus.
3. Wählen Sie im Drop-down-Menü Validierung aus.

Erneutes Validieren von Tabellen während einer Aufgabe

Während eine Aufgabe ausgeführt wird, können Sie von AWS DMS eine Datenvalidierung anfordern.

AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als AWS Identity and Access Management (IAM)-Benutzer angemeldet sind, müssen Sie über die entsprechenden Berechtigungen für den Zugriff auf AWS DMS verfügen. Informationen zu den erforderlichen Berechtigungen finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich Tasks (Aufgaben) aus.

3. Wählen Sie die laufende Aufgabe mit der Tabelle aus, die sie erneut validieren möchten.
4. Wählen Sie die Registerkarte Table Statistics (Tabellenstatistiken).
5. Wählen Sie die Tabelle aus, die Sie erneut validieren möchten (Sie können bis zu 10 Tabellen gleichzeitig auswählen). Für Aufgaben, die derzeit nicht mehr ausgeführt werden, kann die Tabelle bzw. können die Tabellen nicht erneut validiert werden.
6. Wählen Sie Revalidate (Erneut validieren).

Ändern von Validierungsregeln mithilfe des JSON-Editors

Gehen Sie wie folgt vor, um einer Aufgabe mithilfe des JSON-Editors von der AWS DMS-Konsole aus eine Validierungsregel hinzuzufügen:

1. Wählen Sie Datenbankmigrationsaufgaben aus.
2. Wählen Sie Ihre Aufgabe in der Liste der Migrationsaufgaben aus.
3. Wenn Ihre Aufgabe gerade ausgeführt wird, wählen Sie im Drop-down-Menü Aktionen die Option Stoppen aus.
4. Nachdem die Aufgabe beendet wurde, wählen Sie im Drop-down-Menü Aktionen die Option Ändern aus, um die Aufgabe zu ändern.
5. Wählen Sie im Abschnitt Tabellenzuordnungen die Option JSON-Editor aus und fügen Sie den Tabellenzuordnungen Ihre Validierungsregel hinzu.

Sie können beispielsweise die folgende Validierungsregel hinzufügen, um eine Ersetzungsfunktion für die Quelle auszuführen. Wenn die Validierungsregel in diesem Fall auf ein Null-Byte trifft, wird dieses als Leerzeichen validiert.

```
{
  "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}"  
}
```

Reine Validierungsaufgaben

Sie können reine Validierungsaufgaben erstellen, um Daten in der Vorschau anzuzeigen und zu validieren, ohne eine Migration oder Datenreplikation auszuführen. Setzen Sie die Einstellungen `EnableValidation` und `ValidationOnly` auf `true`, um eine reine Validierungsaufgabe zu erstellen. Bei Aktivierung von `ValidationOnly` gelten zusätzliche Anforderungen. Weitere Informationen finden Sie unter [Aufgabeneinstellungen zur Datenvalidierung](#).

Bei einer reinen Volllast-Migration wird eine reine Validierungsaufgabe viel schneller abgeschlossen als ihr CDC-Äquivalent, wenn viele Fehler gemeldet werden. Änderungen am Quell- oder Zielpunkt werden im Volllastmodus jedoch als Fehler gemeldet, was einen möglichen Nachteil darstellt.

Bei einer reinen CDC-Validierungsaufgabe werden die Validierung auf der Grundlage der durchschnittlichen Latenz verzögert und Fehlschläge mehrmals wiederholt, bevor sie gemeldet werden. Wenn die meisten Datenvergleiche zu Fehlern führen, ist eine reine Validierungsaufgabe für den CDC-Modus sehr langsam, was einen möglichen Nachteil darstellt.

Eine reine Validierungsaufgabe muss in der gleichen Richtung wie die Replikationsaufgabe eingerichtet werden, insbesondere für CDC. Dies liegt daran, dass bei einer reinen CDC-Validierungsaufgabe anhand des Änderungsprotokolls der Quelle ermittelt wird, welche Zeilen geändert wurden und neu validiert werden müssen. Wenn das Ziel als Quelle angegeben ist, sind nur Änderungen bekannt, die von DMS an das Ziel gesendet wurden, und es kann nicht garantiert werden, dass Replikationsfehler erkannt werden.

Reine Volllastvalidierung

Ab AWS DMS Version 3.4.6 und höher werden bei einer Aufgabe zur reinen Volllastvalidierung schnell alle Zeilen aus den Quell- und Zieltabellen in einem einzigen Durchgang verglichen, alle Fehler sofort gemeldet und die Aufgabe dann beendet. In diesem Modus wird die Validierung nie aufgrund von Fehlern unterbrochen; sie ist auf Geschwindigkeit optimiert. Änderungen am Quell- oder Zielpunkt werden jedoch als Fehler gemeldet.

Note

Ab AWS DMS Version 3.4.6 und höher gilt dieses Validierungsverhalten auch für Volllast-Migrationsaufgaben mit aktivierter Validierung.

Reine CDC-Validierung

Bei einer reinen CDC-Validierungsaufgabe werden alle vorhandenen Zeilen zwischen den Quell- und Zieltabellen bei einem Neustart validiert. Darüber hinaus wird eine reine CDC-Validierungsaufgabe kontinuierlich ausgeführt, laufende Replikationsänderungen werden erneut validiert, die Anzahl der bei jedem Durchlauf gemeldeten Fehler werden begrenzt und es wird versucht, nicht übereinstimmende Zeilen erneut auszuführen, bevor sie fehlschlagen. Diese Aufgaben sind so optimiert, dass Fehlalarme vermieden werden.

Die Validierung einer Tabelle (oder der gesamten Aufgabe) wird unterbrochen, wenn die Schwellenwerte `FailureMaxCount` oder `TableFailureMaxCount` überschritten werden. Dies gilt auch für eine CDC- oder Volllast-und-CDC-Migrationsaufgabe mit aktivierter Validierung. Bei einer CDC-Aufgabe mit aktivierter Validierung wird die erneute Validierung für jede geänderte Zeile auf der Grundlage der durchschnittlichen Quell- und Ziellatenz verzögert.

Bei einer reinen CDC-Validierungsaufgabe werden jedoch keine Daten migriert und es ist keine Latenz vorhanden. Standardmäßig ist `ValidationQueryCdcDelaySeconds` auf 180 festgelegt. Sie können ihn erhöhen, um Umgebungen mit hoher Latenz zu berücksichtigen und Fehlalarme zu vermeiden.

Anwendungsfälle für die reine Validierung

Zu den Anwendungsfällen für die Aufteilung des Datenvalidierungsteils einer Migrations- oder Replikationsaufgabe in eine separate reine Validierungsaufgabe gehören unter anderem die folgenden:

- Genaue Kontrolle über den Zeitpunkt der Validierung – Validierungsabfragen belasten sowohl die Quell- als auch die Zielendpunkte zusätzlich. Daher kann es von Vorteil sein, zuerst Daten in einer Aufgabe zu migrieren oder zu replizieren und dann die Ergebnisse in einer anderen Aufgabe zu validieren.
- Geringere Belastung der Replikations-Instance – Es kann von Vorteil sein, die Datenvalidierung aufzuteilen, sodass sie auf einer eigenen Instance ausgeführt wird.

- Schnelle Ermittlung, wie viele Zeilen zu einem bestimmten Zeitpunkt nicht übereinstimmen – Sie können beispielsweise kurz vor oder während eines Produktions-Cut-overs zu einem Zielendpunkt in einem Wartungsfenster eine reine Volllast-Validierungsaufgabe erstellen, um eine Antwort auf Ihre Frage zu erhalten.
- Erwartetes Auftreten von Validierungsfehlern für eine Migrationsaufgabe mit einer CDC-Komponente – Wenn beispielsweise Oracle `varchar2` zu PostgreSQL `jsonb` migriert wird, wiederholt die CDC-Validierung diese fehlgeschlagenen Zeilen und begrenzt die Anzahl der Fehler, die jedes Mal gemeldet werden. Sie können jedoch eine reine Volllast-Validierungsaufgabe erstellen und eine schnellere Antwort erhalten.
- Sie haben ein Skript/Hilfsprogramm zur Datenreparatur entwickelt, das die Tabelle mit den Validierungsfehlern liest – (siehe auch [Fehlerbehebung](#)). Eine reine Volllast-Validierungsaufgabe meldet schnell Fehler, damit das Skript zur Datenreparatur darauf reagieren kann.

Ein Beispiel für `ValidationSettings`-Aufgabeneinstellungen in einer JSON-Datei finden Sie unter [Beispiel für Aufgabeneinstellungen](#).

Fehlerbehebung

Während der Validierung erstellt AWS DMS eine neue Tabelle am Zielendpunkt:

`awsdms_control.awsdms_validation_failures_v1`. Wenn ein Datensatz in den Status `ValidationSuspended` oder `ValidationFailed` wechselt, schreibt AWS DMS Diagnosedaten in `awsdms_control.awsdms_validation_failures_v1`. Sie können diese Tabelle zur Behebung von Validierungsfehlern abfragen.

Informationen zum Ändern des Standardschemas, in dem die Tabelle auf dem Ziel erstellt wird, finden Sie unter [Control table task settings](#).

Im Folgenden finden Sie eine Beschreibung der Tabelle `awsdms_control.awsdms_validation_failures_v1`:

Spaltenname	Datentyp	Beschreibung
TASK_NAME	VARCHAR(128) NOT NULL	AWS DMS-Aufgaben-ID.

Spaltenname	Datentyp	Beschreibung
TABLE_OWNER	VARCHAR(128) NOT NULL	Der Name des Schemabesitzers.
TABLE_NAME	VARCHAR(128) NOT NULL	Tabellenname.
FAILURE_TIME	DATETIME(3) NOT NULL	Zeitpunkt, zu dem der Fehler aufgetreten ist.
KEY_TYPE	VARCHAR(128) NOT NULL	Reserviert für zukünftige Verwendung (Wert ist immer „Zeile“)
KEY	TEXT NOT NULL	Dies ist der Primärschlüssel für den Zeilendatensatztyp.
FAILURE_TYPE	VARCHAR(128) NOT NULL	Schweregrad des Validierungsfehlers. Kann RECORD_DIFF , MISSING_SOURCE oder MISSING_TARGET sein.
DETAILS	VARCHAR(8000) NOT NULL	Zeichenfolge im JSON-Format mit allen Quell-/Zielspaltenwerten, die nicht mit dem angegebenen Schlüssel übereinstimmen.

Die folgende Abfrage zeigt Ihnen alle Fehler einer Aufgabe an, indem die Tabelle `awsdms_control.awsdms_validation_failures_v1` abgefragt wird. Der Aufgabenname sollte die externe Ressourcen-ID der Aufgabe sein. Die externe Ressourcen-ID der Aufgabe ist der letzte Wert des Aufgaben-ARNs. Bei einer Aufgabe mit einem ARN-Wert von `arn:aws:dms:us-west-2:5599:task:VFPFKH4FJR3FTYKK2RYSI` wäre die externe Ressourcen-ID beispielsweise `VFPFKH4FJR3FTYKK2RYSI`.

```
select * from awsdms_validation_failures_v1 where TASK_NAME = 'VFPFKH4FJR3FTYKK2RYSI'
```

```
TASK_NAME      VFPFKH4FJR3FTYKK2RYSI
TABLE_OWNER    DB2PERF
TABLE_NAME     PERFTTEST
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'}],]
```

Sie können sich das Feld DETAILS ansehen, um zu ermitteln, welche Spalten nicht übereinstimmen. Da Sie den Primärschlüssel des fehlgeschlagenen Datensatzes haben, können Sie die Quell- und Zielpunkte abfragen, um zu erfahren, welcher Teil des Datensatzes nicht übereinstimmt.

Redshift-Validierungsleistung

Amazon Redshift unterscheidet sich in mehrfacher Hinsicht von relationalen Datenbanken, darunter bei spaltenbasierter Speicherung, MPP, Datenkomprimierung und anderen Faktoren. Diese Unterschiede geben Redshift ein anderes Leistungsprofil als relationalen Datenbanken.

Während der Volllast-Replikationsphase verwendet die Validierung Bereichsabfragen, wobei die Datengröße durch die Einstellung `PartitionSize` bestimmt wird. Diese bereichsbasierten Abfragen wählen alle Datensätze aus der Quelltable aus.

Bei einer fortlaufenden Replikation wechseln Abfragen zwischen bereichsbasierten und individuellen Datensatzabrufen. Der Abfragetyp wird dynamisch anhand mehrerer Faktoren bestimmt, wie z. B. den folgenden:

- Volumen der Abfrage
- Typen von DML-Abfragen in der Quelltable
- Latenz der Aufgabe
- Gesamtzahl der Datensätze
- Validierungseinstellungen wie `PartitionSize`

Möglicherweise wird Ihr Amazon-Redshift-Cluster aufgrund von Validierungsabfragen zusätzlich belastet. Da die oben genannten Faktoren je nach Anwendungsfall variieren, müssen Sie die Leistung Ihrer Validierungsabfrage überprüfen und Ihren Cluster und Ihre Tabelle entsprechend anpassen. Einige Optionen zur Minderung von Leistungsproblemen sind die folgenden:

- Reduzieren Sie die Einstellungen `PartitionSize` und `ThreadCount`, um den Workload bei der Volllastvalidierung zu reduzieren. Beachten Sie, dass dies die Datenvalidierung verlangsamt.

- Redshift erzwingt zwar keine Primärschlüssel, AWS DMS stützt sich jedoch auf Primärschlüssel, um Datensätze auf dem Ziel für die Datenvalidierung eindeutig zu identifizieren. Wenn möglich, legen Sie den Primärschlüssel so fest, dass er dem Sortierschlüssel entspricht, so dass Abfragen zur Validierung bei vollständiger Auslastung schneller ausgeführt werden.

Einschränkungen

- Die Validierung erfordert, dass die Tabelle über einen Primärschlüssel oder eindeutigen Index verfügt.
 - Primärschlüsselspalten dürfen nicht vom Typ CLOB, BLOB oder BYTE sein.
 - Bei Primärschlüsselspalten vom Typ VARCHAR oder CHAR muss die Länge kleiner sein als 1024. Sie müssen die Länge im Datentyp angeben. Sie können keine unbegrenzten Datentypen als Primärschlüssel für die Datenüberprüfung verwenden.
 - Ein mit der Klausel NOVALIDATE erstellter Oracle-Schlüssel wird nicht als Primärschlüssel oder eindeutiger Index betrachtet.
 - Bei einer Oracle-Tabelle ohne Primärschlüssel und nur mit einem eindeutigen Schlüssel müssen die Spalten mit der eindeutigen Einschränkung ebenfalls über eine NOT NULL-Einschränkung verfügen.
- Die Validierung von NULL-PK/UK-Werten wird nicht unterstützt.
- Wenn die Kollation der Primärschlüsselspalte in der Ziel-PostgreSQL-Instance nicht auf "C" gesetzt ist, weicht die Sortierreihenfolge des Primärschlüssels von der Sortierreihenfolge in Oracle ab. Wenn sich die Sortierreihenfolgen in PostgreSQL und Oracle unterscheiden, kann die Datenvalidierung die Datensätze nicht validieren.
- Die Validierung generiert zusätzliche Abfragen für die Quell- und Zieldatenbanken. Sie müssen dafür sorgen, dass beide Datenbanken über ausreichend Ressourcen für diese zusätzliche Auslastung verfügen. Dies gilt insbesondere für Redshift-Ziele. Weitere Informationen finden Sie unter [Redshift-Validierungsleistung](#).
- Die Datenvalidierung wird nicht unterstützt, wenn mehrere Datenbanken zu einer zusammengeführt werden.
- Bei einem Quell- oder Ziel-Oracle-Endpunkt nutzt AWS DMS DBMS_CRYPTO zur Validierung von LOBs. Wenn Ihr Oracle-Endpunkt LOBs verwendet, müssen Sie dem Benutzerkonto, das für den Zugriff auf den Oracle-Endpunkt verwendet wird, die Ausführungserlaubnis auf dbms_crypto erteilen. Sie können dazu die folgende Anweisung ausführen:

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

- Wenn die Zieldatenbank während der Validierung außerhalb von AWS DMS geändert wird, werden Abweichungen möglicherweise nicht korrekt wiedergegeben. Dies kann der Fall sein, wenn eine Ihrer Anwendungen Daten in die Zieltabelle schreibt, während AWS DMS diese Tabelle validiert.
- Wenn eine oder mehrere Zeilen während der Validierung kontinuierlich geändert werden, kann AWS DMS diese Zeilen nicht validieren.
- Wenn AWS DMS feststellt, dass mehr als 10 000 Datensätze fehlschlagen oder ausgesetzt wurden, wird die Validierung gestoppt. Bevor Sie fortfahren, müssen Sie sämtliche zugrundeliegenden Datenprobleme lösen.
- AWS DMS unterstützt keine Datenvalidierung von Ansichten.
- AWS DMS unterstützt keine Datenvalidierung, wenn Aufgabeneinstellungen für die Zeichenersetzung verwendet werden.
- AWS DMS unterstützt die Validierung des OracleTyps LONG nicht.
- AWS DMS unterstützt die Validierung des Typs Oracle Spatial während einer heterogenen Migration nicht.

Informationen zu Einschränkungen bei Verwendung der S3-Zielvalidierung finden Sie unter [Einschränkungen bei Verwendung der S3-Zielvalidierung](#).

Zieldatenvalidierung in Amazon S3

AWS DMS unterstützt die Validierung replizierter Daten in Amazon-S3-Zielen. Da AWS DMS replizierte Daten in Amazon S3 als Flatfiles speichert, verwenden wir CREATE TABLE AS SELECT (CTAS)-Abfragen in [Amazon Athena](#) zur Datenvalidierung.

Abfragen von Daten, die in Amazon S3 gespeichert sind, sind rechenintensiv. Daher führt AWS DMS die Validierung von Amazon-S3-Daten während der Erfassung von Datenänderungen (Change Data Capture, CDC) nur einmal täglich durch, um Mitternacht (00:00 Uhr) UTC. Jede tägliche Validierung, die AWS DMS ausführt, wird als Intervallvalidierung bezeichnet. Während einer Intervallvalidierung validiert AWS DMS alle Änderungsdatensätze, die in den letzten 24 Stunden auf den Ziel-Bucket von Amazon S3 migriert wurden. Weitere Informationen zu Einschränkungen bei der Intervallvalidierung finden Sie unter [Einschränkungen bei Verwendung der S3-Zielvalidierung](#).

Die Amazon-S3-Zielvalidierung nutzt Amazon Athena. Daher fallen zusätzliche Kosten an. Weitere Informationen hierzu finden Sie unter [Preise zu Amazon Athena](#).

Note

Die S3-Zielvalidierung erfordert AWS DMS-Version 3.5.0 oder höher.

Themen

- [Voraussetzungen für die S3-Zielvalidierung](#)
- [Berechtigungen für die Verwendung der S3-Zielvalidierung](#)
- [Einschränkungen bei Verwendung der S3-Zielvalidierung](#)
- [Verwendung von reinen Validierungsaufgaben mit der S3-Zielvalidierung](#)

Voraussetzungen für die S3-Zielvalidierung

Bevor Sie die S3-Zielvalidierung verwenden, überprüfen Sie die folgenden Einstellungen und Berechtigungen:

- Setzen Sie den Wert `DataFormat` für die [S3Settings](#) Ihres Endpunkts auf `parquet`. Weitere Informationen finden Sie unter [Parquet-Einstellungen für S3](#).
- Vergewissern Sie sich, dass die dem Benutzerkonto zugewiesene Rolle, mit der die Migrationsaufgabe erstellt wurde, den richtigen Satz von Berechtigungen aufweist. Weitere Informationen finden Sie im Nachfolgenden unter [Berechtigungen](#).

Überprüfen Sie für Aufgaben, die die laufende Replikation (CDC) verwenden, die folgenden Einstellungen:

- Aktivieren Sie die zusätzliche Protokollierung, damit die CDC-Daten vollständige Datensätze enthalten. Informationen zum Aktivieren der zusätzlichen Protokollierung finden Sie unter [Automatisches Hinzufügen der zusätzlichen Protokollierung zu einem Oracle-Quellendpunkt](#) im Abschnitt [Unterstützung bei der Fehlersuche und Diagnose](#) dieses Handbuchs.
- Legen Sie den Parameter `TimestampColumnName` für den Zielendpunkt fest. Es gibt keine Einschränkungen hinsichtlich des Namens der Zeitstempelspalte. Weitere Informationen finden Sie unter [S3Settings](#).
- Richten Sie eine datumsbasierte Ordnerpartitionierung für das Ziel ein. Weitere Informationen finden Sie unter [Verwenden einer datumsbasierten Ordnerpartitionierung](#).

Berechtigungen für die Verwendung der S3-Zielvalidierung

Wenn Sie Zugriff für die Verwendung der S3-Zielvalidierung einrichten möchten, vergewissern Sie sich, dass die dem Benutzerkonto zugewiesene Rolle, mit der die Migrationsaufgabe erstellt wurde, den folgenden Satz von Berechtigungen aufweist. Ersetzen Sie die Beispielwerte durch Ihre Werte.

```
{
  "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>/*"
    ]
  }
]
```

Einschränkungen bei Verwendung der S3-Zielvalidierung

Sehen Sie sich die folgenden zusätzlichen Einschränkungen an, die bei Verwendung der S3-Zielvalidierung gelten. Informationen zu Einschränkungen, die für alle Validierungen gelten, finden Sie unter [Einschränkungen](#).

- Ihr Wert `DatePartitionSequence` erfordert eine Komponente für den Tag. Die S3-Zielvalidierung unterstützt das Format `YYYYMM` nicht.
- Wenn die Intervallvalidierung während CDC ausgeführt wird, werden in der Tabelle `awsdms_validation_failures_v1` möglicherweise falsche Validierungsfehler angezeigt. Diese Fehler treten auf, weil AWS DMS Änderungen, die während der Intervallvalidierung eingegangen sind, in den Partitionsordner des nächsten Tages migriert. Üblicherweise werden diese Änderungen in den Partitionsordner des aktuellen Tages geschrieben. Diese falschen Fehler stellen eine Einschränkung bei der Validierung der Replikation von einer dynamischen Quelldatenbank zu einem statischen Ziel wie beispielsweise Amazon S3 dar. Wenn Sie diese falschen Fehler untersuchen möchten, suchen Sie nach Datensätzen am Ende des Validierungsfensters (00:00 UTC), wenn diese Fehler in der Regel auftreten.

Stellen Sie sicher, dass der Wert `CDCLatencySource` für die Aufgabe niedrig ist, um die Anzahl falscher Fehler zu minimieren. Weitere Informationen zur Überwachung der Latenz finden Sie unter [Metriken für die Replikationsaufgabe](#).

- Bei Aufgaben mit dem Status `failed` oder `stopped` werden die Änderungen des Vortages nicht validiert. Erstellen Sie separate reine Validierungsaufgaben mit den gleichen Tabellenzuordnungen

sowie den gleichen Quell- und Zielendpunkten, um Validierungsfehler aufgrund unerwarteter Fehler zu minimieren. Weitere Informationen zu reinen Validierungsaufgaben finden Sie unter [Verwendung von reinen Validierungsaufgaben mit der S3-Zielvalidierung](#).

- Die Spalte Validierungsstatus in der Tabellenstatistik gibt den Status der letzten Intervallvalidierung wieder. Daher kann es vorkommen, dass eine Tabelle mit Nichtübereinstimmungen nach der Intervallvalidierung am nächsten Tag als validiert angezeigt wird. Überprüfen Sie `s3_validation_failures` folder im Amazon-S3-Ziel-Bucket auf Nichtübereinstimmungen, die vor mehr als einem Tag aufgetreten sind.
- S3 Validation verwendet die Bucket-Tabellenfunktion von Amazon Athena . Dadurch kann die S3-Validierung eine bucketierte Kopie der Zieltabellendaten erstellen. Das bedeutet, dass die Kopie der Tabellendaten in Teilmengen unterteilt ist, die der internen Partitionierung der DMS-Validierung entsprechen. Athena-Bucket-Tabellen haben ein Limit von 100.000 Buckets. Alle Tabellen, die die S3-Validierung zu validieren versucht, die dieses Limit überschreiten, schlagen bei der Validierung fehl. Die Anzahl der Buckets, die die S3-Validierung zu erstellen versucht, entspricht der folgenden Anzahl:

$$(\text{\#records in the table}) / (\text{validation partition size setting})$$

Um diese Einschränkung zu umgehen, erhöhen Sie die Einstellung für die Größe der Validierungspartition, sodass die Anzahl der von S3 Validation erstellten Buckets weniger als 100.000 beträgt. Weitere Informationen zum Bucketing finden Sie unter [Partitionieren und Bucketing in Athena](#) im Amazon Athena-Benutzerhandbuch.

Verwendung von reinen Validierungsaufgaben mit der S3-Zielvalidierung

Bei einer reinen Validierungsaufgabe wird eine Validierung der zu migrierenden Daten ausgeführt, ohne dass die Migration ausgeführt wird.

Reine Validierungsaufgaben werden auch dann weiter ausgeführt, wenn die Migrationsaufgabe beendet wird. Dadurch wird sichergestellt, dass AWS DMS das Intervallvalidierungsfenster 00:00 UTC nicht verpasst.

Für reine Validierungsaufgaben mit Ziel-Endpunkten von Amazon S3 gelten die folgenden Einschränkungen:

- Amazon-S3-Validierungsaufgaben für Volllast mit aktivierter Einstellung Nur Validierung werden unterstützt, funktionieren aber anders als Volllast- und reine Validierungsaufgaben für andere

Endpunkte. Wenn S3 ein Ziel ist, führt eine Aufgabe dieses Typs die Validierung ausschließlich anhand der Volllast-Daten im S3-Ziel durch und nicht anhand von Daten, die im Rahmen einer CDC-Migration migriert wurden. Verwenden Sie dieses Feature nur, um Daten zu validieren, die von einer reinen Volllastaufgabe erstellt wurden. Wenn dieser Modus verwendet wird, um Daten in einem Ziel zu validieren, auf dem eine aktive CDC-Aufgabe ausgeführt wird, kommt es nicht zu einer effektiven Validierung.

- Bei reinen Validierungsaufgaben werden nur Änderungen seit dem letzten Intervallvalidierungsfenster (00:00 UTC) überprüft. Bei reinen Validierungsaufgaben werden keine Volllastdaten oder CDC-Daten aus vorherigen Tagen validiert.

Tagging von Ressourcen im AWS Database Migration Service

Mithilfe von Tags können Sie in AWS Database Migration Service (AWS DMS) Metadaten zu Ihren Ressourcen hinzufügen. Darüber hinaus können Sie diese Tags mit AWS Identity and Access Management (IAM)-Richtlinien verwenden, um den Zugriff auf AWS-DMS-Ressourcen zu verwalten und zu steuern, welche Aktionen auf die AWS-DMS-Ressourcen angewendet werden können. Außerdem können Sie mit diesen Tags auch Kosten verfolgen, indem Ausgaben für ähnlich getaggte Ressourcen gruppiert werden.

Alle AWS-DMS-Ressourcen können mit Tags versehen werden:

- Zertifikate
- Datenanbieter
- Datenmigrationen
- Endpunkte
- Ereignisabonnements
- Instance-Profile
- Migrationsprojekte
- Replikations-Instances
- Replikations-Subnetzgruppen
- Replikationsaufgaben

Ein AWS-DMS-Tag ist ein Name-Wert-Paar, das Sie definieren und mit einer AWS-DMS-Ressource verknüpfen. Der Name wird als Schlüssel bezeichnet. Die Angabe eines Wertes für den Schlüssel ist optional. Sie können Tags verwenden, um einer AWS-DMS-Ressource beliebige Informationen zuzuweisen. Ein Tag-Schlüssel könnte z. B. verwendet werden, um eine Kategorie zu definieren, und ein Tag-Wert könnte ein Element in dieser Kategorie sein. Sie können beispielsweise den Tag-Schlüssel „Projekt“ und den Tag-Wert „Salix“ definieren. Dies bedeutet, dass die AWS-DMS-Ressource dem Projekt „Salix“ zugewiesen ist. Sie können mit Tags auch AWS-DMS-Ressourcen kennzeichnen, die zu Test- oder Produktionszwecken verwendet werden, indem Sie einen Schlüssel wie z. B. Umgebung = Test oder Umgebung = Produktion verwenden. Wir empfehlen, einheitliche Tag-Schlüssel zu verwenden, um die mit AWS-DMS-Ressourcen verknüpften Metadaten einfacher verfolgen zu können.

Sie können Tags auch zum Organisieren Ihrer AWS-Kontorechnung verwenden, um Ihre eigene Kostenstruktur darzustellen. Dazu müssen Sie sich registrieren, um Ihre AWS-Konto-Rechnung mit Tag-Schlüsselwerten zu erhalten. Um dann die Kosten kombinierter Ressourcen anzuzeigen, organisieren Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag-Schlüsselwerten. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können. Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#) im AWS Billing-Benutzerhandbuch.

Jede AWS-DMS-Ressource verfügt über einen Tag-Satz, der alle Tags enthält, die dieser AWS-DMS-Ressource zugewiesen sind. Ein Tag-Satz kann bis zu zehn Tags enthalten oder leer sein. Wenn Sie einer AWS-DMS-Ressource ein Tag hinzufügen, das denselben Schlüssel hat wie ein bereits vorhandenes Tag der Ressource, überschreibt der neue Wert den alten.

AWS wendet auf Ihre Tags keine semantische Bedeutung an. Tags werden als reine Zeichenfolgen interpretiert. AWS DMS kann Tags für eine AWS-DMS-Ressource festlegen, je nach den Einstellungen, die Sie beim Erstellen der Ressource verwenden.

In der folgenden Liste werden die Merkmale eines AWS-DMS-Tags beschrieben.

- Der Tag-Schlüssel ist der erforderliche Name des Tags. Der Zeichenfolgenwert kann aus 1 bis 128 Unicode-Zeichen bestehen. Ihm darf kein "aws:" oder "dms:" vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie '_', ':', '/', '=', '+', '-' enthalten (Java-Regex: `"^([\p{L}\p{Z}\p{N}_.: /+=\ -]*)$"`).
- Der Tag-Wert ist ein optionaler Zeichenfolgenwert des Tags. Der Zeichenfolgenwert kann aus 1 bis 256 Unicode-Zeichen bestehen. Ihm darf kein "aws:" oder "dms:" vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie '_', ':', '/', '=', '+', '-' enthalten (Java-Regex: `"^([\p{L}\p{Z}\p{N}_.: /+=\ -]*)$"`).

Die Werte innerhalb eines Tag-Satzes müssen nicht eindeutig und können null sein. Sie können beispielsweise über ein Schlüssel-Wert-Paar in einem Tag-Satz "Projekt/Trinity" und "Kostenstelle/Trinity" verfügen.

Sie können die AWS CLI oder die AWS-DMS-API verwenden, um Tags für AWS-DMS-Ressourcen hinzuzufügen, aufzulisten oder zu löschen. Bei der AWS CLI- oder der AWS-DMS-API müssen Sie den Amazon-Ressourcennamen (ARN) für die zu verwendende AWS-Ressource angeben.

Weitere Informationen zum Erstellen eines ARN finden Sie unter [Erstellung eines Amazon-Ressourcennamens \(ARN\) für AWS DMS](#).

Beachten Sie, dass Tags für Autorisierungszwecke zwischengespeichert werden. Daher können Ergänzungen und Aktualisierungen für Tags von AWS-DMS-Ressourcen einige Minuten in Anspruch nehmen, bevor sie verfügbar sind.

API

Mithilfe der AWS-DMS-API können Sie Tags für eine AWS-DMS-Ressource hinzufügen, auflisten oder entfernen.

- Verwenden Sie die API-Operation [AddTagsToResource](#), um ein Tag zu einer AWS-DMS-Ressource hinzuzufügen.
- Um die einer AWS-DMS-Ressource zugewiesenen Tags aufzulisten, verwenden Sie den Vorgang [ListTagsForResource](#).
- Verwenden Sie die API-Operation [RemoveTagsFromResource](#), um Tags aus einer AWS-DMS-Ressource zu entfernen.

Weitere Informationen zum Erstellen des erforderlichen ARN finden Sie unter [Erstellung eines Amazon-Ressourcennamens \(ARN\) für AWS DMS](#).

Wenn Sie mit der AWS-DMS-API mit XML arbeiten, nutzen Sie das folgende Schema:

```
<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Trinity</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>Jones</Value>
    </Tag>
  </TagSet>
</Tagging>
```

Die folgende Tabelle enthält eine Liste der zulässigen XML-Tags und deren Eigenschaften. Beachten Sie, dass für die Werte für Schlüssel und Wert zwischen Klein- und Großbuchstaben unterschieden wird. Zum Beispiel sind "project = Trinity" und "PROJECT = Trinity" zwei verschiedene Tags.

Markieren von Elementen	Beschreibung
Tag-Satz	Ein Tag-Satz ist ein Container für alle Tags, die einer Amazon Amazon RDS-Ressource zugewiesen sind. Es ist nur ein Tag-Satz pro Ressource zulässig. Sie arbeiten mit einem TagSet ausschließlich über die AWS-DMS-API.
Tag	Ein Tag ist ein benutzerdefiniertes Schlüssel-Wert-Paar. Ein Tag-Satz kann 1 bis 10 Tags enthalten.
Schlüssel	<p>Ein Schlüssel ist der erforderliche Name des Tags. Der Zeichenfolgenwert kann aus 1 bis 128 Unicode-Zeichen bestehen. Ihm darf kein "aws:" oder "dms:" vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie '_', '!', '/', '=', '+', '-' enthalten (Java-Regex: <code>"^([\p{L}\p{Z}\p{N}_.:/+\\-]*)\$"</code>).</p> <p>Schlüssel müssen in einem Tag-Satz eindeutig sein. Sie können z. B. in einem Tag-Satz kein Schlüsselpaar mit gleichem Schlüssel, aber unterschiedlichen Werten verwenden, wie "Projekt/Trinity" und "Projekt/Xanadu".</p>
Value	<p>Ein Wert ist der optionale Wert des Tags. Der Zeichenfolgenwert kann aus 1 bis 256 Unicode-Zeichen bestehen. Ihm darf kein "aws:" oder "dms:" vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie '_', '!', '/', '=', '+', '-' enthalten (Java-Regex: <code>"^([\p{L}\p{Z}\p{N}_.:/+\\-]*)\$"</code>).</p> <p>Die Werte innerhalb eines Tag-Satzes müssen nicht eindeutig und können null sein. Sie können beispielsweise über ein Schlüssel-Wert-Paar in einem Tag-Satz "Projekt/Trinity" und "Kostenstelle/Trinity" verfügen.</p>

Sicherheit in AWS Database Migration Service

Cloud-Sicherheit hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die eingerichtet wurde, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS-Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für AWS DMS gelten, finden Sie unter [Vom Compliance-Programm abgedeckte AWS-Services](#).
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von AWS DMS einsetzen können. Die folgenden Themen veranschaulichen, wie Sie AWS DMS zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren außerdem, wie Sie andere AWS-Services verwenden, um Ihre AWS DMS-Ressourcen zu überwachen und zu schützen.

Sie können den Zugriff auf Ihre AWS DMS-Ressourcen und Ihre Datenbanken (DBs) verwalten. Die Methode, die Sie zum Verwalten des Zugriffs verwenden, hängt von der Replikationsaufgabe ab, die Sie mit AWS DMS ausführen müssen:

- Verwenden Sie AWS Identity and Access Management (IAM)-Richtlinien, um Berechtigungen zuzuweisen, die bestimmen, wer AWS DMS-Ressourcen verwalten darf. AWS DMS erfordert, dass Sie über die entsprechenden Berechtigungen verfügen, wenn Sie sich als IAM-Benutzer anmelden. Beispielsweise können Sie IAM verwenden, um zu bestimmen, wer DB-Instances und -Cluster erstellen, beschreiben, ändern und löschen, Ressourcen markieren oder Sicherheitsgruppen ändern darf. Weitere Informationen zu IAM und seiner Verwendung mit AWS DMS finden Sie unter [Identity and Access Management für AWS Database Migration Service](#).

- AWS DMS verwendet Secure Sockets Layer (SSL) für Ihre Endpunktverbindungen mit Transport Layer Security (TLS). Weitere Informationen zur Verwendung von SSL/TLS mit AWS DMS finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).
- AWS DMS verwendet AWS Key Management Service (AWS KMS)-Verschlüsselungsschlüssel, um den von Ihrer Replikations-Instance verwendeten Speicher und deren Endpunktverbindungsinformationen zu verschlüsseln. AWS DMS verwendet auch AWS KMS-Verschlüsselungsschlüssel, um Ihre Zieldaten im Ruhezustand für Amazon-S3- und Amazon-Redshift-Zielendpunkte zu sichern. Weitere Informationen finden Sie unter [Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen](#).
- AWS DMS erstellt Ihre Replikations-Instance immer in einer Virtual Private Cloud (VPC) basierend auf dem Amazon-VPC-Service für die größtmögliche Netzwerkzugriffskontrolle. Verwenden Sie für Ihre DB-Instances und Instance-Cluster dieselbe VPC wie Ihre Replikations-Instance oder zusätzliche VPCs, um diese Zugriffskontrolle zu erreichen. Jede von Ihnen verwendete Amazon VPC muss einer Sicherheitsgruppe zugeordnet sein, die Regeln enthält, mit denen der gesamte Datenverkehr auf allen Ports die VPC verlassen kann. Dieser Ansatz ermöglicht die Kommunikation von der Replikations-Instance zu Ihren Quell- und Zieldatenbankendpunkten, sofern auf diesen Endpunkten korrekte Eingangsregeln aktiviert sind.

Weitere Hinweise zu verfügbaren Netzwerkkonfigurationen für AWS DMS finden Sie unter [Einrichten eines Netzwerks für eine Replikations-Instance](#). Weitere Informationen zum Erstellen einer DB-Instance oder eines Instance-Clusters in einer VPC finden Sie in der Dokumentation zur Sicherheits- und Cluster-Verwaltung für Ihre Amazon-Datenbanken in der [AWS-Dokumentation](#). Weitere Hinweise zu von AWS DMS unterstützten Netzwerkkonfigurationen finden Sie unter [Einrichten eines Netzwerks für eine Replikations-Instance](#).

- Wenn Sie Datenbankmigrationsprotokolle anzeigen möchten, müssen Sie über die entsprechenden Amazon-CloudWatch-Logs-Berechtigungen für die IAM-Rolle, die Sie verwenden, verfügen. Weitere Informationen zur Protokollierung für AWS DMS finden Sie unter [Überwachen von Replikationsaufgaben mit Amazon CloudWatch](#).

Themen

- [Datenschutz in AWS Database Migration Service](#)
- [Identity and Access Management für AWS Database Migration Service](#)
- [Compliance-Validierung für AWS Database Migration Service](#)
- [Ausfallsicherheit in AWS Database Migration Service](#)
- [Sicherheit der Infrastruktur in AWS Database Migration Service](#)

- [Differenzierte Zugriffskontrolle mit Ressourcennamen und Ressourcen-Tags](#)
- [Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen](#)
- [Netzwerksicherheit für AWS Database Migration Service](#)
- [Verwenden von SSL mit AWS Database Migration Service](#)
- [Ändern des Datenbankpassworts](#)

Datenschutz in AWS Database Migration Service

Datenverschlüsselung

Sie können die Verschlüsselung für Datenressourcen unterstützter AWS DMS-Zielendpunkte aktivieren. AWS DMS verschlüsselt auch Verbindungen zu AWS DMS und zwischen AWS DMS und allen seinen Quell- und Zielendpunkten. Darüber hinaus können Sie die Schlüssel verwalten, die AWS DMS und die unterstützten Zielendpunkte verwenden, um diese Verschlüsselung zu aktivieren.

Themen

- [Verschlüsselung im Ruhezustand](#)
- [Verschlüsselung während der Übertragung](#)
- [Schlüsselverwaltung](#)

Verschlüsselung im Ruhezustand

AWS DMS unterstützt die Verschlüsselung im Ruhezustand, indem Sie den serverseitigen Verschlüsselungsmodus angeben können, in dem Sie die replizierten Daten zu Amazon S3 übertragen möchten, bevor sie auf unterstützte AWS DMS-Zielendpunkte kopiert werden. Sie können diesen Verschlüsselungsmodus angeben, indem Sie das zusätzliche `encryptionMode`-Verbindungsattribut für den Endpunkt festlegen. Wenn diese `encryptionMode`-Einstellung den KMS-Schlüsselverschlüsselungsmodus angibt, können Sie auch benutzerdefinierte AWS KMS-Schlüssel speziell zum Verschlüsseln der Zieldaten für die folgenden AWS DMS-Zielendpunkte erstellen:

- Amazon Redshift – Weitere Informationen zur Einstellung von `encryptionMode` finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon Redshift als Ziel für AWS DMS](#). Weitere Informationen zum Erstellen eines benutzerdefinierten AWS KMS-Verschlüsselungsschlüssels finden Sie unter [Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten](#).
- Amazon S3 – Weitere Informationen zur Einstellung von `encryptionMode` finden Sie unter [Endpunkteinstellungen bei Verwendung von Amazon S3 als Ziel für AWS DMS](#). Weitere Informationen zum Erstellen eines benutzerdefinierten AWS KMS-Verschlüsselungsschlüssels finden Sie unter [Erstellen von AWS KMS-Schlüsseln zum Verschlüsseln von Amazon-S3-Zielobjekten](#).

Verschlüsselung während der Übertragung

AWS DMS unterstützt die Verschlüsselung während der Übertragung, indem sichergestellt wird, dass die replizierten Daten sicher vom Quellendpunkt zum Zielendpunkt verschoben werden. Dazu gehört das Verschlüsseln eines S3-Buckets auf der Replikations-Instance, die Ihre Replikationsaufgabe für Zwischenspeicher verwendet, wenn die Daten durch die Replikationspipeline verschoben werden. Zum Verschlüsseln von Aufgabenverbindungen zu Quell- und Zielendpunkten verwendet AWS DMS Secure Socket Layer (SSL) oder Transport Layer Security (TLS). Durch die Verschlüsselung von Verbindungen zu beiden Endpunkten stellt AWS DMS sicher, dass Ihre Daten sicher sind, wenn sie sowohl vom Quellendpunkt zur Replikationsaufgabe als auch von der Aufgabe zum Zielendpunkt verschoben werden. Weitere Informationen zur Verwendung von SSL/TLS mit AWS DMS finden Sie unter [Verwenden von SSL mit AWS Database Migration Service](#).

AWS DMS unterstützt sowohl Standard- als auch benutzerdefinierte Schlüssel, um sowohl Zwischenreplikationsspeicher als auch Verbindungsinformationen zu verschlüsseln. Sie verwalten diese Schlüssel mit AWS KMS. Weitere Informationen finden Sie unter [Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen](#).

Schlüsselverwaltung

AWS DMS unterstützt Standard- oder benutzerdefinierte Schlüssel zum Verschlüsseln des Replikationsspeichers, der Verbindungsinformationen und des Zieldatenspeichers für bestimmte Zielendpunkte. Sie verwalten diese Schlüssel mit AWS KMS. Weitere Informationen finden Sie unter [Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen](#).

Richtlinie für den Datenverkehr zwischen Netzwerken

Verbindungen werden mit einem Schutz zwischen AWS DMS-Quell- und Zielendpunkten in derselben AWS-Region bereitgestellt, unabhängig davon, ob sie On-Premise oder als Teil eines AWS-Services in der Cloud ausgeführt werden. (Mindestens ein Endpunkt, eine Quelle oder ein Ziel muss als Teil eines AWS-Services in der Cloud ausgeführt werden.) Dieser Schutz gilt, unabhängig davon, ob diese Komponenten dieselbe Virtual Private Cloud (VPC) verwenden oder in separaten VPCs existieren, wenn sich die VPCs alle in derselben AWS-Region befinden. Weitere Hinweise zu den unterstützten Netzwerkkonfigurationen für AWS DMS finden Sie unter [Einrichten eines Netzwerks für eine Replikations-Instance](#). Weitere Hinweise zu Sicherheitsüberlegungen bei der Verwendung dieser Netzwerkkonfigurationen finden Sie unter [Netzwerksicherheit für AWS Database Migration Service](#).

Datenschutz in DMS Fleet Advisor

DMS Fleet Advisor erfasst und analysiert Ihre Datenbank-Metadaten, um die korrekte Größe des Migrationsziels zu ermitteln. DMS Fleet Advisor greift nicht auf Daten in Ihren Tabellen zu und überträgt sie nicht. Außerdem verfolgt DMS Fleet Advisor nicht die Nutzung der Datenbank-Features und greift nicht auf Ihre Nutzungsstatistiken zu.

Sie kontrollieren den Zugriff auf Ihre Datenbanken, wenn Sie Datenbankbenutzer erstellen, die DMS Fleet Advisor für die Arbeit mit Ihren Datenbanken verwendet. Sie gewähren diesen Benutzern die erforderlichen Berechtigungen. Um DMS Fleet Advisor verwenden zu können, gewähren Sie Ihren Datenbankbenutzern Leseberechtigungen. DMS Fleet Advisor verändert Ihre Datenbanken nicht und benötigt keine Schreibberechtigungen. Weitere Informationen finden Sie unter [Erstellen von Datenbankbenutzern für AWS DMS Fleet Advisor](#).

Sie können Datenverschlüsselung in Ihren Datenbanken verwenden. AWS DMS verschlüsselt auch Verbindungen innerhalb von DMS Fleet Advisor und innerhalb seiner Datenkollektoren.

Der DMS-Datenkollektor verwendet Data Protection Application Programming Interface (DPAPI), um Informationen über die Umgebung und die Datenbankanmeldedaten des Kunden zu verschlüsseln, zu schützen und zu speichern. DMS Fleet Advisor speichert diese verschlüsselten Daten in einer Datei auf dem Server, auf dem Ihr DMS-Datenkollektor arbeitet. DMS Fleet Advisor überträgt diese Daten nicht von diesem Server. Weitere Informationen zu DPAPI finden Sie unter [Verwendung des Datenschutzes](#).

Nachdem Sie den DMS-Datenkollektor installiert haben, können Sie alle Abfragen anzeigen, die diese Anwendung zur Erfassung von Metriken ausführt. Sie können den DMS-Datenkollektor im Offline-Modus ausführen und dann die erfassten Daten auf Ihrem Server überprüfen. Sie können diese erfassten Daten auch in Ihrem Amazon-S3-Bucket überprüfen. Weitere Informationen finden Sie unter [Wie funktioniert der DMS-Datenkollektor?](#).

Identity and Access Management für AWS Database Migration Service

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem ein Administrator den Zugriff auf - AWS Ressourcen sicher steuern kann. IAM-Administratoren steuern, wer für die Nutzung von - AWS DMS Ressourcen authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) werden kann. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Funktionsweise AWS Database Migration Service von mit IAM](#)
- [AWS Database Migration Service Beispiele für identitätsbasierte Richtlinien](#)
- [Beispiele für ressourcenbasierte Richtlinien für AWS KMS](#)
- [Verwenden von Secrets für den Zugriff auf AWS Database Migration Service -Endpunkte](#)
- [Verwenden von serviceverknüpften Rollen für AWS DMS](#)
- [Fehlerbehebung für AWS Database Migration Service Identität und Zugriff](#)
- [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#)
- [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [AWS Von verwaltete Richtlinien für AWS Database Migration Service](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit in AWS DMS.

Service-Benutzer – Wenn Sie den AWS DMS Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie für Ihre Arbeit weitere AWS DMS Funktionen ausführen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Unter

[Fehlerbehebung für AWS Database Migration Service Identität und Zugriff](#) finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Funktion in AWS DMS haben.

Service-Administrator – Wenn Sie in Ihrem Unternehmen für AWS DMS Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollständigen Zugriff auf AWS DMS. Ihre Aufgabe besteht darin, zu bestimmen, auf welche AWS DMS Funktionen und Ressourcen Ihre Service-Benutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit verwenden kann AWS DMS, finden Sie unter [Funktionsweise AWS Database Migration Service von mit IAM](#).

IAM-Administrator: Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf AWS DMS verfassen können. Beispiele für AWS DMS identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter [AWS Database Migration Service Beispiele für identitätsbasierte Richtlinien](#).

Authentifizierung mit Identitäten

Die Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten bei anmelden. Sie müssen als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle authentifiziert (bei angemeldet AWS) sein.

Sie können sich bei AWS als Verbundidentität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt werden. AWS IAM Identity Center (IAM Identity Center)-Benutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie AWS über einen Verbund auf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, um welchen Benutzertyp es sich handelt, können Sie sich bei der AWS Management Console oder im - AWS Zugriffsportal anmelden. Weitere Informationen zur Anmeldung bei AWS finden Sie unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung - Benutzerhandbuch.

Wenn Sie AWS programmgesteuert auf zugreifen, AWS stellt ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (Command Line Interface, CLI) bereit, um Ihre Anforderungen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der

empfohlenen Methode zum eigenständigen Signieren von Anforderungen finden Sie unter [Signieren von AWS API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. empfiehlt beispielsweise, AWS die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet und Sie melden sich mit der E-Mail-Adresse und dem Passwort an, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen.

Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die [Rollen wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI - oder AWS -API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können AWS-Services Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.
- **Serviceübergreifender Zugriff** – Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann

dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.

- **Forward Access Sessions (FAS)** – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen in auszuführen AWS, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung AWS-Service , Anforderungen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Service eine Anfrage erhält, die Interaktionen mit anderen AWS-Services oder -Ressourcen erfordert. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle:** Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Serviceverknüpfte Rolle** – Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem angezeigt AWS-Konto und gehören dem Service. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen, die auf Amazon EC2 ausgeführt werden** – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und - AWS CLI oder AWS -API-Anforderungen stellen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine - AWS Rolle zuzuweisen und sie für alle ihre Anwendungen verfügbar zu machen, erstellen Sie ein Instance-Profil, das der Instance zugeordnet ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff in , AWS indem Sie Richtlinien erstellen und sie an AWS Identitäten oder Ressourcen anfügen. Eine Richtlinie ist ein Objekt in , AWS das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem anfügen können AWS-Konto. Verwaltete Richtlinien umfassen AWS -verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie

wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder umfassen AWS-Services.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services AWS WAF, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter [Zugriffssteuerungsliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch

Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.

- Service-Kontrollrichtlinien (SCPs) – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in angeben AWS Organizations. AWS Organizations ist ein Service zum Gruppieren und zentralen Verwalten mehrerer AWS-Konten, die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Die SCP beschränkt Berechtigungen für Entitäten in Mitgliedskonten, einschließlich jeder Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS bestimmt, ob eine Anforderung zugelassen werden soll, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter [Logik zur Richtlinienbewertung](#) im IAM-Benutzerhandbuch.

Funktionsweise AWS Database Migration Service von mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf zu verwalten AWS DMS, sollten Sie verstehen, welche IAM-Funktionen Sie mit verwenden können AWS DMS. Einen Überblick über das Zusammenwirken von AWS DMS und anderen - AWS Services mit IAM finden Sie unter [-AWS Services, die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Themen

- [Identitätsbasierte AWS DMS -Richtlinien](#)
- [Ressourcenbasierte AWS DMS -Richtlinien](#)

- [Autorisierung auf der Basis von AWS DMS -Tags](#)
- [IAM-Rollen für AWS DMS](#)
- [Identitäts- und Zugriffsverwaltung für DMS Fleet Advisor](#)

Identitätsbasierte AWS DMS -Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie festlegen, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. AWS DMS unterstützt spezifische Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Aktionen

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben in der Regel denselben Namen wie die zugehörige AWS API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in AWS DMS verwenden das folgende Präfix vor der Aktion: `dms :`

Um beispielsweise jemandem die Berechtigung zum Erstellen einer Replikationsaufgabe mit der AWS DMS `CreateReplicationTask` -API-Operation zu erteilen, fügen Sie die `dms:CreateReplicationTask` Aktion in seine Richtlinie ein. Richtlinienanweisungen müssen entweder ein `-Action` oder `-NotActionElement` enthalten. AWS DMS definiert einen eigenen Satz von Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service durchführen können.

Um mehrere -Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie folgendermaßen durch Kommas.

```
"Action": [  
  "dms:action1",  
  "dms:action2"
```

Sie können auch Platzhalter (*) verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "dms:Describe*"
```

Eine Liste der AWS DMS Aktionen finden Sie unter [Von definierte Aktionen AWS Database Migration Service](#) im IAM-Benutzerhandbuch.

Ressourcen

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

AWS DMS funktioniert mit den folgenden Ressourcen:

- Zertifikate
- Endpunkte
- Ereignisabonnements

- Replikations-Instances
- Replikations-Subnetzgruppen (Sicherheit)
- Replikationsaufgaben

Welche Ressource oder Ressourcen AWS DMS benötigt, hängt von der Aktion oder den Aktionen ab, die Sie aufrufen. Sie benötigen eine Richtlinie, die diese Aktionen für die zugeordnete Ressource oder Ressourcen zulässt, die von den Ressourcen-ARNs angegeben werden.

Beispielsweise hat eine AWS DMS Endpunktresource den folgenden ARN:

```
arn:${Partition}:dms:${Region}:${Account}:endpoint/${InstanceId}
```

Weitere Informationen zum Format von ARNs finden Sie unter [Amazon-Ressourcennamen \(ARNs\) und AWS Service-Namespaces](#).

Um beispielsweise die 1A2B3C4D5E6F7G8H9I0J1K2L3M-Endpunkt-Instance für die use-east-2-Region in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN.

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/1A2B3C4D5E6F7G8H9I0J1K2L3M"
```

Um alle Endpunkte anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/*"
```

Einige AWS DMS Aktionen, z. B. das Erstellen von Ressourcen, können nicht für eine bestimmte Ressource ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"
```

Einige AWS DMS API-Aktionen umfassen mehrere Ressourcen. Beispielsweise startet `StartReplicationTask` eine Replikationsaufgabe und verbindet sie mit zwei Datenbankendpunktresourcen, einer Quelle und einem Ziel, sodass ein IAM-Benutzer Berechtigungen zum Lesen des Quellendpunkts und zum Schreiben auf den Zielendpunkt haben muss. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie die ARNs durch Kommata voneinander.

```
"Resource": [  
  "resource1",  
  "resource2" ]
```

Weitere Informationen zum Steuern des Zugriffs auf AWS DMS Ressourcen mithilfe von Richtlinien finden Sie unter [Verwenden von Ressourcennamen für die Zugriffskontrolle](#). Eine Liste der AWS DMS -Ressourcentypen und ihrer ARNs finden Sie unter [Von AWS Database Migration Service definierte Ressourcentypen](#) im IAM-Benutzerhandbuch. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von AWS Database Migration Service definierte Aktionen](#).

Bedingungsschlüssel

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. Informationen zum Anzeigen aller AWS globalen Bedingungsschlüssel finden Sie unter [AWS Globale Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

AWS DMS definiert einen eigenen Satz von Bedingungsschlüsseln und unterstützt auch die Verwendung einiger globaler Bedingungsschlüssel. Informationen zum Anzeigen aller AWS

globalen Bedingungsschlüssel finden Sie unter [AWS Globale Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

AWS DMS definiert eine Reihe von Standard-Tags, die Sie in seinen Bedingungsschlüsseln verwenden können, und ermöglicht Ihnen auch, Ihre eigenen benutzerdefinierten Tags zu definieren. Weitere Informationen finden Sie unter [Verwendung von Tags zur Zugriffssteuerung](#).

Eine Liste der AWS DMS Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für AWS Database Migration Service](#) im IAM-Benutzerhandbuch. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von AWS Database Migration Service definierte Aktionen](#) und [Von AWS Database Migration Service definierte Ressourcentypen](#).

Beispiele

Beispiele für AWS DMS identitätsbasierte Richtlinien finden Sie unter [AWS Database Migration Service Beispiele für identitätsbasierte Richtlinien](#).

Ressourcenbasierte AWS DMS -Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die angeben, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für eine bestimmte AWS DMS Ressource ausführen kann. AWS DMS unterstützt ressourcenbasierte Berechtigungsrichtlinien für AWS KMS Verschlüsselungsschlüssel, die Sie zum Verschlüsseln von Daten erstellen, die zu unterstützten Zielendpunkten migriert wurden. Die unterstützten Ziel-Endpunkte enthalten Amazon Redshift und Amazon S3. Mithilfe ressourcenbasierter Richtlinien können Sie anderen Konten für jeden Zielendpunkt die Berechtigung zur Verwendung dieser Verschlüsselungsschlüssel erteilen.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als [Prinzipal in einer ressourcenbasierten Richtlinie](#) angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource in verschiedenen AWS Konten befinden, müssen Sie der Prinzipal-Entität auch die Berechtigung für den Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der Entität eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden](#) im IAM-Benutzerhandbuch.

Der AWS DMS Service unterstützt nur eine Art von ressourcenbasierter Richtlinie, die als Schlüsselrichtlinie bezeichnet wird und an einen AWS KMS Verschlüsselungsschlüssel angehängt ist. Diese Richtlinie legt fest, welche Prinzipal-Entitäten (Konten, Benutzer, Rollen und verbundene Benutzer) migrierte Daten auf dem unterstützten Zielendpunkt verschlüsseln können.

Informationen zum Anfügen einer ressourcenbasierten Richtlinie an einen Verschlüsselungsschlüssel, den Sie für die unterstützten Zielendpunkte erstellen, finden Sie unter [Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten](#) und [Erstellen von AWS KMS-Schlüsseln zum Verschlüsseln von Amazon-S3-Zielobjekten](#).

Beispiele

Beispiele für AWS DMS ressourcenbasierte Richtlinien finden Sie unter [Beispiele für ressourcenbasierte Richtlinien für AWS KMS](#).

Autorisierung auf der Basis von AWS DMS -Tags

Sie können Tags an AWS DMS Ressourcen anfügen oder Tags in einer Anforderung an übergeben AWS DMS. Um den Zugriff basierend auf Tags zu steuern, geben Sie Tag-Informationen im [Bedingungselement](#) einer Richtlinie mit dem `aws:TagKeys` Bedingungsschlüssel `dms:ResourceTag/key-name`, oder an. AWS DMS definiert eine Reihe von Standard-Tags, die Sie in seinen Bedingungsschlüsseln verwenden können, und ermöglicht Ihnen auch `aws:RequestTag/key-name`, Ihre eigenen benutzerdefinierten Tags zu definieren. Weitere Informationen finden Sie unter [Verwendung von Tags zur Zugriffssteuerung](#).

Eine identitätsbasierte Beispielrichtlinie, die den Zugriff auf eine Ressource auf Tags beschränkt, finden Sie unter [Zugreifen auf AWS DMS -Ressourcen basierend auf Tags](#).

IAM-Rollen für AWS DMS

Eine [IAM-Rolle](#) ist eine Entität in Ihrem AWS Konto mit bestimmten Berechtigungen.

Verwenden temporärer Anmeldeinformationen mit AWS DMS

Sie können temporäre Anmeldeinformationen verwenden, um sich mit dem Verbund anzumelden, eine IAM-Rolle zu übernehmen oder eine kontoübergreifende Rolle zu übernehmen. Sie erhalten temporäre Sicherheitsanmeldeinformationen, indem Sie AWS STS -API-Operationen wie [AssumeRole](#) oder aufrufen [GetFederationToken](#).

AWS DMS unterstützt die Verwendung temporärer Anmeldeinformationen.

Service-verknüpfte Rollen

[Serviceverknüpfte Rollen](#) ermöglichen AWS es Services, auf Ressourcen in anderen Services zuzugreifen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Weitere Informationen zum Erstellen oder Verwalten von AWS DMS serviceverknüpften Rollen finden Sie unter [Verwenden von serviceverknüpften Rollen](#).

Service rollen

Dieses Feature ermöglicht einem Service das Annehmen einer [Service rolle](#) in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Service rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein IAM-Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Dienstes beeinträchtigen.

AWS DMS unterstützt zwei Arten von Service rollen, die Sie erstellen müssen, um bestimmte Quell- oder Zielendpunkte zu verwenden:

- Rollen mit Berechtigungen, um AWS DMS Zugriff auf die folgenden Quell- und Zielendpunkte (oder deren Ressourcen) zu gewähren:
 - Amazon DynamoDB als Ziel – weitere Informationen finden Sie unter [Voraussetzungen für die Verwendung von DynamoDB als Ziel für AWS Database Migration Service](#).
 - OpenSearch als Ziel – Weitere Informationen finden Sie unter [Voraussetzungen für die Verwendung von Amazon OpenSearch Service als Ziel für AWS Database Migration Service](#).
 - Amazon Kinesis als Ziel – weitere Informationen finden Sie unter [Voraussetzungen für die Verwendung eines Kinesis-Datenstroms als Ziel für AWS Database Migration Service](#).
 - Amazon Redshift als Ziel – Sie müssen die angegebene Rolle nur zum Erstellen eines benutzerdefinierten KMS-Verschlüsselungsschlüssels zum Verschlüsseln der Zieldaten oder zum Angeben eines benutzerdefinierten S3-Buckets für Zwischenaufgabenspeicher erstellen. Weitere Informationen finden Sie unter [Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten](#) oder [Einstellungen von Amazon-S3-Buckets](#).
 - Amazon S3 als Quelle oder als Ziel – weitere Informationen finden Sie unter [Voraussetzungen für die Verwendung von Amazon S3 als Quelle für AWS DMS](#) oder [Voraussetzungen für die Verwendung von Amazon S3 als Ziel](#).

Um beispielsweise Daten von einem S3-Quellendpunkt zu lesen oder Daten an einen S3-Zielendpunkt zu übertragen, müssen Sie eine Servicerolle als Voraussetzung für den Zugriff auf S3 für jede dieser Endpunktoperationen erstellen.

- Rollen mit Berechtigungen, die für die Verwendung der AWS CLI und der AWS DMS-API erforderlich sind – Zwei IAM-Rollen, die Sie erstellen müssen, sind `dms-vpc-role` und `dms-cloudwatch-logs-role`. Wenn Sie Amazon Redshift als Zieldatenbank verwenden, müssen Sie auch die IAM-Rolle erstellen und `dms-access-for-endpoint` Ihrem AWS Konto hinzufügen. Weitere Informationen finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Auswählen einer IAM-Rolle in AWS DMS

Wenn Sie die AWS CLI oder die AWS DMS-API für Ihre Datenbankmigration verwenden, müssen Sie Ihrem AWS Konto bestimmte IAM-Rollen hinzufügen, bevor Sie die Funktionen von AWS DMS verwenden können. Zwei dieser Optionen sind `dms-vpc-role` und `dms-cloudwatch-logs-role`. Wenn Sie Amazon Redshift als Zieldatenbank verwenden, müssen Sie Ihrem AWS Konto auch die IAM `dms-access-for-endpoint`-Rolle hinzufügen. Weitere Informationen finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Identitäts- und Zugriffsverwaltung für DMS Fleet Advisor

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. DMS Fleet Advisor unterstützt spezifische Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

DMS Fleet Advisor verwendet IAM-Rollen, um auf Amazon Simple Storage Service zuzugreifen. Eine [IAM-Rolle](#) ist eine Entität in Ihrem AWS Konto mit bestimmten Berechtigungen. Weitere Informationen finden Sie unter [Erstellen von IAM-Ressourcen](#).

AWS Database Migration Service Beispiele für identitätsbasierte Richtlinien

IAM-Benutzer besitzen keine Berechtigungen zum Erstellen oder Ändern von AWS DMS - Ressourcen. Sie können auch keine Aufgaben über die AWS Management Console AWS CLI, die oder die AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen

Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von Richtlinien auf der JSON-Registerkarte](#) im IAM-Benutzerhandbuch.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der AWS DMS -Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Zugreifen auf einen Amazon-S3-Bucket](#)
- [Zugreifen auf AWS DMS -Ressourcen basierend auf Tags](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS DMS Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS -verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu erteilen, verwenden Sie die -AWS verwalteten Richtlinien, die Berechtigungen für viele häufige Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar AWS-Konto. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die für Ihre Anwendungsfälle spezifisch sind. Weitere Informationen finden Sie unter [AWS -verwaltete Richtlinien](#) oder [AWS -verwaltete Richtlinien für Auftrags-Funktionen](#) im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten: Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn sie über eine bestimmte verwendet werden AWS-Service, z. B. AWS CloudFormation. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich – Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der AWS DMS -Konsole

Die folgende Richtlinie gewährt Ihnen Zugriff auf AWS DMS, einschließlich der AWS DMS-Konsole, und legt auch Berechtigungen für bestimmte Aktionen fest, die von anderen Amazon-Services wie Amazon EC2 benötigt werden.

```
{
  "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"
}
]
}

```

Eine Aufschlüsselung dieser Berechtigungen kann Ihnen helfen, besser zu verstehen, warum alle für die Verwendung der Konsole erforderlichen Berechtigungen erforderlich sind.

Der folgende Abschnitt ist erforderlich, damit die Benutzer berechtigt sind, ihre verfügbaren AWS KMS -Schlüssel und den Alias für die Anzeige in der Konsole aufzulisten. Dieser Eintrag ist nicht erforderlich, wenn Sie den Amazon-Ressourcennamen (ARN) für den KMS-Schlüssel kennen und nur die AWS Command Line Interface (AWS CLI) verwenden.

```

{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

Der folgende Abschnitt ist für bestimmte Endpunkttypen erforderlich, die die Übergabe eines Rollen-ARN mit dem Endpunkt erfordern. Wenn die erforderlichen AWS DMS Rollen nicht im Voraus erstellt werden, kann die AWS DMS Konsole die Rolle erstellen. Wenn alle Rollen vorab konfiguriert werden, muss dies innerhalb von `iam:GetRole` und `iam:PassRole` geschehen. Weitere Informationen zu Rollen finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

```

{
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:PassRole",
        "iam:CreateRole",

```

```

        "iam:AttachRolePolicy"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

Der folgende Abschnitt ist erforderlich, da die Amazon EC2-Instance erstellen und das Netzwerk für die erstellte Replikations-Instance konfigurieren AWS DMS muss. Diese Ressourcen sind im Konto des Kunden vorhanden, deshalb muss es möglich sein, diese Aktionen im Namen des Kunden auszuführen.

```

{
    "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"
}

```

Der folgende Abschnitt ist erforderlich, damit der Benutzer Replikations-Instance-Metriken anzeigen kann.

```

{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:Get*",
        "cloudwatch:List*"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

Dieser Abschnitt ist erforderlich, damit der Benutzer Replikationsprotokolle anzeigen kann.

```

{
    "Effect": "Allow",
    "Action": [

```

```

        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:FilterLogEvents",
        "logs:GetLogEvents"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

Die AWS DMS-Konsole erstellt mehrere Rollen, die Ihrem AWS Konto automatisch zugeordnet werden, wenn Sie die AWS DMS-Konsole verwenden. Wenn Sie die AWS Command Line Interface (AWS CLI) oder die AWS DMS-API für Ihre Migration verwenden, müssen Sie diese Rollen zu Ihrem Konto hinzufügen. Weitere Informationen zum Hinzufügen dieser Rollen finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Weitere Informationen zu den Anforderungen für die Verwendung dieser Richtlinie für den Zugriff auf AWS DMS finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI oder AWS API.

```

{
  "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": "*"
  }
]
```

Zugreifen auf einen Amazon-S3-Bucket

AWS DMS verwendet Amazon S3-Buckets als Zwischenspeicher für die Datenbankmigration. In der Regel verwaltet DMS zu diesem Zweck Standard- AWS S3-Buckets. S3 In bestimmten Fällen, insbesondere wenn Sie die AWS CLI oder die AWS DMS-API verwenden, können Sie mit AWS DMS jedoch stattdessen Ihren eigenen S3-Bucket angeben. Sie können beispielsweise einen eigenen S3-Bucket für die Migration von Daten zu einem Amazon-Redshift-Zielendpunkt angeben. In diesem Fall müssen Sie eine Rolle mit Berechtigungen erstellen, die auf der von AWS verwalteten `AmazonDMSRedshiftS3Role` Richtlinie basieren.

Das folgende Beispiel zeigt eine Version der Richtlinie `AmazonDMSRedshiftS3Role`. Es ermöglicht AWS DMS, einem IAM-Benutzer in Ihrem AWS Konto Zugriff auf einen Ihrer Amazon S3-Buckets zu gewähren. Außerdem kann der Benutzer Objekte hinzufügen, aktualisieren und löschen.

Zusätzlich zum Erteilen der Berechtigungen `s3:PutObject`, `s3:GetObject` und `s3:DeleteObject` für den Benutzer, gewährt die Richtlinie die Berechtigungen `s3:ListAllMyBuckets`, `s3:GetBucketLocation` und `s3:ListBucket`. Dies sind die zusätzlichen Berechtigungen, die von der Konsole benötigt werden. Andere Berechtigungen ermöglichen es AWS DMS, den Bucket-Lebenszyklus zu verwalten. Außerdem ist die `s3:GetObjectAcl`-Aktion erforderlich, um Objekte kopieren zu können.

```
{
  "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-*"
  }
]
}

```

Weitere Informationen zum Erstellen einer Rolle auf der Grundlage dieser Richtlinie finden Sie unter [Einstellungen von Amazon-S3-Buckets](#).

Zugreifen auf AWS DMS -Ressourcen basierend auf Tags

Sie können in Ihrer identitätsbasierten Richtlinie Bedingungen für die Steuerung des Zugriffs auf AWS DMS -Ressourcen auf der Basis von Tags verwenden. Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen können, die den Zugriff auf alle AWS DMS-Endpunkte erlaubt. Die Berechtigung wird jedoch nur gewährt, wenn der Wert des Endpunktdatenbank-Tags `Owner` der Name des Benutzers ist.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "dms:*",
      "Resource": "arn:aws:dms:*:*:endpoint/*",
      "Condition": {
        "StringEquals": {"dms:endpoint-tag/Owner": "${aws:username}"}
      }
    }
  ]
}

```

```
    }  
  ]  
}
```

Sie können diese Richtlinie den IAM-Benutzern in Ihrem Konto anfügen. Wenn ein Benutzer mit dem Namen `richard-roe` versucht, auf einen AWS DMS Endpunkt zuzugreifen, muss die Endpunktdatenbank mit dem Tag `Owner=richard-roe` oder `versehen seinowner=richard-roe`. Andernfalls wird diesem Benutzer der Zugriff verweigert. Der Tag-Schlüssel `Owner` der Bedingung stimmt sowohl mit `Owner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

Beispiele für ressourcenbasierte Richtlinien für AWS KMS

AWS Mit DMS können Sie benutzerdefinierte AWS KMS Verschlüsselungsschlüssel erstellen, um unterstützte Zielendpunktdaten zu verschlüsseln. Informationen zum Erstellen und Anfügen einer Schlüsselrichtlinie an den Verschlüsselungsschlüssel, den Sie für die unterstützte Verschlüsselung von Zieldaten erstellen, finden Sie unter [Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten](#) und [Erstellen von AWS KMS-Schlüsseln zum Verschlüsseln von Amazon-S3-Zielobjekten](#).

Themen

- [Eine Richtlinie für einen benutzerdefinierten AWS KMS Verschlüsselungsschlüssel zum Verschlüsseln von Amazon-Redshift-Zieldaten](#)
- [Eine Richtlinie für einen benutzerdefinierten AWS KMS Verschlüsselungsschlüssel zum Verschlüsseln von Amazon S3-Zieldaten](#)

Eine Richtlinie für einen benutzerdefinierten AWS KMS Verschlüsselungsschlüssel zum Verschlüsseln von Amazon-Redshift-Zieldaten

Das folgende Beispiel zeigt den JSON für die Schlüsselrichtlinie, die für einen AWS KMS - Verschlüsselungsschlüssel erstellt wurde, den Sie zum Verschlüsseln von Amazon-Redshift-Zieldaten erstellen.

```
{  
  "Id": "key-consolepolicy-3",  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": "kms:Encrypt",  
      "Resource": "*",  
      "Effect": "Allow",  
      "Principal": "AWS:*",  
      "Condition": {  
        "StringEquals": {  
          "kms:Owner": "richard-roe"  
        }  
      }  
    }  
  ]  
}
```

```
{
  "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
      }
    }
  }
]
}

```

Hier können Sie sehen, wo die Schlüsselrichtlinie auf die Rolle für den Zugriff auf Amazon-Redshift-Zielendpunktdaten verweist, die Sie vor dem Erstellen des Schlüssels erstellt haben. Im Beispiel ist dies `DMS-Redshift-endpoint-access-role`. Sie können auch die verschiedenen Schlüsselaktionen sehen, die für die verschiedenen Prinzipale (Benutzer und Rollen) zulässig sind. Beispielsweise kann jeder Benutzer mit `DMS-Redshift-endpoint-access-role` die Zieldaten verschlüsseln, entschlüsseln und neu verschlüsseln. Ein solcher Benutzer kann auch Datenschlüssel für den Export generieren, um die Daten außerhalb von zu verschlüsseln AWS KMS. Sie können auch detaillierte Informationen zu einem AWS KMS Schlüssel zurückgeben, z. B. den Schlüssel, den

Sie gerade erstellt haben. Darüber hinaus kann ein solcher Benutzer Anhänge an AWS -Ressourcen verwalten, z. B. den Zielendpunkt.

Eine Richtlinie für einen benutzerdefinierten AWS KMS Verschlüsselungsschlüssel zum Verschlüsseln von Amazon S3-Zieldaten

Das folgende Beispiel zeigt den JSON für die Schlüsselrichtlinie, die für einen AWS KMS - Verschlüsselungsschlüssel erstellt wurde, den Sie zum Verschlüsseln von Amazon-S3-Zieldaten erstellen.

```
{
  "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
    }
  }
}

```

```
}  
]
```

Hier können Sie sehen, wo die Schlüsselrichtlinie auf die Rolle für den Zugriff auf Amazon-S3-Zielendpunktdaten verweist, die Sie vor dem Erstellen des Schlüssels erstellt haben. Im Beispiel ist dies `DMS-S3-endpoint-access-role`. Sie können auch die verschiedenen Schlüsselaktionen sehen, die für die verschiedenen Prinzipale (Benutzer und Rollen) zulässig sind. Beispielsweise kann jeder Benutzer mit `DMS-S3-endpoint-access-role` die Zieldaten verschlüsseln, entschlüsseln und neu verschlüsseln. Ein solcher Benutzer kann auch Datenschlüssel für den Export generieren, um die Daten außerhalb von zu verschlüsseln AWS KMS. Sie können auch detaillierte Informationen zu einem AWS KMS Schlüssel zurückgeben, z. B. den Schlüssel, den Sie gerade erstellt haben. Darüber hinaus kann ein solcher Benutzer Anhänge an AWS -Ressourcen verwalten, z. B. den Zielendpunkt.

Verwenden von Secrets für den Zugriff auf AWS Database Migration Service -Endpunkte

Für ist ein Secret ein verschlüsselter Schlüssel AWS DMS, mit dem Sie eine Reihe von Benutzeranmeldeinformationen zur Authentifizierung über die AWS DMS geheime Authentifizierung darstellen können. Für einen Oracle-Endpunkt, der auch Oracle Automatic Storage Management (ASM) verwendet, AWS DMS benötigt ein zusätzliches Secret, das die Benutzeranmeldeinformationen für den Zugriff auf Oracle ASM darstellt.

Sie können das Secret oder die Secrets erstellen, die für die Secret-Authentifizierung AWS DMS benötigt, indem Sie AWS Secrets Manager, einen Service zum sicheren Erstellen, Speichern und Abrufen von Anmeldeinformationen für den Zugriff auf Anwendungen, Services und IT-Ressourcen in der Cloud und On-Premise, verwenden. Dies umfasst die Unterstützung der automatischen regelmäßigen Rotation des verschlüsselten Secret-Werts ohne Ihr Eingreifen, wodurch zusätzliche Sicherheit für Ihre Anmeldeinformationen gewährleistet wird. Durch die Aktivierung der Rotation von Secret-Werten in wird AWS Secrets Manager auch sichergestellt, dass diese Rotation von Secret-Werten ohne Auswirkungen auf eine Datenbankmigration erfolgt, die auf dem Secret basiert. Um eine Endpunkt-Datenbankverbindung verborgen zu authentifizieren, erstellen Sie ein Secret, dessen Identität oder ARN Sie `SecretsManagerSecretId` zuweisen und das Sie in Ihre Endpunkteinstellungen aufnehmen. Um Oracle ASM als Teil eines Oracle-Endpunkts verborgen zu authentifizieren, erstellen Sie ein Secret, dessen Identität oder ARN Sie `SecretsManagerOracleAsmSecretId` zuweisen und das Sie in Ihre Endpunkteinstellungen aufnehmen.

Note

Sie können keine von Amazon RDS Aurora verwalteten Master-Anmeldeinformationen verwenden. Diese Anmeldeinformationen enthalten keine Host- oder Portinformationen, die Verbindungen herstellen AWS DMS müssen. Erstellen Sie stattdessen einen neuen Benutzer und ein neues Secret. Informationen zum Erstellen eines Benutzers und eines Secrets finden Sie im Folgenden unter [Verwenden der AWS Management Console zum Erstellen eines Secrets und einer geheimen Zugriffsrolle](#).

Weitere Informationen zu AWS Secrets Manager finden Sie unter [Was ist AWS Secrets Manager?](#) im AWS Secrets Manager -Benutzerhandbuch.

AWS DMS unterstützt die Secret-Authentifizierung für die folgenden On-Premises- oder AWS-verwalteten Datenbanken auf unterstützten Quell- und Zielpunkten:

- Amazon DocumentDB
- IBM Db2 (LUW)
- Microsoft SQL Server
- MongoDB
- MySQL
- Oracle
- PostgreSQL
- Amazon Redshift
- SAP ASE

Um eine Verbindung zu einer dieser Datenbanken herzustellen, können Sie einen der folgenden Werte, aber nicht beide, als Teil Ihrer Endpunkteinstellungen eingeben:

- Klartext-Werte zur Authentifizierung der Datenbankverbindung mithilfe der Einstellungen `UserName`, `Password`, `ServerName` und `Port`. Fügen Sie für einen Oracle-Endpunkt, der auch Oracle ASM verwendet, zusätzliche Klartext-Werte hinzu, um ASM mithilfe der Einstellungen `AsmUserName`, `AsmPassword` und `AsmServerName` zu authentifizieren.
- Secret-Authentifizierung mit Werten für die Einstellungen `SecretsManagerSecretId` und `SecretsManagerAccessRoleArn`. Fügen Sie für einen Oracle-Endpunkt, der Oracle ASM verwendet, zusätzliche Werte für die Einstellungen `SecretsManagerOracleAsmSecretId` und

SecretsManagerOracleAsmAccessRoleArn hinzu. Die Secret-Werte für diese Einstellungen können Folgendes beinhalten für:

- SecretsManagerSecretId – den vollständigen Amazon-Ressourcennamen (ARN), einen Teil des ARN oder den Anzeigenamen eines Secrets, das Sie für den Endpunktdatenbank-Zugriff in AWS Secrets Manager erstellt haben.
- SecretsManagerAccessRoleArn – Der ARN einer geheimen Zugriffsrolle, die Sie in IAM erstellt haben, um in Ihrem Namen AWS DMS Zugriff auf dieses SecretsManagerSecretId Secret zu gewähren.
- SecretsManagerOracleAsmSecretId – den vollständigen Amazon-Ressourcennamen (ARN), einen Teil des ARN oder den Anzeigenamen eines Secrets, das Sie für den Zugriff auf Oracle SAM in AWS Secrets Manager erstellt haben.
- SecretsManagerOracleAsmAccessRoleArn – den ARN einer geheimen Zugriffsrolle, die Sie in IAM erstellt haben, um AWS DMS in Ihrem Namen Zugriff auf dieses SecretsManagerOracleAsmSecretId-Secret zu gewähren.

Note

Sie können auch eine einzelne geheime Zugriffsrolle verwenden, um Zugriff AWS DMS sowohl auf das SecretsManagerSecretId Secret als auch auf das SecretsManagerOracleAsmSecretId Secret zu gewähren. Wenn Sie diese einzelne geheime Zugriffsrolle für beide Secrets erstellen, müssen Sie SecretsManagerAccessRoleArn und SecretsManagerOracleAsmAccessRoleArn denselben ARN für diese Zugriffsrolle zuweisen. Wenn beispielsweise der ARN Ihrer geheimen Zugriffsrolle für beide Secrets der Variablen ARN2xsecrets zugewiesen ist, können Sie diese ARN-Einstellungen wie folgt festlegen:

```
SecretsManagerAccessRoleArn = ARN2xsecrets;  
SecretsManagerOracleAsmAccessRoleArn = ARN2xsecrets;
```

Weitere Informationen zum Erstellen dieser Werte finden Sie unter [Verwenden der AWS Management Console zum Erstellen eines Secrets und einer geheimen Zugriffsrolle](#).

Nachdem Sie das erforderliche Secret und die Endpunkteinstellungen für die geheime Zugriffsrolle für Ihre Endpunkte erstellt und angegeben haben, aktualisieren Sie die Berechtigungen

für die Benutzerkonten, die die API-Anfrage `CreateEndpoint` oder `ModifyEndpoint` ausführen werden, mit diesen Secret-Informationen. Stellen Sie sicher, dass diese Kontoberechtigungen die `-IAM:GetRole`-Berechtigung für die geheime Zugriffsrolle und die `-SecretsManager:DescribeSecret`-Berechtigung für das Secret enthalten. AWS DMS erfordert diese Berechtigungen, um sowohl die Zugriffsrolle als auch ihr Secret zu validieren.

So können Sie die erforderlichen Benutzerberechtigungen bereitstellen und überprüfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS Identity and Access Management Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie Benutzer und dann die Benutzer-ID aus, die für die API-Aufrufe `CreateEndpoint` und `ModifyEndpoint` verwendet wird.
3. Wählen Sie auf der Registerkarte Berechtigungen die Option `{}` JSON aus.
4. Vergewissern Sie sich, dass der Benutzer über die im Folgenden angezeigten Berechtigungen verfügt:

```
{
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "iam:GetRole",
      "iam:PassRole"
    ],
    "Resource": "SECRET_ACCESS_ROLE_ARN"
  },
  {
    "Effect": "Allow",
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "SECRET_ARN"
  }
]
```

5. Wenn der Benutzer nicht über diese Berechtigungen verfügt, fügen Sie sie hinzu.
6. Wenn Sie eine IAM-Rolle für DMS-API-Aufrufe verwenden, wiederholen Sie die obigen Schritte für die entsprechende Rolle.

7. Öffnen Sie ein Terminal und überprüfen Sie mit der AWS CLI, ob die Berechtigungen korrekt erteilt wurden, indem Sie die oben verwendete Rolle oder den oben verwendeten Benutzer übernehmen.
 - a. Überprüfen Sie die Benutzerberechtigung `SecretAccessRole` für mit dem `IAM-get-role` Befehl.

```
aws iam get-role --role-name ROLE_NAME
```

Ersetzen Sie *ROLE_NAME* durch den Namen von `SecretsManagerAccessRole`.

Wenn der Befehl eine Fehlermeldung zurückgibt, stellen Sie sicher, dass die Berechtigungen korrekt erteilt wurden.

- b. Überprüfen Sie die Benutzerberechtigung für das Secret mithilfe des `Secrets-Manager`-Befehls `describe-secret`.

```
aws secretsmanager describe-secret --secret-id SECRET_NAME OR SECRET_ARN --  
region=REGION_NAME
```

Der Benutzer kann der Anzeigename, ein Teil des ARN oder der vollständige ARN sein. Weitere Informationen finden Sie unter [describe-secret](#).


Wenn der Befehl eine Fehlermeldung zurückgibt, stellen Sie sicher, dass die Berechtigungen korrekt erteilt wurden.

Verwenden der AWS Management Console zum Erstellen eines Secrets und einer geheimen Zugriffsrolle

Sie können die verwendete AWS Management Console, um ein Secret für die Endpunktauthentifizierung zu erstellen und die Richtlinie und Rolle zu erstellen, damit AWS DMS in Ihrem Namen auf das Secret zugreifen kann.

So erstellen Sie ein Secret mit der AWS Management Console, die verwenden AWS DMS kann, um eine Datenbank für Quell- und Zielendpunktverbindungen zu authentifizieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS Secrets Manager Konsole unter <https://console.aws.amazon.com/secretsmanager/>.
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Wählen Sie unter Secret-Typ auswählen auf der Seite Ein neues Secret speichern die Option Anderer Secret-Typ und anschließend Klartext aus.


 Note

Dies ist die einzige Stelle, an der Sie ab diesem Zeitpunkt Klartext-Anmeldeinformationen eingeben müssen, um eine Verbindung zu Ihrer Endpunktdatenbank herzustellen.

4. Im Feld Klartext:

- Geben Sie für ein Secret, dessen Identität Sie SecretsManagerSecretId zuweisen, die folgende JSON-Struktur ein.

```
{
  "username": db_username,
  "password": db_user_password,
  "port": db_port_number,
  "host": db_server_name
}
```

 Note

Diese Liste enthält die für die Authentifizierung der Endpunktdatenbank mindestens erforderlichen JSON-Elemente. Sie können zusätzliche JSON-Endpunkteinstellungen als JSON-Elemente in Kleinbuchstaben hinzufügen, wenn Sie möchten. AWS DMS ignoriert jedoch alle zusätzlichen JSON-Elemente für die Endpunktauthentifizierung.

Hier ist *db_username* der Name des Benutzers, der auf die Datenbank zugreift, *db_user_password* ist das Passwort des Datenbankbenutzers, *db_port_number* die Portnummer für den Zugriff auf die Datenbank und *db_server_name* der Name (die Adresse) des Datenbankservers im Internet, wie im folgenden Beispiel gezeigt.

```
{
  "username": "admin",
  "password": "some_password",
  "port": "8190",
  "host": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com"
}
```

- Geben Sie für ein Secret, dessen Identität Sie `SecretsManagerOracleAsmSecretId` zuweisen, die folgende JSON-Struktur ein.

```
{
  "asm_user": asm_username,
  "asm_password": asm_user_password,
  "asm_server": asm_server_name
}
```

Note

Diese Liste enthält die für die Authentifizierung von Oracle ASM für einen Oracle-Endpunkt mindestens erforderlichen JSON-Elemente. Darüber hinaus ist dies die vollständige Liste, die Sie auf der Grundlage der verfügbaren Oracle-ASM-Endpunkteinstellungen angeben können.

Hier ist *asm_username* der Name des Benutzers, der auf Oracle ASM zugreift, *asm_user_password* das Passwort des Oracle-ASM-Benutzers und *asm_server_name* der Name (die Adresse) des Oracle-ASM-Servers im Internet, einschließlich des Ports, wie im folgenden Beispiel gezeigt.

```
{
  "asm_user": "oracle_asm_user",
  "asm_password": "oracle_asm_password",
  "asm_server": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com:8190/+ASM"
}
```

5. Wählen Sie einen AWS KMS Verschlüsselungsschlüssel aus, um das Secret zu verschlüsseln. Sie können den von für Ihren Service erstellten AWS Secrets Manager Standardverschlüsselungsschlüssel akzeptieren oder einen von Ihnen erstellten AWS KMS Schlüssel auswählen.

6. Geben Sie einen Namen für den Verweis auf dieses Secret und eine optionale Beschreibung an. Dies ist der Anzeigenname, den Sie als Wert für `SecretsManagerSecretId` oder `SecretsManagerOracleAsmSecretId` verwenden.
7. Wenn Sie die automatische Drehung im Secret aktivieren möchten, müssen Sie eine - AWS Lambda Funktion mit der Berechtigung zum Rotieren der Anmeldeinformationen für das Secret auswählen oder erstellen, wie beschrieben. Bevor Sie jedoch die automatische Rotation für die Verwendung Ihrer Lambda-Funktion einrichten, müssen Sie sicherstellen, dass die Konfigurationseinstellungen für die Funktion dem Wert der Umgebungsvariablen `EXCLUDE_CHARACTERS` die folgenden vier Zeichen hinzufügen.

```
;.:+{}
```

AWS DMS erlaubt diese Zeichen nicht in Passwörtern, die für Endpunktanmeldeinformationen verwendet werden. Wenn Sie Ihre Lambda-Funktion so konfigurieren, dass sie ausgeschlossen werden, generiert AWS Secrets Manager diese Zeichen nicht als Teil der rotierten Passwortwerte. Nachdem Sie die automatische Drehung für die Verwendung Ihrer Lambda-Funktion festgelegt haben, rotiert das Secret AWS Secrets Manager sofort, um Ihre Secret-Konfiguration zu validieren.

Note

Je nach der Konfiguration Ihrer Datenbank-Engine ruft Ihre Datenbank die rotierten Anmeldeinformationen möglicherweise nicht ab. In diesem Fall müssen Sie die Aufgabe manuell neu starten, um die Anmeldeinformationen zu aktualisieren.

8. Überprüfen und speichern Sie Ihr Secret in AWS Secrets Manager. Sie können dann jedes Secret nach seinem Anzeigenamen in suchen AWS Secrets Manager und dann den geheimen ARN als Wert für `SecretsManagerSecretId` oder `SecretsManagerOracleAsmSecretId` nach Bedarf abrufen, um den Zugriff auf Ihre Endpunktdatenbankverbindung und Oracle ASM (falls verwendet) zu authentifizieren.

So erstellen Sie die geheime Zugriffsrichtlinie und -rolle, um Ihre **SecretsManagerAccessRoleArn** oder festzulegen **SecretsManagerOracleAsmAccessRoleArn**, wodurch AWS Secrets Manager auf Ihr entsprechendes Secret zugreifen kann AWS DMS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Identity and Access Management (IAM)-Konsole unter <https://console.aws.amazon.com/iam/>.

2. Wählen Sie Richtlinien und anschließend Richtlinie erstellen aus.
3. Wählen Sie JSON aus und geben Sie die folgende Richtlinie ein, um den Zugriff auf Ihr Secret und dessen Entschlüsselung zu ermöglichen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": secret_arn,
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
      ],
      "Resource": kms_key_arn,
    }
  ]
}
```

Hier ist *secret_arn* der ARN Ihres Secrets, den Sie wie erforderlich von `SecretsManagerSecretId` oder `SecretsManagerOracleAsmSecretId` abrufen können, und *kms_key_arn* der ARN des AWS KMS -Schlüssels, mit dem Sie Ihr Secret verschlüsseln, wie im folgenden Beispiel gezeigt.

```
{
  "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

Wenn Sie den von erstellten Standardverschlüsselungsschlüssel verwenden AWS Secrets Manager, müssen Sie die AWS KMS Berechtigungen für nicht angeben *kms_key_arn*.

Wenn Ihre Richtlinie Zugriff auf beide Secrets gewähren soll, geben Sie einfach ein zusätzliches JSON-Ressourcenobjekt für den anderen *secret_arn* an.

Wenn sich Ihr Secret in einem anderen Konto befindet, benötigt die Rolle `SecretsManagerAccessRoleArn` eine zusätzliche Richtlinie, um das kontoübergreifende Secret zu überprüfen. Fügen Sie der Richtlinie in solchen Anwendungsfällen die Aktion `secretsmanager:DescribeSecret` hinzu. Weitere Informationen zum Einrichten eines kontoübergreifenden Secrets finden Sie unter [Berechtigungen für AWS Secrets-Manager-Secrets für Benutzer in einem anderen Konto](#).

- Überprüfen und erstellen Sie die Richtlinie mit einem Anzeigenamen und einer optionalen Beschreibung.
- Wählen Sie Rollen und anschließend Rolle erstellen aus.
- Wählen Sie als Typ der vertrauenswürdigen Entität AWS -Service aus.
- Wählen Sie in der Liste der Services DMS als vertrauenswürdigen Service aus und wählen Sie dann Weiter: Berechtigungen aus.
- Suchen Sie nach der Richtlinie, die Sie in Schritt 4 erstellt haben, und fügen Sie sie an. Fügen Sie dann alle Tags hinzu und überprüfen Sie Ihre Rolle. Bearbeiten Sie zu diesem Zeitpunkt die Vertrauensstellungen für die Rolle, um Ihren AWS DMS regionalen Service-Prinzipal als vertrauenswürdige Entität zu verwenden. Dieser Prinzipal weist das folgende Format auf:

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


Hier ist *region-name* der Name Ihrer Region, z. B. us-east-1. Daher folgt ein AWS DMS regionaler -Service-Prinzipal für diese Region.

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

9. Nachdem Sie die vertrauenswürdige Entität für die Rolle bearbeitet haben, erstellen Sie die Rolle mit einem Anzeigenamen und einer optionalen Beschreibung. Sie können Ihre neue Rolle jetzt anhand ihres Anzeigenamens in IAM suchen und dann den Rollen-ARN als Wert für `SecretsManagerAccessRoleArn` oder `SecretsManagerOracleAsmAccessRoleArn` abrufen, um Ihre Endpunkt-Datenbankverbindung zu authentifizieren.

So verwenden Sie Secrets Manager mit einer Replikations-Instance in einem privaten Subnetz

1. Erstellen Sie einen Secrets-Manager-VPC-Endpunkt und notieren Sie sich das DNS für den Endpunkt. Weitere Informationen zum Erstellen eines Secrets-Manager-VPC-Endpunkts finden Sie unter [Connecting to Secrets Manager through a VPC endpoint](#) im Benutzerhandbuch für AWS Secrets Manager.
2. Fügen Sie die Sicherheitsgruppe der Replikations-Instance an den Secrets-Manager-VPC-Endpunkt an.
3. Lassen Sie für die Ausgangsregeln für die Sicherheitsgruppe der Replikations-Instance den gesamten Datenverkehr für das Ziel `0.0.0.0/0` zu.
4. Legen Sie das zusätzliche Verbindungsattribut `secretsManagerEndpointOverride=secretsManager endpoint DNS` des Endpunkts fest, um das DNS des Secrets-Manager-VPC-Endpunkts bereitzustellen, wie im folgenden Beispiel gezeigt.

```
secretsManagerEndpointOverride=vpce-1234a5678b9012c-12345678.secretsmanager.eu-west-1.vpce.amazonaws.com
```

Verwenden von serviceverknüpften Rollen für AWS DMS

AWS Database Migration Service verwendet [serviceverknüpfte Rollen](#) von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit AWS DMS verknüpft ist. Serviceverknüpfte Rollen werden von AWS DMS vordefiniert und schließen

alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von AWS DMS, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS DMS definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur AWS DMS die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre AWS DMS-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rollen angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollen für AWS DMS-Features

Themen

- [Serviceverknüpfte Rollen für AWS DMS Fleet Advisor](#)
- [Serviceverknüpfte Rolle für AWS DMS Serverless](#)

Serviceverknüpfte Rollen für AWS DMS Fleet Advisor

AWS DMS Fleet Advisor verwendet die serviceverknüpfte Rolle `AWSServiceRoleForDMSFleetAdvisor` – DMS Fleet Advisor verwendet diese serviceverknüpfte Rolle zum Verwalten von Amazon-CloudWatch-Metriken. Diese verwaltete Richtlinie ist mit der folgenden serviceverknüpften Rolle verbunden: `AWSDMSFleetAdvisorServiceRolePolicy`. Aktualisierungen dieser Richtlinie finden Sie unter [AWS Von verwaltete Richtlinien für AWS Database Migration Service](#).

Die serviceverknüpfte Rolle `AWSServiceRoleForDMSFleetAdvisor` vertraut den folgenden Services, um die Rolle zu übernehmen:

- `dms-fleet-advisor.amazonaws.com`

Die Rollenberechtigungsrichtlinie `AWSDMSFleetAdvisorServiceRolePolicy` gestattet AWS DMS Fleet Advisor, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `cloudwatch:PutMetricData` für `all AWS resources`

Diese Berechtigung ermöglicht Prinzipalen die Veröffentlichung von Metrikdatenpunkten in Amazon CloudWatch. AWS DMS Fleet Advisor erfordert diese Berechtigung, um Diagramme mit Datenbankmetriken von CloudWatch anzuzeigen.

Das folgende Codebeispiel zeigt die Richtlinie `AWSDMSFleetAdvisorServiceRolePolicy`, mit der Sie die Rolle `AWSDMSFleetAdvisorServiceRolePolicy` erstellen.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Resource": "*",
    "Action": "cloudwatch:PutMetricData",
    "Condition": {
      "StringEquals": {
        "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
      }
    }
  }
}
```

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für AWS DMS Fleet Advisor

Sie können die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit dem Anwendungsfall DMS – Fleet Advisor zu erstellen. Erstellen Sie in der AWS CLI oder der AWS-API eine servicegebundene Rolle mit dem Servicenamen `dms-fleet-advisor.amazonaws.com`. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch. Wenn Sie diese servicegebundene Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Erstellen Sie diese Rolle unbedingt, bevor Sie einen Datensammler erstellen. DMS Fleet Advisor verwendet diese Rolle, um Diagramme mit Datenbankmetriken in der AWS Management Console anzuzeigen. Weitere Informationen finden Sie unter [Erstellen eines Datenkollektors](#).

Bearbeiten einer serviceverknüpften Rolle für AWS DMS Fleet Advisor

AWS DMS erlaubt das Bearbeiten der serviceverknüpften Rolle `AWSServiceRoleForDMSFleetAdvisor` nicht. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für AWS DMS Fleet Advisor

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Dadurch haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der AWS DMS-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie AWS DMS-Ressourcen, die von `AWSServiceRoleForDMSFleetAdvisor` verwendet werden

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Datensammler unter Entdecken aus. Die Seite Datensammler wird geöffnet.
3. Wählen Sie Ihren Datensammler aus und klicken Sie auf Löschen.
4. Geben Sie den Namen des Datensammlers in das Texteingabefeld ein, um den Löschvorgang zu bestätigen. Wählen Sie dann Löschen aus.

⚠ Important

Wenn Sie einen DMS-Datensammler löschen, löscht DMS Fleet Advisor alle Datenbanken aus dem Inventar, die Sie mit diesem Sammler entdeckt haben.

Nachdem Sie alle Datensammelpunkte gelöscht haben, können Sie die serviceverknüpfte Rolle löschen.

So löschen Sie die servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die AWS CLI oder die AWS-API, um die serviceverknüpfte Rolle `AWSServiceRoleForDMSFleetAdvisor` zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte Rollen für AWS DMS Fleet Advisor

AWS DMS Fleet Advisor unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [Unterstützte AWS-Regionen](#).

Serviceverknüpfte Rolle für AWS DMS Serverless

AWS DMS Serverless verwendet die benannte dienstverknüpfte Rolle.

`AWSServiceRoleForDMSServerless` AWS DMS verwendet diese servicebezogene Rolle, um in Ihrem Namen AWS DMS Ressourcen wie CloudWatch Amazon-Metriken zu erstellen und zu verwalten. AWS DMS verwendet diese Rolle, sodass Sie sich nur mit Replikationen befassen müssen. Diese verwaltete Richtlinie ist mit der folgenden serviceverknüpften Rolle verbunden: `AWSDMSServerlessServiceRolePolicy`. Aktualisierungen dieser Richtlinie finden Sie unter [AWS Von verwaltete Richtlinien für AWS Database Migration Service](#).

Die `AWSServiceRoleForDMSServerless` dienstverknüpfte Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `dms.amazonaws.com`

Das folgende Codebeispiel zeigt die `AWSDMSServerlessServiceRolePolicy` Richtlinie, mit der Sie die `AWSServiceRoleForDMSServerless` Rolle erstellen.

```
{
```

```

"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:*"
  ]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für AWS DMS Serverless

Wenn Sie eine Replikation erstellen, erstellt AWS DMS Serverless programmgesteuert eine AWS DMS serverlose, dienstverknüpfte Rolle. Sie können diese Rolle in der IAM-Konsole anzeigen. Sie können diese Rolle auch manuell erstellen. Um die Rolle manuell zu erstellen, verwenden Sie die IAM-Konsole, um eine dienstverknüpfte Rolle mit dem DMS-Anwendungsfall zu erstellen. Erstellen Sie in der AWS CLI oder der AWS API eine serviceverknüpfte Rolle, die Sie `dms.amazonaws.com` für den Dienstenamen verwenden. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Note

Wenn Sie eine Rolle löschen, obwohl in Ihrem Konto Replikationen vorhanden sind, führt die Replikation zu einem Fehler.

Bearbeiten einer serviceverknüpften Rolle für AWS DMS Serverless

AWS DMS erlaubt es Ihnen nicht, die `AWSServiceRoleForDMSServerless` dienstverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename

nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für AWS DMS Serverless

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Dadurch haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der AWS DMS Dienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um AWS DMS Ressourcen zu löschen, die verwendet werden von `AWSServiceRoleForDMSServerless`

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS DMS Konsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie im Navigationsbereich Serverless unter Entdecken aus. Die Seite Serverless wird geöffnet.
3. Wählen Sie Ihre Serverless-Replikation und dann Löschen aus.
4. Geben Sie den Namen der Serverless-Replikation in das Texteingabefeld ein, um das Löschen zu bestätigen. Wählen Sie dann Löschen aus.

Nachdem Sie alle Serverless-Replikationen gelöscht haben, können Sie die serviceverknüpfte Rolle löschen.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForDMSServerless` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte Rollen für AWS DMS Serverless

AWS DMS Serverless unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist.

Fehlerbehebung für AWS Database Migration Service Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Arbeiten mit AWS DMS und IAM auftreten können.

Themen

- [Ich bin nicht autorisiert, eine Aktion in auszuführen AWS DMS](#)
- [Ich bin nicht autorisiert, iam durchzuführen:PassRole](#)
- [Ich bin Administrator und möchte anderen Zugriff auf gewähren. AWS DMS](#)
- [Ich möchte Personen außerhalb meines - AWS Kontos Zugriff auf meine - AWS DMS Ressourcen gewähren](#)

Ich bin nicht autorisiert, eine Aktion in auszuführen AWS DMS

Wenn Ihnen AWS Management Console mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der mateojackson IAM-Benutzer versucht, die Konsole zu verwenden, um Details zu einem AWS -DMS-Endpunkt anzuzeigen, aber keine `dms:DescribeEndpoint` Berechtigungen besitzt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
dms:DescribeEndpoint on resource:my-postgresql-target
```

In diesem Fall bittet Mateo seinen Administrator, seine Richtlinien zu aktualisieren, um ihm den Zugriff auf die `my-postgresql-target`-Endpunktressource mit der `dms:DescribeEndpoint`-Aktion zu ermöglichen.

Ich bin nicht autorisiert, iam durchzuführen:PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS DMS übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine vorhandene Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder serviceverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS DMS auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin Administrator und möchte anderen Zugriff auf gewähren. AWS DMS

Um anderen Personen oder einer Anwendung Zugriff auf zu gewähren AWS DMS, müssen Sie eine IAM-Entität (Benutzer oder Rolle) für die Person oder Anwendung erstellen, die Zugriff benötigt. Sie werden die Anmeldeinformationen für diese Einrichtung verwenden, um auf AWS zuzugreifen. Anschließend müssen Sie der Entität eine Richtlinie anfügen, die dieser die korrekten Berechtigungen in AWS DMS gewährt.

Informationen zum Einstieg finden Sie unter [Erstellen Ihrer ersten delegierten IAM-Benutzer und -Gruppen](#) im IAM-Benutzerhandbuch.

Ich möchte Personen außerhalb meines - AWS Kontos Zugriff auf meine - AWS DMS Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffssteuerungslisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob diese Funktionen AWS DMS unterstützt, finden Sie unter [Funktionsweise AWS Database Migration Service von mit IAM](#).
- Informationen zum Gewähren des Zugriffs auf Ihre AWS-Konten -Ressourcen in Ihrem Besitz finden Sie unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto , das Sie besitzen](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre -Ressourcen gewähren AWS-Konten, finden Sie unter [Gewähren von Zugriff auf im AWS-Konten Besitz von Dritten](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS

Sie verwenden bestimmte IAM-Berechtigungen und IAM-Rollen zur Verwendung von AWS DMS. Wenn Sie als IAM-Benutzer angemeldet sind und verwenden möchten AWS DMS, muss Ihr Kontoadministrator die in diesem Abschnitt beschriebene Richtlinie an den IAM-Benutzer, die IAM-Gruppe oder -Rolle anfügen, die Sie zum Ausführen von verwenden AWS DMS. Weitere Informationen zu IAM-Berechtigungen finden Sie im [IAM-Benutzerhandbuch](#).

Die folgende Richtlinie gewährt Ihnen Zugriff auf sowie Berechtigungen für bestimmte Aktionen AWS DMS, die von anderen Amazon-Services wie AWS KMS, IAM, Amazon EC2 und Amazon benötigt werden CloudWatch. CloudWatch überwacht Ihre AWS DMS Migration in Echtzeit und sammelt und verfolgt Metriken, die den Fortschritt Ihrer Migration angeben. Sie können - CloudWatch Protokolle verwenden, um Probleme mit einer Aufgabe zu debuggen.

Note

Sie können den Zugriff auf AWS DMS Ressourcen mithilfe von Tagging weiter einschränken. Weitere Informationen zum Einschränken des Zugriffs auf AWS DMS Ressourcen mithilfe von Tagging finden Sie unter [Differenzierte Zugriffskontrolle mit Ressourcennamen und Ressourcen-Tags](#).

```
{
  "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"
}
]
}

```

Die Aufschlüsselung dieser Berechtigungen hilft Ihnen zu verstehen, warum die einzelnen Berechtigungen nötig sind.

Der folgende Abschnitt ist erforderlich, damit der Benutzer AWS DMS -API-Operationen aufrufen kann.

```
{
    "Effect": "Allow",
    "Action": "dms:*",
    "Resource": "arn:aws:dms:region:account:resourcetype/id"
}
```

Der folgende Abschnitt ist erforderlich, damit der Benutzer seine verfügbaren AWS KMS Schlüssel und Alias zur Anzeige in der Konsole auflisten kann. Dieser Eintrag ist nicht erforderlich, wenn Sie den Amazon-Ressourcennamen (ARN) für den KMS-Schlüssel kennen und nur die AWS Command Line Interface () verwendenAWS CLI.

```
{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

Der folgende Abschnitt ist für bestimmte Endpunkttypen erforderlich, die die Übergabe eines IAM-Rollen-ARN mit dem Endpunkt erfordern. Wenn die erforderlichen AWS DMS Rollen nicht im Voraus erstellt werden, kann die AWS DMS Konsole die Rolle erstellen. Wenn alle Rollen vorab konfiguriert werden, sind nur `iam:GetRole` und `iam:PassRole` erforderlich. Weitere Informationen zu Rollen finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

```
{
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:PassRole",
        "iam:CreateRole",

```

```

        "iam:AttachRolePolicy"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

Der folgende Abschnitt ist erforderlich, da die Amazon EC2-Instance erstellen und das Netzwerk für die erstellte Replikations-Instance konfigurieren AWS DMS muss. Diese Ressourcen sind im Konto des Kunden vorhanden, deshalb muss es möglich sein, diese Aktionen im Namen des Kunden auszuführen.

```

{
    "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"
}

```

Der folgende Abschnitt ist erforderlich, damit der Benutzer Replikations-Instance-Metriken anzeigen kann.

```

{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:Get*",
        "cloudwatch:List*"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

Dieser Abschnitt ist erforderlich, damit der Benutzer Replikationsprotokolle anzeigen kann.

```

{
    "Effect": "Allow",

```

```
"Action": [  
  "logs:DescribeLogGroups",  
  "logs:DescribeLogStreams",  
  "logs:FilterLogEvents",  
  "logs:GetLogEvents"  
],  
"Resource": "arn:aws:service:region:account:resourcetype/id"  
}
```

Die AWS DMS Konsole erstellt mehrere Rollen, die Ihrem AWS Konto automatisch zugeordnet werden, wenn Sie die AWS DMS Konsole verwenden. Wenn Sie die AWS Command Line Interface (AWS CLI) oder die AWS DMS -API für Ihre Migration verwenden, müssen Sie diese Rollen zu Ihrem Konto hinzufügen. Weitere Informationen zum Hinzufügen dieser Rollen finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS

Wenn Sie die AWS CLI oder die AWS DMS -API für Ihre Datenbankmigration verwenden, müssen Sie Ihrem AWS Konto drei IAM-Rollen hinzufügen, bevor Sie die Funktionen von verwenden können AWS DMS. Zwei dieser Optionen sind `dms-vpc-role` und `dms-cloudwatch-logs-role`. Wenn Sie Amazon Redshift als Zieldatenbank verwenden, müssen Sie Ihrem AWS Konto auch die `IAMdms-access-for-endpoint`-Rolle hinzufügen.

Aktualisierungen zu verwalteten Richtlinien erfolgen automatisch. Wenn Sie eine benutzerdefinierte Richtlinie mit den IAM-Rollen verwenden, stellen Sie sicher, dass Sie regelmäßig überprüfen, ob Aktualisierungen für die verwaltete Richtlinie in dieser Dokumentation vorliegen. Sie können die Details der verwalteten Richtlinie anzeigen, indem Sie eine Kombination der Befehle `get-policy` und `get-policy-version` verwenden.

Der folgende `get-policy`-Befehl ruft beispielsweise Informationen über die angegebene IAM-Rolle ab.

```
aws iam get-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole
```

Die von dem Befehl zurückgegebenen Informationen sind wie folgt.

```
{
  "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"
  }
}
```

```
}
```


Mit dem folgenden `get-policy-version`-Befehl werden IAM-Richtlinieninformationen abgerufen.

```
aws iam get-policy-version --policy-arn arn:aws:iam::aws:policy/service-role/  
AmazonDMSVPCManagementRole --version-id v3
```

Die von dem Befehl zurückgegebenen Informationen sind wie folgt.

```
{  
  "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  
  }  
}
```

Sie können dieselben Befehle verwenden, um Informationen über `AmazonDMSCloudWatchLogsRole` und die `AmazonDMSRedshiftS3Role`-verwaltete Richtlinie abrufen.

 Note

Wenn Sie die AWS DMS Konsole für Ihre Datenbankmigration verwenden, werden diese Rollen Ihrem AWS Konto automatisch hinzugefügt.

Mit den folgenden Verfahren werden die IAM-Rollen `dms-vpc-role`, `dms-cloudwatch-logs-role` und `dms-access-for-endpoint` erstellt.

So erstellen Sie die `dms-vpc-role` IAM-Rolle für die Verwendung mit der - AWS CLI oder AWS DMS - API

1. Erstellen Sie eine JSON-Datei mit der folgenden IAM-Richtlinie. Weisen Sie der JSON-Datei die Bezeichnung `dmsAssumeRolePolicyDocument.json` zu.

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

Erstellen Sie die Rolle mit dem AWS CLI folgenden Befehl.

```
aws iam create-role --role-name dms-vpc-role --assume-role-policy-document file://
dmsAssumeRolePolicyDocument.json
```

2. Fügen Sie die Richtlinie `AmazonDMSVPCManagementRole` mithilfe des folgenden Befehls an `dms-vpc-role` an.

```
aws iam attach-role-policy --role-name dms-vpc-role --policy-arn
arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole
```

So erstellen Sie die `dms-cloudwatch-logs-role` IAM-Rolle für die Verwendung mit der - AWS CLI oder AWS DMS -API

1. Erstellen Sie eine JSON-Datei mit der folgenden IAM-Richtlinie. Weisen Sie der JSON-Datei die Bezeichnung `dmsAssumeRolePolicyDocument2.json` zu.

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

Erstellen Sie die Rolle mit dem AWS CLI folgenden Befehl.

```
aws iam create-role --role-name dms-cloudwatch-logs-role --assume-role-policy-
document file://dmsAssumeRolePolicyDocument2.json
```

2. Fügen Sie die Richtlinie `AmazonDMSCloudWatchLogsRole` mithilfe des folgenden Befehls an `dms-cloudwatch-logs-role` an.

```
aws iam attach-role-policy --role-name dms-cloudwatch-logs-role --policy-arn
arn:aws:iam::aws:policy/service-role/AmazonDMSCloudWatchLogsRole
```

Wenn Sie Amazon Redshift als Zieldatenbank verwenden, müssen Sie die IAM-Rolle `dms-access-for-endpoint` erstellen, um den Zugriff auf Amazon S3 zu ermöglichen.

So erstellen Sie die `dms-access-for-endpoint` IAM-Rolle für die Verwendung mit Amazon Redshift als Zieldatenbank

1. Erstellen Sie eine JSON-Datei mit der folgenden IAM-Richtlinie. Weisen Sie der JSON-Datei die Bezeichnung `dmsAssumeRolePolicyDocument3.json` zu.

```
{
  "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. Erstellen Sie die Rolle mit dem AWS CLI folgenden Befehl.

```
aws iam create-role --role-name dms-access-for-endpoint --assume-role-policy-document file://dmsAssumeRolePolicyDocument3.json
```

3. Fügen Sie die Richtlinie AmazonDMSRedshiftS3Role mithilfe des folgenden Befehls an die Rolle `dms-access-for-endpoint` an.

```
aws iam attach-role-policy --role-name dms-access-for-endpoint \  
--policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSRedshiftS3Role
```

Sie sollten jetzt über die IAM-Richtlinien verfügen, um die - AWS CLI oder AWS DMS -API zu verwenden.

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS, dienstübergreifender Identitätswechsel kann zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel [aws:SourceArn](#) und die [aws:SourceAccount](#) globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die der AWS Database Migration Service Ressource einen anderen Dienst gewähren. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält, z. B. einen AWS DMS -Replikations-Instance-Namen (ARN), müssen Sie beide globalen Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

AWS DMS unterstützt Confused Deputy Options ab Version 3.4.7 und höher. Weitere Informationen finden Sie unter [AWS Versionshinweise zu Database Migration Service 3.4.7](#). Wenn Ihre Replikations-Instance AWS DMS Version 3.4.6 oder niedriger verwendet, stellen Sie sicher, dass Sie auf die neueste Version aktualisieren, bevor Sie die Confused-Deputy-Optionen festlegen.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Kontextbedingungsschlüssel `aws:SourceArn` mit Platzhalterzeichen (*) für die unbekanntenen Teile des ARN. z. B.

```
arn:aws:dms:*:123456789012:rep:*
```

Themen

- [IAM-Rollen, die zusammen mit der AWS DMS API verwendet werden können, um dienstübergreifend Verwirrung bei Stellvertretern zu vermeiden](#)
- [IAM-Richtlinie zur Speicherung von Preflight-Bewertungen in Amazon S3 zur serviceübergreifenden Vermeidung des Confused-Deputy-Problems](#)
- [Verwendung von Amazon DynamoDB als Zielendpunkt AWS DMS zur dienstübergreifenden Prävention verwirrter Stellvertreter](#)

IAM-Rollen, die zusammen mit der AWS DMS API verwendet werden können, um dienstübergreifend Verwirrung bei Stellvertretern zu vermeiden

Um die AWS CLI oder die AWS DMS API für Ihre Datenbankmigration zu verwenden, müssen Sie Ihrem AWS Konto die Rollen `dms-vpc-role` und die `dms-cloudwatch-logs-role` IAM-Rollen hinzufügen, bevor Sie die Funktionen von verwenden können. AWS DMS Weitere Informationen finden Sie unter [Erstellen der IAM-Rollen zur Verwendung mit der AWS CLI und API AWS DMS](#).

Das folgende Beispiel zeigt Richtlinien für die Verwendung der `dms-vpc-role`-Rolle mit der `my-replication-instance`-Replikations-Instance. Verwenden Sie diese Richtlinien, um das Confused-Deputy-Problem zu verhindern.

```
{
  "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"
        }
      }
    }
  ]
}
```



```

    }
  ]
}

```

IAM-Richtlinie zur Speicherung von Preflight-Bewertungen in Amazon S3 zur serviceübergreifenden Vermeidung des Confused-Deputy-Problems

Um die Ergebnisse der Vorabbeurteilung in Ihrem S3-Bucket zu speichern, erstellen Sie eine IAM-Richtlinie, die AWS DMS die Verwaltung von Objekten in Amazon S3 ermöglicht. Weitere Informationen finden Sie unter [Erstellen von IAM-Ressourcen](#).

Das folgende Beispiel zeigt eine Vertrauensrichtlinie mit verwirrten stellvertretenden Bedingungen, die für eine IAM-Rolle festgelegt sind und AWS DMS den Zugriff auf alle Aufgaben und Bewertungsläufe unter einem bestimmten Benutzerkonto ermöglichen.

```

{
  "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:*"
          ]
        }
      }
    }
  ]
}

```

Verwendung von Amazon DynamoDB als Zielpunkt AWS DMS zur dienstübergreifenden Prävention verwirrter Stellvertreter

Um Amazon DynamoDB als Zielpunkt für Ihre Datenbankmigration zu verwenden, müssen Sie die IAM-Rolle erstellen, die es ermöglicht, Zugriff auf die DynamoDB-Tabellen AWS DMS zu übernehmen und zu gewähren. Verwenden Sie dann diese Rolle, wenn Sie Ihren DynamoDB-Zielpunkt in AWS DMS erstellen. Weitere Informationen finden Sie unter [Verwenden von Amazon DynamoDB als Ziel](#).

Das folgende Beispiel zeigt eine Vertrauensrichtlinie mit verwirrten stellvertretenden Bedingungen, die für eine IAM-Rolle festgelegt sind, die allen AWS DMS Endpunkten den Zugriff auf DynamoDB-Tabellen ermöglicht.

```
{
  "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:*"
        }
      }
    }
  ]
}
```

AWS Von verwaltete Richtlinien für AWS Database Migration Service

Themen

- [Von AWS verwaltete Richtlinie: AWSDMSServerlessServiceRolePolicy](#)

- [Von AWS verwaltete Richtlinie: AmazonDMSCloudWatchLogsRole](#)
- [Von AWS verwaltete Richtlinie: AWSDMSFleetAdvisorServiceRolePolicy](#)
- [AWS DMS-Aktualisierungen für AWS verwaltete Richtlinien](#)

Von AWS verwaltete Richtlinie: AWSDMSServerlessServiceRolePolicy

Diese Richtlinie ist an die Rolle `AWSServiceRoleForDMSServerless` angefügt, die AWS DMS das Ausführen von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter [Serviceverknüpfte Rolle für AWS DMS Serverless](#).

Diese Richtlinie gewährt Beitragendenberechtigungen, mit denen AWS DMS Replikationsressourcen verwalten kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `dms` – ermöglicht Prinzipalen die Interaktion mit AWS DMS-Ressourcen.

```
{
  "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:*"
    ]
}
]
}

```

Von AWS verwaltete Richtlinie: AmazonDMSCloudWatchLogsRole

Diese Richtlinie ist an die Rolle `dms-cloudwatch-logs-role` angefügt, die AWS DMS das Ausführen von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für AWS DMS](#).

Diese Richtlinie gewährt Beitragendenberechtigungen, mit denen AWS DMS Replikationsprotokolle in CloudWatch Logs veröffentlichen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- **logs** – ermöglicht Prinzipalen die Veröffentlichung von Protokollen in CloudWatch Logs. Diese Berechtigung ist erforderlich, damit AWS DMS CloudWatch zum Anzeigen von Replikationsprotokollen verwenden kann.

```
{
  "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-*"
    ]
  }
]
}

```

Von AWS verwaltete Richtlinie: AWSDMSFleetAdvisorServiceRolePolicy

Sie können AWSDMSFleetAdvisorServiceRolePolicy nicht an Ihre IAM-Entitäten anfügen. Diese Richtlinie ist an eine serviceverknüpfte Rolle angefügt, die AWS DMS Fleet Advisor das Ausführen von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für AWS DMS](#).

Diese Richtlinie gewährt Beitragendenberechtigungen, mit denen AWS DMS Fleet Advisor Amazon-CloudWatch-Metriken veröffentlichen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `cloudwatch` – ermöglicht Prinzipalen die Veröffentlichung von Metrikdatenpunkten in Amazon CloudWatch. Diese Berechtigung ist erforderlich, damit AWS DMS Fleet Advisor CloudWatch verwenden kann, um Diagramme mit Datenbankmetriken anzuzeigen.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Resource": "*",
    "Action": "cloudwatch:PutMetricData",
    "Condition": {
      "StringEquals": {
        "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
      }
    }
  }
}
```

AWS DMS-Aktualisierungen für AWS verwaltete Richtlinien

Anzeigen von Details zu Aktualisierungen für AWS-verwaltete Richtlinien für AWS DMS, seit dieser Dienst mit der Verfolgung dieser Änderungen begonnen hat. Um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Seite [AWS DMS-Dokumentverlauf](#).

Änderung	Beschreibung	Datum
AWSDMSServerlessServiceRolePolicy – neue Richtlinie	AWS DMS hat die Rolle <code>AWSDMSServerlessServiceRolePolicy</code> hinzugefügt, damit AWS DMS Services in Ihrem Namen erstellen und verwalten kann, z. B. das Veröffentlichen von Amazon-CloudWatch-Metriken.	22. Mai 2023
AmazonDMSCloudWatchLogsRole – Änderung	AWS DMS hat den ARN für Serverless-Ressourcen zu jeder der gewährten Berechtigungen hinzugefügt, um das Hochladen von AWS DMS-Replikationsprotokollen aus Serverless-Replikationskonfigurationen in CloudWatch Logs zu ermöglichen.	22. Mai 2023
AWSDMSFleetAdvisorServiceRolePolicy – neue Richtlinie	AWS DMS Fleet Advisor hat eine neue Richtlinie hinzugefügt, um das Veröffentlichen von Metrikdatenpunkten in Amazon CloudWatch zu ermöglichen.	6. März 2023
AWS DMS hat die Änderungsverfolgung gestartet	AWS DMS hat mit der Verfolgung von Änderungen für seine AWS-verwalteten Richtlinien begonnen.	6. März 2023

Compliance-Validierung für AWS Database Migration Service

Die Auditoren Dritter bewerten die Sicherheit und die Compliance von AWS Database Migration Service im Rahmen mehrerer AWS-Compliance-Programme. Diese umfassen u. a. die folgenden Programme:

- SOC
- PCI
- ISO
- FedRAMP
- DoD CC SRG
- HIPAA BAA
- MTCS
- CS
- K-ISMS
- ENS High
- OSPAR
- HITRUST CSF

Eine Liste der AWS Services, die in den Anwendungsbereich bestimmter Compliance-Programme fallen, finden Sie unter [AWS Services im Anwendungsbereich nach Compliance-Programm](#) . Allgemeine Informationen finden Sie unter [AWS-Compliance-Programme](#).

Sie können Auditberichte von Drittanbietern unter AWS Artifact herunterladen. Weitere Informationen finden Sie unter [Herunterladen von Berichten in AWS Artifact](#) .

Ihre Compliance-Verantwortung bei der Verwendung von AWS DMS ist von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften abhängig. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- [Kurzanleitungen für Sicherheit und Compliance](#) – In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von sicherheits- und Compliance-orientierten Basisumgebungen in AWS.

- [Erstellen einer Architektur für HIPAA-Sicherheit und -Konformität in Amazon Web Services](#) – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-konforme Anwendungen erstellen können.
- [AWS Ressourcen zur Einhaltung von Vorschriften](#) zur Einhaltung von Vorschriften - Diese Sammlung von Arbeitsbüchern und Leitfäden könnte für Ihre Branche und Ihren Standort zutreffen.
- [AWS Config](#) – Dieser AWS-Service bewertet, zu welchem Grad die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- [AWS Security Hub](#) – Dieser AWS-Service liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS. So können Sie die Compliance mit den Sicherheitsstandards in der Branche und den bewährten Methoden abgleichen.

Ausfallsicherheit in AWS Database Migration Service

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und Availability Zones. AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die mit Netzwerken mit geringer Latenz, hohem Durchsatz und hochredundanten Vernetzungen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und -Availability Zones finden Sie unter [Globale AWS-Infrastruktur](#).

Neben der globalen AWS-Infrastruktur bietet AWS DMS Hochverfügbarkeit und Failover-Unterstützung für eine Replikations-Instance mit einer Multi-AZ-Bereitstellung, wenn Sie die Multi-AZ-Option wählen.

Bei einer Multi-AZ-Bereitstellung sorgt AWS DMS für die automatische Bereitstellung und Verwaltung eines Standby-Replikats der Replikations-Instance in einer anderen Availability Zone. Die primäre Replikations-Instance wird auf das Standby-Replikat repliziert. Wenn die primäre Replikations-Instance ausfällt oder nicht mehr reagiert, nimmt der Standby-Modus alle laufenden Tasks mit minimaler Unterbrechung wieder auf. Da die primäre Replikations-Instance ihren Status ständig in den Standby-Modus repliziert, verursacht die Multi-AZ-Bereitstellung einen gewissen Leistungs-Overhead.

Weitere Informationen zum Arbeiten mit Multi-AZ-Bereitstellungen finden Sie unter [Mit einer AWS DMS Replikationsinstanz arbeiten](#).

Sicherheit der Infrastruktur in AWS Database Migration Service

Als verwalteter Service ist AWS Database Migration Service durch die globalen Verfahren zur Gewährleistung der Netzwerksicherheit von AWS geschützt. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) im Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf AWS DMS zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Sie können diese API-Operationen von jedem Netzwerkstandort aus aufrufen. AWS DMS unterstützt darüber hinaus ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen für Aktionen und Ressourcen beispielsweise auf der Basis der Quell-IP-Adresse angeben können. Sie können AWS DMS-Richtlinien auch verwenden, um den Zugriff über spezifische Amazon-VPC-Endpunkte oder spezifische Virtual Private Clouds (VPCs) zu steuern. Tatsächlich wird der Netzwerkzugriff hierdurch auf eine bestimmte AWS DMS-Ressource eingeschränkt, sodass er ausschließlich über eine bestimmte VPC innerhalb des AWS-Netzwerks ausgeführt werden kann. Weitere Hinweise zum Verwenden von ressourcenbasierten Zugriffsrichtlinien mit AWS DMS, mit Beispielen, finden Sie unter [Differenzierte Zugriffskontrolle mit Ressourcennamen und Ressourcen-Tags](#).

Um Ihre Kommunikation mit AWS DMS auf eine einzige VPC zu beschränken, können Sie einen VPC-Schnittstellenendpunkt erstellen, der es Ihnen ermöglicht, über AWS PrivateLink eine Verbindung mit AWS DMS herzustellen. AWS PrivateLink stellt sicher, dass alle Aufrufe von AWS DMS sowie die zugehörigen Ergebnisse auf die spezifische VPC beschränkt bleiben, für die der Schnittstellenendpunkt erstellt wurde. Sie können dann die URL für diesen Schnittstellenendpunkt

optional bei jedem AWS DMS-Befehl angeben, den Sie mit der AWS CLI oder einem SDK ausführen. Auf diese Weise können Sie sicherstellen, dass Ihre gesamte Kommunikation mit AWS DMS auf die VPC beschränkt bleibt und ansonsten für das öffentliche Internet unsichtbar ist.

So erstellen Sie einen Schnittstellenendpunkt für den Zugriff auf DMS in einer einzelnen VPC

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im Navigationsbereich Endpunkte aus. Dadurch wird die Seite Endpunkte erstellen geöffnet, auf der Sie den Schnittstellenendpunkt von einer VPC zu AWS DMS erstellen können.
3. Wählen Sie AWS-Services aus, suchen Sie dann nach einem Wert für Service-Name, in diesem Fall AWS DMS, und wählen Sie ihn in der folgenden Form aus.

```
com.amazonaws.region.dms
```

Hier gibt *region* die AWS-Region an, in der AWS DMS ausgeführt wird, z. B. `com.amazonaws.us-west-2.dms`.

4. Wählen Sie für VPC die VPC aus, für die der Schnittstellenendpunkt erstellt werden soll, z. B. `vpc-12abcd34`.
5. Wählen Sie einen Wert für Availability Zone und für Subnetz-ID aus. Diese Werte sollten einen Standort angeben, an dem der gewählte AWS DMS-Endpunkt ausgeführt werden kann, zum Beispiel `us-west-2a` (`usw2-az1`) und `subnet-ab123cd4`.
6. Wählen Sie DNS-Namen aktivieren aus, um den Endpunkt mit einem DNS-Namen zu erstellen. Dieser DNS-Name besteht aus der Endpunkt-ID (`vpce-12abcd34efg567hij`) mit einem Bindestrich und einer zufälligen Zeichenfolge (`ab12dc34`). Diese werden in umgekehrter Reihenfolge durch Punkte vom Service-Namen getrennt und mit dem Zusatz `vpce` versehen (`dms.us-west-2.vpce.amazonaws.com`).

Ein Beispiel ist `vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com`.

7. Wählen Sie für Sicherheitsgruppe eine Gruppe aus, die für den Endpunkt verwendet werden soll.

Achten Sie bei der Einrichtung Ihrer Sicherheitsgruppe darauf, dass ausgehende HTTPS-Aufrufe innerhalb der Gruppe zugelassen sind. Weitere Informationen finden Sie unter [Creating security groups](#) im Benutzerhandbuch für Amazon VPC.

8. Wählen Sie entweder Vollzugriff oder einen benutzerdefinierten Wert für Richtlinie aus. Sie können beispielsweise eine benutzerdefinierte Richtlinie ähnlich der folgenden wählen, die den Zugriff Ihres Endpunkts auf bestimmte Aktionen und Ressourcen einschränkt.

```
{
  "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": "*"
    }
  ]
}
```

Hier erlaubt die Beispielrichtlinie jeden AWS DMS-API-Aufruf, mit Ausnahme des Löschens oder Änderns einer spezifischen Replikations-Instance.

Sie können jetzt optional eine URL angeben, die mit dem in Schritt 6 erstellten DNS-Namen gebildet wurde. Sie geben diese für jeden AWS DMS-CLI-Befehl oder jede API-Operation an, um über den erstellten Schnittstellenendpunkt auf die Service-Instance zuzugreifen. Beispielsweise können Sie den DMS-CLI-Befehl `DescribeEndpoints` wie folgt in dieser VPC ausführen.

```
$ aws dms describe-endpoints --endpoint-url https://vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com
```

Wenn Sie die Option für das private DNS aktivieren, müssen Sie die Endpunkt-URL in der Anfrage nicht angeben.

Weitere Informationen zum Erstellen und Verwenden von VPC-Schnittstellenendpunkten (einschließlich der Aktivierung der privaten DNS-Option) finden Sie unter [Interface VPC endpoints \(AWS PrivateLink\)](#) im Benutzerhandbuch für Amazon VPC.

Differenzierte Zugriffskontrolle mit Ressourcennamen und Ressourcen-Tags

Sie können Ressourcennamen und Ressourcen-Tags basierend auf Amazon-Ressourcennamen (ARNs) verwenden, um den Zugriff auf AWS DMS-Ressourcen zu verwalten. Dazu definieren Sie erlaubte Aktionen oder schließen Bedingungsanweisungen in IAM-Richtlinien ein.

Verwenden von Ressourcennamen für die Zugriffskontrolle

Sie können ein IAM-Benutzerkonto erstellen und eine Richtlinie basierend auf dem ARN der AWS DMS-Ressource zuweisen.

Mit der folgenden Richtlinie wird der Zugriff auf die AWS DMS-Replikations-Instance mit dem ARN `arn:aws:dms:us-east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV` verweigert:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV"
    }
  ]
}
```

Bei diesem Beispiel würden bei Geltung der Richtlinie die folgenden Befehle fehlschlagen:

```
$ 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:D0H67ZT0XGLIXMIHKITV
```

```
$ 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

Sie können auch IAM-Richtlinien angeben, die den Zugriff auf AWS DMS-Endpunkte und Replikationsaufgaben einschränken.

Mit der folgenden Richtlinie wird der Zugriff auf einen AWS DMS-Endpunkt mithilfe des ARN des Endpunkts eingeschränkt:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-
east-1:152683116:endpoint:D6E37YBXTNH0A6XRQSZCUGX"
    }
  ]
}
```

Hier würden beispielsweise die folgenden Befehle fehlschlagen, wenn die Richtlinie, die den ARN des Endpunkts verwendet, gelten würde:

```
$ 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
```

Mit der folgenden Richtlinie wird der Zugriff auf eine AWS DMS-Aufgabe mithilfe des ARN der Aufgabe eingeschränkt:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-east-1:152683116:task:U03YR4N47DXH3ATT4YMW0IT"
    }
  ]
}
```

Hier würden beispielsweise die folgenden Befehle fehlschlagen, wenn die Richtlinie, die den ARN der Aufgabe verwendet, gelten würde.

```
$ 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
```

Verwendung von Tags zur Zugriffssteuerung

AWS DMS definiert eine Reihe allgemeiner Schlüssel-Wert-Paare, die in benutzerdefinierten Richtlinien ohne zusätzliche Tagging-Anforderungen verfügbar sind. Weitere Informationen über das Markieren von AWS DMS-Ressourcen mit Tags finden Sie unter [Tagging von Ressourcen im AWS Database Migration Service](#).

Die folgende Liste enthält die Standard-Tags für die Verwendung mit AWS DMS:

- `aws:CurrentTime` – Stellt das Datum und die Uhrzeit der Anforderung dar, sodass eine Beschränkung des Zugriffs basierend auf zeitlichen Kriterien möglich ist.
- `aws:EpochTime` – Dieses Tag ähnelt dem obigen `aws:CurrentTime`-Tag, mit der Ausnahme, dass die aktuelle Zeit in Form der verstrichenen Sekunden seit der Unix-Epoche dargestellt wird.
- `aws:MultiFactorAuthPresent` – Hierbei handelt es sich um ein boolesches Tag, das angibt, ob die Anforderung mit Multi-Faktor-Authentifizierung signiert wurde.
- `aws:MultiFactorAuthAge` – Bietet Zugriff auf das Alter des Tokens der Multi-Factor Authentication (in Sekunden).
- `aws:principaltype` – Bietet Zugriff auf den Prinzipaltyp (Benutzer, Konto, Verbundbenutzer usw.) für die aktuelle Anforderung.
- `aws:SourceIp` – Stellt die Quell-IP-Adresse für den Benutzer dar, der die Anforderung ausgibt.
- `aws:UserAgent` – Bietet Informationen über die Client-Anwendung, die eine Ressource angefordert hat.
- `aws:userid` – Bietet Zugriff auf die ID des Benutzers, der die Anforderung ausgibt.
- `aws:username` – Bietet Zugriff auf den Namen des Benutzers, der die Anforderung ausgibt.
- `dms:InstanceClass` – Bietet Zugriff auf die Datenverarbeitungsgröße der/s Replikations-Instance-Hosts.
- `dms:StorageSize` – Bietet Zugriff auf die Speichergröße (in GB).

Sie können auch eigene Tags definieren. Benutzerdefinierte Tags sind einfache Schlüssel-Wert-Paare, die im AWS-Tagging-Service persistent sind. Sie können diese zu AWS DMS-Ressourcen hinzufügen, einschließlich Replikations-Instances, Endpunkte und Aufgaben. Diese Tags werden durch das Verwenden von "konditionalen" IAM-Anweisungen in Richtlinien abgestimmt. Der Verweis auf diese Tags erfolgt mittels eines spezifischen konditionalen Tags. Die Tag-Schlüssel weisen das Präfix "dms", den Ressourcentyp und das "tag"-Präfix auf. Im Folgenden sehen Sie das Tag-Format.

```
dms:{resource type}-tag/{tag key}={tag value}
```

Angenommen, Sie möchten eine Richtlinie definieren, durch die ein API-Aufruf nur dann erfolgreich durchgeführt werden kann, wenn eine Replikations-Instance das Tag "stage=production" enthält. Die folgende Bedingungsanweisung stimmt mit einer Ressource mit dem angegebenen Tag überein.

```
"Condition":
{
  "streq":
  {
    "dms:rep-tag/stage":"production"
  }
}
```

Sie fügen das folgende Tag zu einer Replikations-Instance hinzu, die mit dieser Richtlinienbedingung übereinstimmt.

```
stage production
```

Zusätzlich zu bereits zu den AWS DMS-Ressourcen hinzugefügten Tags können Richtlinien auch erstellt werden, um die Tag-Schlüssel und -Werte zu begrenzen, die auf eine bestimmte Ressource angewendet werden können. In diesem Fall ist das Tag-Präfix "req".

Mit der folgenden Richtlinienanweisung werden die Tags auf eine bestimmte Liste zulässiger Werte beschränkt, die ein Benutzer einer bestimmten Ressource zuweisen kann.

```
"Condition":
{
  "streq":
  {
    "dms:rep-tag/stage": [ "production", "development", "testing" ]
  }
}
```

```
}
```

Die folgenden Richtlinienbeispiele beschränken den Zugriff auf eine AWS DMS-Ressource basierend auf Ressourcen-Tags.

Die folgende Richtlinie beschränkt den Zugriff auf eine Replikations-Instance, bei der der Tag-Wert "Desktop" und der Tag-Schlüssel "Env" ist:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "dms:rep-tag/Env": [
            "Desktop"
          ]
        }
      }
    }
  ]
}
```

Die folgenden Befehle können erfolgreich ausgeführt werden oder schlagen fehl. Dies hängt von der IAM-Richtlinie ab, die den Zugriff einschränkt, wenn der Tag-Wert "Desktop" und der Tag-Schlüssel "Env" ist.

```
$ 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:46DHOU7J0JY0JXWDOZNFEN"
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
```

Die folgende Richtlinie beschränkt den Zugriff auf einen AWS DMS-Endpunkt, bei dem der Tag-Wert „Desktop“ und der Tag-Schlüssel „Env“ ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "dms:endpoint-tag/Env": [
            "Desktop"
          ]
        }
      }
    }
  ]
}
```

Die folgenden Befehle können erfolgreich ausgeführt werden oder schlagen fehl. Dies hängt von der IAM-Richtlinie ab, die den Zugriff einschränkt, wenn der Tag-Wert "Desktop" und der Tag-Schlüssel "Env" ist.

```
$ 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:J2YCZPNGOLF52344IZWA6I

```
$ aws dms modify-endpoint
  --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLF52344IZWA6I"
```

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

Die folgende Richtlinie beschränkt den Zugriff auf eine Replikationsaufgabe, bei der der Tag-Wert "Desktop" und der Tag-Schlüssel "Env" ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```



```

    "Action": [
      "dms:*"
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "dms:task-tag/Env": [
          "Desktop"
        ]
      }
    }
  ]
}

```

Die folgenden Befehle können erfolgreich ausgeführt werden oder schlagen fehl. Dies hängt von der IAM-Richtlinie ab, die den Zugriff einschränkt, wenn der Tag-Wert "Desktop" und der Tag-Schlüssel "Env" ist.

```

$ 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

Einrichten eines Verschlüsselungsschlüssels und Angeben von AWS KMS-Berechtigungen

AWS DMS verschlüsselt den von einer Replikations-Instance verwendeten Speicher und die Verbindungsinformationen für den Endpunkt. Um den von einer Replikations-Instance verwendeten Speicher zu verschlüsseln, verwendet AWS DMS einen AWS Key Management Service (AWS KMS)-Schlüssel, der nur für Ihr AWS-Konto gültig ist. Sie können diesen Schlüssel mit AWS KMS anzeigen und verwalten. Sie können den Standard-KMS-Schlüssel in Ihrem Konto (`aws/dms`) verwenden oder Sie können einen benutzerdefinierten KMS-Schlüssel erstellen. Wenn Sie bereits über einen KMS-Schlüssel verfügen, können Sie auch diesen für die Verschlüsselung verwenden.

Note

Jeder benutzerdefinierte oder vorhandene AWS KMS-Schlüssel, den Sie als Verschlüsselungsschlüssel verwenden, muss ein symmetrischer Schlüssel sein. AWS DMS unterstützt die Verwendung asymmetrischer Verschlüsselungsschlüssel nicht. Weitere Informationen zu symmetrischen und asymmetrischen Verschlüsselungsschlüsseln finden Sie unter <https://docs.aws.amazon.com/kms/latest/developerguide/symmetric-asymmetric.html> im AWS Key Management Service-Entwicklerhandbuch.

Der Standard-KMS-Schlüssel (`aws/dms`) wird erstellt, wenn Sie zum ersten Mal eine Replikations-Instance starten und keinen benutzerdefinierten KMS-Schlüssel aus dem Bereich Erweitert der Seite Replikations-Instance erstellen ausgewählt haben. Wenn Sie den Standard-KMS-Schlüssel verwenden, müssen Sie dem IAM-Benutzerkonto, das Sie für die Migration verwenden, nur die Berechtigungen `kms:ListAliases` und `kms:DescribeKey` erteilen. Weitere Informationen zum Verwenden des Standard-KMS-Schlüssels finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

Um einen benutzerdefinierten KMS-Schlüssel zu verwenden, weisen Sie Berechtigungen für den benutzerdefinierten KMS-Schlüssel mithilfe einer der folgenden Optionen zu:

- Fügen Sie das IAM-Benutzerkonto, das für die Migration verwendet wird, als Schlüsseladministrator oder Schlüsselbenutzer für den benutzerdefinierten AWS KMS-Schlüssel hinzu. Auf diese Weise wird sichergestellt, dass dem IAM-Benutzerkonto die erforderlichen AWS KMS-Berechtigungen erteilt werden. Diese Aktion besteht zusätzlich zu den IAM-Berechtigungen, die Sie dem IAM-Benutzerkonto für die Nutzung von AWS DMS erteilen müssen. Weitere

Informationen zum Erteilen von Berechtigungen für einen Schlüsselbenutzer finden Sie unter [Gestattet Schlüsselbenutzern die Verwendung des KMS-Schlüssels](#) im AWS Key Management Service-Entwicklerhandbuch.

- Wenn Sie das IAM-Benutzerkonto nicht als Schlüsseladministrator oder Schlüsselbenutzer für Ihren benutzerdefinierten KMS-Schlüssel hinzufügen möchten, dann fügen Sie folgende zusätzliche Berechtigungen zu den IAM-Berechtigungen hinzu, die Sie dem IAM-Benutzerkonto für die Nutzung von AWS DMS erteilen müssen.

```
{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:Encrypt",
        "kms:ReEncrypt*"
    ],
    "Resource": "*"
},
```

AWS DMS funktioniert auch mit KMS-Schlüsselalassen. Weitere Informationen zum Erstellen Ihrer eigenen AWS KMS-Schlüssel und zum Gewähren des Zugriffs auf einen KMS-Schlüssel für Benutzer finden Sie im [AWS KMS-Entwicklerhandbuch](#).

Wenn Sie keine KMS-Schlüsselkennung angeben, verwendet AWS DMS Ihren Standardverschlüsselungsschlüssel. AWS KMS erstellt den Standardverschlüsselungsschlüssel für AWS DMS für Ihr AWS-Konto. Ihr AWS-Konto verfügt für jede AWS-Region über einen anderen Standardverschlüsselungsschlüssel.

Verwenden Sie zur Verwaltung der AWS KMS-Schlüssel für die Verschlüsselung Ihrer AWS DMS-Ressourcen den AWS Key Management Service. AWS KMS kombiniert sichere, hoch verfügbare Hard- und Software, um ein System für die Schlüsselverwaltung bereitzustellen, das für die Cloud skaliert ist. Mit AWS KMS können Sie Verschlüsselungsschlüssel erstellen und Richtlinien definieren, die steuern, wie diese Schlüssel verwendet werden können.

Sie finden AWS KMS in der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter <https://console.aws.amazon.com/kms>.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie eine der folgenden Optionen, um mit AWS KMS-Schlüsseln zu arbeiten:
 - Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus.
 - Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus.

AWS KMS unterstützt AWS CloudTrail, so dass Sie die Schlüsselverwendung prüfen und sicherstellen können, dass die Schlüssel korrekt verwendet werden. Ihre AWS KMS-Schlüssel können in Kombination mit AWS DMS und unterstützten AWS-Services wie Amazon RDS, Amazon S3, Amazon Redshift und Amazon EBS verwendet werden.

Sie können auch benutzerdefinierte AWS KMS-Schlüssel speziell zum Verschlüsseln von Zieldaten für die folgenden AWS DMS-Endpunkte erstellen:

- Amazon Redshift – Weitere Informationen finden Sie unter [Erstellen und Verwenden von AWS KMS-Schlüsseln für die Verschlüsselung von Amazon-Redshift-Zieldaten](#).
- Amazon S3 – Weitere Informationen finden Sie unter [Erstellen von AWS KMS-Schlüsseln zum Verschlüsseln von Amazon-S3-Zielobjekten](#).

Nachdem Sie Ihre AWS DMS-Ressourcen mit einem KMS-Schlüssel erstellt haben, können Sie den Verschlüsselungsschlüssel für diese Ressourcen nicht mehr ändern. Stellen Sie sicher, die Anforderungen für Ihre Verschlüsselungsschlüssel zu definieren, bevor Sie Ihre AWS DMS-Ressourcen erstellen.

Netzwerksicherheit für AWS Database Migration Service

Die Sicherheitsanforderungen für das Netzwerk, das Sie erstellen, wenn Sie AWS Database Migration Service verwenden, hängen davon ab, wie Sie das Netzwerk konfigurieren. Die allgemeinen Regeln zur Netzwerksicherheit für AWS DMS lauten wie folgt:

- Die Replikations-Instance muss Zugriff auf die Quell- und Zielendpunkte haben. Die Sicherheitsgruppe für die Replikations-Instance muss Netzwerk-ACLs oder Regeln haben, die den Ausgang von der Instance über den Datenbankport zu den Datenbankendpunkten erlauben.
- Datenbankendpunkte müssen Netzwerk-ACLs und Sicherheitsgruppenregeln beinhalten, die eingehenden Zugriff von der Replikations-Instance erlauben. Sie können dies mithilfe der Sicherheitsgruppe der Replikations-Instance, der privaten IP-Adresse, der öffentlichen IP-Adresse oder der öffentlichen Adresse des NAT-Gateways erreichen, je nach Konfiguration.
- Wenn Ihr Netzwerk einen VPN-Tunnel verwendet, muss die Amazon-EC2-Instance, die als NAT-Gateway fungiert, eine Sicherheitsgruppe mit Regeln verwenden, die der Replikations-Instance erlauben, Datenverkehr durch sie zu senden.

Standardmäßig hat die VPC-Sicherheitsgruppe, die von der AWS DMS-Replikations-Instance verwendet wird, Regeln, die den Ausgang zu 0.0.0.0/0 auf allen Ports erlauben. Wenn Sie diese Sicherheitsgruppe ändern oder Ihre eigene Sicherheitsgruppe Egress verwenden, muss der Ausgang mindestens zu den Quell- und Zielendpunkten auf den jeweiligen Datenbankports zugelassen werden.

Die Netzwerkkonfigurationen, die Sie für die Datenbankmigration verwenden können, erfordern spezifische Sicherheitsüberlegungen:

- [Konfiguration mit allen Datenbankmigrationskomponenten in einer VPC](#) – Die Sicherheitsgruppe, die von den Endpunkten verwendet wird, muss den Eingang über den Datenbankport von der Replikations-Instance erlauben. Stellen Sie sicher, dass die von der Replikations-Instance verwendete Sicherheitsgruppe Eingang zu den Endpunkten hat, oder Sie können eine Regel in der von den Endpunkten verwendeten Sicherheitsgruppe erstellen, die der privaten IP-Adresse der Replikations-Instance den Zugriff erlaubt.
- [Konfiguration mit mehreren VPCs](#) – Die Sicherheitsgruppe, die von der Replikations-Instance verwendet wird, muss eine Regel für den VPC-Bereich und den DB-Port auf der Datenbank haben.
- [Konfiguration für ein Netzwerk zu einer VPC unter Verwendung AWS Direct Connect eines VPN](#) – ein VPN-Tunnel, der Datenverkehr durch einen Tunnel aus der VPC in ein On-Premises-

VPN erlaubt. Bei dieser Konfiguration enthält das VPC eine Routing-Regel, die Datenverkehr, der für eine IP-Adresse oder einen IP-Bereich bestimmt ist, zu einem Host sendet, der den Datenverkehr aus der VPC in das lokale VPN leiten kann. In diesem Fall verfügt der NAT-Host über eigene Sicherheitsgruppeneinstellungen, die den Datenverkehr von der privaten IP-Adresse oder Sicherheitsgruppe der Replikations-Instance zur NAT-Instance zulassen müssen.

- [Konfiguration für ein Netzwerk zu einer VPC über das Internet](#) – Die VPC-Sicherheitsgruppe muss Routing-Regeln enthalten, die den nicht für die VPC bestimmten Datenverkehr an das Internet-Gateway sendet. In dieser Konfiguration scheint es, dass die Verbindung zum Endpunkt von der öffentlichen IP-Adresse auf der Replikations-Instance kommt.
- [Konfiguration mit einer RDS-DB-Instance, die sich nicht in einer VPC befindet, zu einer DB-Instance in einer VPC mit ClassicLink](#) – Wenn die Amazon-RDS-Quell- oder Ziel-DB-Instance sich nicht in einer VPC befindet und keine Sicherheitsgruppe mit der VPC teilt, in der sich die Replikations-Instance befindet, können Sie einen Proxy-Server einrichten und ClassicLink verwenden, um die Quell- und Zieldatenbanken zu verbinden.
- Der Quellendpunkt befindet sich außerhalb der von der Replikations-Instance verwendeten VPC (über NAT-Gateway) – Sie können ein Network Address Translation (NAT)-Gateway mit einer einzelnen Elastic-IP-Adresse konfigurieren, die an eine einzelne Elastic-Network-Schnittstelle gebunden ist. Diese Elastic Network-Schnittstelle empfängt dann eine NAT-Kennung (nat-######). Wenn die VPC eine Standardroute zu diesem NAT-Gateway anstatt zum Internet-Gateway enthält, scheint die Replikations-Instance den Datenbankendpunkt mit der öffentlichen IP-Adresse des Internet-Gateways zu kontaktieren. In diesem Fall muss der Eingang zum Datenbankendpunkt außerhalb der VPC den Eingang von der NAT-Adresse anstatt von der öffentlichen IP-Adresse der Replikations-Instance erlauben.
- VPC-Endpunkte für Nicht-RDBMS-Engines – AWS DMS unterstützt keine VPC-Endpunkte für Nicht-RDBMS-Engines.

Verwenden von SSL mit AWS Database Migration Service

Sie können Verbindungen für Quell- und Zielendpunkte mithilfe von SSL (Secure Sockets Layer) verschlüsseln. Um dies zu tun, können Sie mithilfe der AWS DMS-Managementkonsole oder der AWS DMS-API ein Zertifikat einem Endpunkt zuweisen. Sie können mithilfe der AWS DMS-Konsole auch Ihre Zertifikate verwalten.

Nicht alle Datenbanken verwenden SSL auf die gleiche Weise. Amazon Aurora MySQL-Compatible Edition verwendet den Servernamen, den Endpunkt der primären Instance im Cluster, als Endpunkt für SSL. Ein Amazon Redshift-Endpunkt verwendet bereits eine SSL-Verbindung und erfordert nicht die Einrichtung einer SSL-Verbindung durch AWS DMS. Ein Oracle-Endpunkt erfordert zusätzliche Schritte. Weitere Informationen finden Sie unter [SSL-Unterstützung für einen Oracle-Endpunkt](#).

Themen

- [Einschränkungen bei der Verwendung von SSL mit AWS DMS](#)
- [Verwalten von Zertifikaten](#)
- [Aktivieren von SSL für einen MySQL-kompatiblen, PostgreSQL- oder SQL Server-Endpunkt](#)

Um ein Zertifikat einem Endpunkt zuzuweisen, geben Sie das Stammzertifikat oder die Kette der Zertifikate von Zwischenzertifizierungsstellen an, die zu dem Stammzertifikat führen (als Zertifikat-Paket), mit dem das Server-SSL-Zertifikat signiert wurde, das auf dem Endpunkt bereitgestellt ist. Zertifikate werden nur als PEM-formatierte X.509-Dateien akzeptiert. Wenn Sie ein Zertifikat importieren, erhalten Sie einen ARN (Amazon Resource Name), mit dem Sie dieses Zertifikat für einen Endpunkt angeben können. Wenn Sie Amazon RDS verwenden, können Sie die Stammzertifizierungsstelle und das Zertifikatbündel herunterladen, die in der von Amazon RDS gehosteten `rds-combined-ca-bundle.pem`-Datei bereitgestellt werden. Weitere Informationen zum Herunterladen dieser Datei finden Sie unter [Verwenden von SSL/TLS für die Verschlüsselung einer Verbindung zu einer DB-Instance](#) im Amazon-RDS-Benutzerhandbuch.

Sie können verschiedene SSL-Modi für die Überprüfung des SSL-Zertifikats auswählen.

- `none` – Die Verbindung ist nicht verschlüsselt. Diese Option ist nicht sicher, erfordert jedoch weniger Aufwand.
- `require` – Die Verbindung ist mit SSL (TLS) verschlüsselt, es wird aber keine CA-Verifizierung durchgeführt. Diese Option ist sicherer und erfordert mehr Aufwand.
- `verify-ca` – Die Verbindung ist verschlüsselt. Diese Option ist sicherer und erfordert mehr Aufwand. Diese Option überprüft das Serverzertifikat.

- **verify-full** – Die Verbindung ist verschlüsselt. Diese Option ist sicherer und erfordert mehr Aufwand. Diese Option überprüft das Serverzertifikat und prüft, ob der Server-Hostname dem Hostnamen-Attribut für das Zertifikat entspricht.

Nicht alle SSL-Modi funktionieren mit allen Datenbankendpunkten. Die folgende Tabelle zeigt, welche SSL-Modi für jede Datenbank-Engine unterstützt werden.

DB-Engine	Keine	require	verify-ca	verify-full
MySQL/MariaDB/ Amazon Aurora MySQL	Standard	Nicht unterstütz t	Unterstützt	Unterstützt
Microsoft SQL Server	Standard	Unterstützt	Nicht unterstütz t	Unterstützt
PostgreSQL	Standard	Unterstützt	Unterstützt	Unterstützt
Amazon Redshift	Standard	SSL nicht aktiviert	SSL nicht aktiviert	SSL nicht aktiviert
Oracle	Standard	Nicht unterstütz t	Unterstützt	Nicht unterstütz t
SAP ASE	Standard	SSL nicht aktiviert	SSL nicht aktiviert	Unterstützt
MongoDB	Standard	Unterstützt	Nicht unterstütz t	Unterstützt
Db2 LUW	Standard	Nicht unterstütz t	Unterstützt	Nicht unterstütz t
Db2 für z/OS	Standard	Nicht unterstütz t	Unterstützt	Nicht unterstütz t

Note

Die Option SSL-Modus auf der DMS-Konsole oder API gilt nicht für einige Daten-Streaming- und NoSQL-Dienste wie Kinesis und DynamoDB. Sie sind standardmäßig sicher, daher zeigt DMS an, dass die Einstellung für den SSL-Modus auf „Keine“ gesetzt ist (VSSL-Modus=Keine). Sie müssen keine zusätzliche Konfiguration für Ihren Endpunkt angeben, um SSL verwenden zu können. Wenn Sie beispielsweise Kinesis als Zielendpunkt verwenden, ist dies standardmäßig sicher. Alle API-Aufrufe an Kinesis verwenden SSL, so dass keine zusätzliche SSL-Option am DMS-Endpunkt erforderlich ist. Mithilfe des HTTPS-Protokolls, das DMS standardmäßig verwendet, wenn eine Verbindung zu einem Kinesis-Datenstrom hergestellt wird, können Sie Daten sicher über SSL-Endpunkte speichern und abrufen.

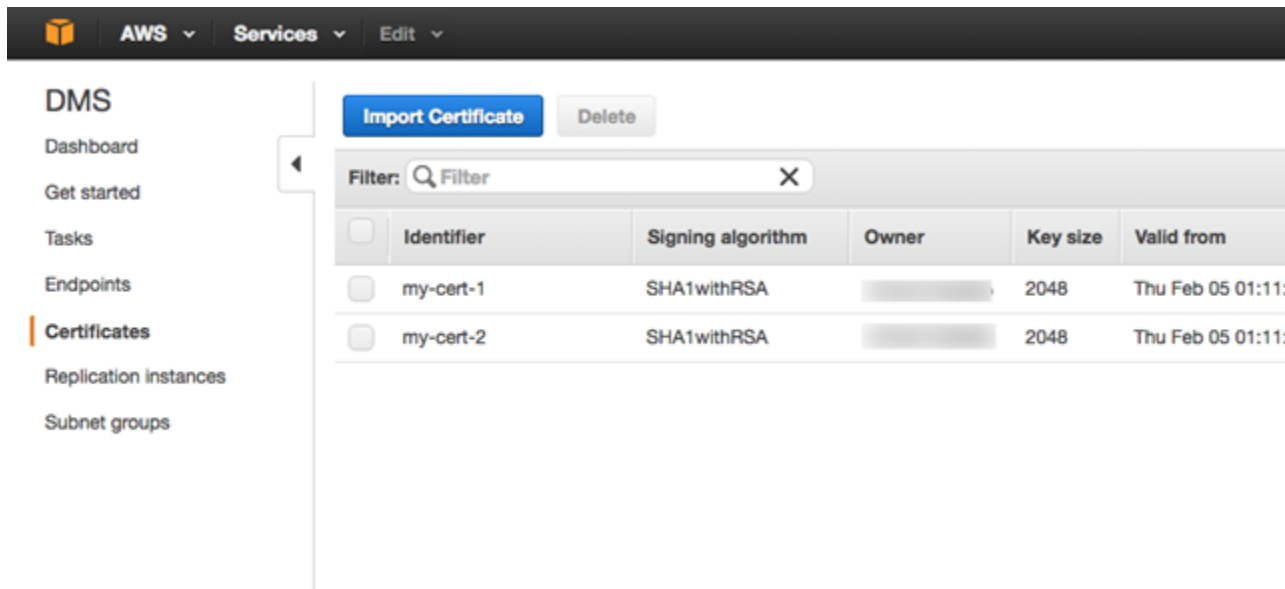
Einschränkungen bei der Verwendung von SSL mit AWS DMS

Im Folgenden werden Einschränkungen bei der Verwendung von SSL mit AWS DMS aufgeführt:

- SSL-Verbindungen zu Amazon-Redshift-Zielendpunkten werden nicht unterstützt. AWS DMS verwendet einen Amazon-S3-Bucket zum Übertragen von Daten zur Amazon-Redshift-Datenbank. Diese Übertragung wird von Amazon Redshift standardmäßig verschlüsselt.
- SQL-Timeouts können auftreten, wenn CDC Aufgaben (Change Data Capture) mit SSL-fähigen Oracle-Endpunkten durchgeführt werden. Wenn dieses Problem auftritt, wobei CDC-Zähler nicht die erwarteten Zahlen anzeigen, legen Sie den Parameter `MinimumTransactionSize` aus dem Abschnitt `ChangeProcessingTuning` der Aufgabeneinstellungen auf einen niedrigeren Wert fest. Sie können mit einem Wert von 100 beginnen. Weitere Informationen zum Parameter `MinimumTransactionSize` erhalten Sie unter [Einstellungen für die Optimierung der Verarbeitung von Änderungen](#).
- Sie können Zertifikate nur in den Formaten `.pem` und `.sso` (Oracle Wallet) importieren.
- In einigen Fällen wird Ihr Server-SSL-Zertifikat möglicherweise von einer Zwischenzertifizierungsstelle signiert. Wenn dies der Fall ist, stellen Sie sicher, dass die gesamte Zertifikatkette, die von der Zwischenzertifizierungsstelle bis zur Stammzertifizierungsstelle führt, als einzelne `.pem`-Datei importiert wird.
- Wenn Sie selbstsignierte Zertifikate auf Ihrem Server verwenden, wählen Sie `require` als SSL-Modus aus. Der SSL-Modus `require` (erforderlich) vertraut dem SSL-Zertifikat des Servers implizit und prüft nicht, ob das Zertifikat von einer Zertifizierungsstelle signiert wurde.

Verwalten von Zertifikaten

Sie können mithilfe der DMS-Konsole Ihre SSL-Zertifikate anzeigen und verwalten. Sie können auch Ihre Zertifikate mithilfe der DMS-Konsole importieren.



Aktivieren von SSL für einen MySQL-kompatiblen, PostgreSQL- oder SQL Server-Endpoint

Sie können eine SSL-Verbindung zu einem neu erstellten Endpoint oder zu einem vorhandenen Endpoint hinzufügen.

So erstellen Sie einen AWS DMS-Endpoint mit SSL

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als AWS Identity and Access Management (IAM)-Benutzer angemeldet sind, müssen Sie über die entsprechenden Berechtigungen für den Zugriff auf AWS DMS verfügen. Weitere Informationen zu den erforderlichen Berechtigungen für die Datenbankmigration finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich Certificates aus.
3. Wählen Sie Import Certificate aus.
4. Laden Sie das Zertifikat hoch, das Sie zur Verschlüsselung der Verbindung zu einem Endpoint verwenden möchten.

Note

Sie können auch ein Zertifikat mithilfe der AWS DMS-Konsole hochladen, wenn Sie einen Endpunkt erstellen oder ändern, indem Sie Add new CA certificate (Neues CA-Zertifikat hinzufügen) auf der Seite Create database endpoint (Datenbank-Endpoint erstellen) auswählen.

Für Aurora Serverless als Ziel holen Sie sich das Zertifikat, das unter [Verwendung von TLS/SSL mit Aurora Serverless](#) beschrieben ist.

5. Erstellen Sie einen Endpunkt wie beschrieben unter [Schritt 2: Angeben von Quell- und Zielendpunkten](#)

So ändern Sie einen vorhandenen AWS DMS-Endpoint für die Verwendung von SSL:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als IAM-Benutzer angemeldet sind, müssen Sie über die entsprechenden Berechtigungen für den Zugriff auf AWS DMS verfügen. Weitere Informationen zu den erforderlichen Berechtigungen für die Datenbankmigration finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich Certificates aus.
3. Wählen Sie Import Certificate aus.
4. Laden Sie das Zertifikat hoch, das Sie zur Verschlüsselung der Verbindung zu einem Endpunkt verwenden möchten.

Note

Sie können auch ein Zertifikat mithilfe der AWS DMS-Konsole hochladen, wenn Sie einen Endpunkt erstellen oder ändern, indem Sie Add new CA certificate (Neues CA-Zertifikat hinzufügen) auf der Seite Create database endpoint (Datenbank-Endpoint erstellen) auswählen.

5. Wählen Sie im Navigationsbereich die Option Endpoints aus. Wählen Sie dann den Endpunkt aus, den Sie ändern möchten, und klicken Sie auf Modify.
6. Wählen Sie einen Wert für den SSL-Modus.

Wenn Sie den Modus `verify-ca` oder `verify-full` auswählen, geben Sie das Zertifikat an, das Sie für das CA-Zertifikat verwenden möchten (siehe unten).

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. Wählen Sie `Ändern` aus.
8. Wenn der Endpunkt geändert wurde, wählen Sie den Endpunkt aus, und klicken Sie auf `Test connection` (Verbindung prüfen), um festzustellen, ob die SSL-Verbindung funktioniert.

Nachdem Sie die Quell- und Zielendpunkte erstellt haben, erstellen Sie eine Aufgabe, die diese Endpunkte verwendet. Weitere Informationen zum Erstellen einer Aufgabe finden Sie unter [Schritt 3: Erstellen einer Aufgabe und Migrieren der Daten](#).

Ändern des Datenbankpassworts

In den meisten Fällen lässt sich das Datenbankpasswort für Ihren Quell- oder Zielendpunkt einfach ändern. Falls Sie das Datenbankpasswort für einen Endpunkt ändern müssen, den Sie gegenwärtig in einer Migrations- oder Replikationsaufgabe verwenden, ist der Prozess etwas komplexer. Das folgende Verfahren zeigt, wie Sie dies tun können.

So ändern Sie das Datenbankpasswort für einen Endpunkt in einer Migrations- oder Replikationsaufgabe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS DMS-Konsole unter <https://console.aws.amazon.com/dms/v2/>.

Wenn Sie als IAM-Benutzer angemeldet sind, müssen Sie über die entsprechenden Berechtigungen für den Zugriff auf AWS DMS verfügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [Erforderliche IAM-Berechtigungen zur Verwendung von AWS DMS](#).

2. Wählen Sie im Navigationsbereich die Option Datenbankmigrationsaufgaben aus.
3. Wählen Sie die Aufgabe aus, die den Endpunkt verwendet, für den Sie das Datenbankpasswort ändern möchten, und klicken Sie dann auf Stop.
4. Während die Aufgabe beendet wird, können Sie das Passwort der Datenbank für den Endpunkt mithilfe der nativen Tools ändern, die Sie für die Arbeit mit der Datenbank verwenden.
5. Kehren Sie zur DMS Management Console zurück und wählen Sie im Navigationsbereich Endpoints aus.
6. Wählen Sie den Endpunkt für die Datenbank aus, für die Sie das Passwort geändert haben, und klicken Sie dann auf Modify.
7. Geben Sie das neue Passwort in das Feld Passwort ein und wählen Sie Ändern aus.
8. Wählen Sie im Navigationsbereich die Option Datenbankmigrationsaufgaben aus.
9. Wählen Sie die Aufgabe aus, die Sie zuvor gestoppt haben, und wählen Sie Neustart/Fortsetzen aus.
10. Wählen Sie entweder Neustart oder Fortsetzen aus, je nachdem, wie Sie mit der Aufgabe fortfahren möchten, und wählen Sie dann Aufgabe starten aus.

Kontingente für AWS Database Migration Service

Im Folgenden finden Sie die Ressourcen-Limits und Benennungseinschränkungen für AWS Database Migration Service (AWS DMS).

Die maximale Größe einer Datenbank, die AWS DMS migrieren kann, hängt von einer Reihe von Faktoren ab. Dazu gehören die Quellumgebung, die Verteilung von Daten in der Quelldatenbank und der Auslastung des Quellsystems.

Ob Ihr System für AWS DMS geeignet ist, können Sie am besten feststellen, indem Sie es testen. Beginnen Sie langsam, damit Sie sich mit der Konfiguration vertraut machen können. Fügen Sie dann einige komplexe Objekte hinzu. Versuchen Sie abschließend testweise eine Volllast.

Ressourcenkontingente für AWS Database Migration Service

Für jedes AWS-Konto gelten für jede AWS-Region Kontingente für die Anzahl der AWS DMS-Ressourcen, die erstellt werden können. Nachdem das Kontingent für eine Ressource erreicht wurde, schlagen zusätzliche Aufrufe zum Erstellen dieser Ressource mit einer Ausnahme fehl.

In der folgenden Tabelle sind die AWS DMS-Ressourcen und ihre Kontingente pro AWS-Region aufgeführt.

Ressource	Standardkontingent
Drosselung von API-Anforderungen	Maximal 200 Anforderungen pro Sekunde
API-Anforderungsaktualisierungsrate	8 Anforderungen pro Sekunde
Replikations-Instances pro Benutzerkonto	60
Gesamt Speichermenge für eine Replikations-Instance	30 000 GB
Event-Abonnements pro Benutzerkonto	60
Replikationssubnetzgruppen pro Benutzerkonto	60

Ressource	Standardkontingent
Subnetze pro Replikations-Subnetzgruppe	60
Endpunkte pro Benutzerkonto	1.000
Endpunkte pro Replikations-Instance	100
Aufgaben pro Benutzerkonto	600
Aufgaben pro Replikations-Instance	200
Zertifikate pro Benutzerkonto	100
Datenanbieter pro Benutzerkonto	1.000
Instance-Profile pro Benutzerkonto	60
Migrationsprojekte pro Benutzerkonto	10
DMS-Datenkollektoren pro Benutzerkonto	10
Zielempfehlungen, die gleichzeitig generiert werden	100
Anzahl der Dateien, die der DMS-Datenkollektor pro Stunde hochladen kann	500
Homogene Datenmigrationen pro Benutzerkonto	600
Homogene Datenmigrationen, die gleichzeitig ausgeführt werden	100
Homogene Datenmigrationen pro Migrationsprojekt	10
Serverless-Replikationen	100

Weitere Informationen zur Drosselung von API-Anfragen und zur Aktualisierungsrate finden Sie unter [Informationen zur API-Anforderungsdrosselung](#).

Das Speicherkontingent von 30 000 GB gilt für alle Ihre AWS DMS-Replikations-Instances in einer bestimmten AWS-Region. Dieser Speicher dient zum Zwischenspeichern von Änderungen, falls das Ziel nicht mit einer Quelle mithalten kann, sowie zum Speichern von Protokollinformationen.

Informationen zur API-Anforderungsdrosselung

AWS DMS unterstützt ein variierendes, aber maximales API-Anforderungskontingent von 200 API-Aufrufen pro Sekunde. Mit anderen Worten: Ihre API-Anforderungen werden gedrosselt, wenn sie diese Rate überschreiten. Außerdem können Sie sich auf weniger API-Aufrufe pro Sekunde beschränken, je nachdem, wie lange AWS DMS dafür braucht, Ihr Kontingent zu aktualisieren, bevor Sie eine weitere API-Anforderung stellen. Dieses Kontingent gilt sowohl, wenn Sie API-Aufrufe direkt tätigen, als auch, wenn sie in Ihrem Namen im Rahmen der Nutzung der AWS DMS Management Console erfolgen.

Um zu verstehen, wie die Drosselung von API-Anforderungen funktioniert, ist es hilfreich, sich vorzustellen, dass AWS DMS einen Token-Bucket verwaltet, der Ihre API-Anfragen verfolgt. In diesem Szenario können Sie mit jedem Token im Bucket einen einzigen API-Aufruf tätigen. Sie können nicht mehr als 200 Token gleichzeitig in dem Bucket haben. Wenn Sie einen API-Aufruf tätigen, entfernt AWS DMS ein Token aus dem Bucket. Wenn Sie in weniger als einer Sekunde 200 API-Aufrufe tätigen, ist Ihr Bucket leer und jeder Versuch, einen weiteren API-Aufruf zu tätigen, schlägt fehl. Für jede Sekunde, in der Sie keinen API-Aufruf tätigen, fügt AWS DMS dem Bucket 8 Token hinzu, bis das Maximum von 200 Token erreicht ist. Dies ist die Aktualisierungsrate für AWS DMS-API-Anforderungen. Sobald Sie nach der Drosselung Token zu Ihrem Bucket hinzugefügt haben, können Sie jederzeit so viele zusätzliche API-Aufrufe tätigen, wie Token verfügbar sind, bis Ihre Aufrufe wieder gedrosselt werden.

Wenn Sie die AWS CLI verwenden, um API-Aufrufe auszuführen, die gedrosselt sind, gibt AWS DMS einen Fehler wie den folgenden zurück:

```
An error occurred (ThrottlingException) when calling the AwsDmsApiCall operation  
(reached max retries: 2): Rate exceeded
```

Hier ist *AwsDmsApiCall* der Name des AWS DMS-API-Vorgangs, der gedrosselt wurde, zum Beispiel *DescribeTableStatistics*. Sie können es dann nach ausreichender Verzögerung erneut versuchen oder einen anderen Aufruf tätigen, um eine Drosselung zu vermeiden.

Note

Im Gegensatz zur Drosselung von API-Anfragen, die von einigen anderen Services wie Amazon EC2 verwaltet wird, können Sie keine Erhöhung der von AWS DMS verwalteten API-Anforderungsdrosselungsquoten anordnen.

Fehlerbehebung für Migrationsaufgaben in AWS Database Migration Service

Im Folgenden finden Sie Themen zur Fehlersuche mit dem AWS Database Migration Service (AWS DMS). Diese Themen können Ihnen helfen, häufig auftretende Probleme bei der Verwendung von AWS DMS und ausgewählten Endpunktdatenbanken zu beheben.

Wenn Sie einen AWS-Support-Fall eröffnet haben, identifiziert der Support-Techniker möglicherweise ein potenzielles Problem mit einer Konfiguration Ihrer Endpunktdatenbank. Der Techniker bittet Sie möglicherweise außerdem, ein Support-Skript auszuführen, um Diagnoseinformationen zu Ihrer Datenbank zu erhalten. Einzelheiten zum Herunterladen, Ausführen und Hochladen der Diagnoseinformationen aus dieser Art von Support-Skript finden Sie unter [Arbeiten mit Diagnoseunterstützungsskripts in AWS DMS](#).

Zur Fehlerbehebung AWS DMS sammelt Trace- und Dump-Dateien in der Replikations-Instance. Sie können diese Dateien dem -AWSSupport zur Verfügung stellen, falls ein Problem auftritt, das eine Fehlerbehebung erfordert. Standardmäßig löscht DMS Trace- und Dump-Dateien, die älter als XML-Tage sind. Um die Erfassung von Ablaufverfolgungs- und Dump-Dateien zu deaktivieren, öffnen Sie einen Fall beim -AWSSupport.

Themen

- [Migrationsaufgaben werden langsam ausgeführt](#)
- [Die Statusleiste der Aufgabe bewegt sich nicht](#)
- [Die Aufgabe ist abgeschlossen, aber es wurde nichts migriert](#)
- [Fremdschlüssel und sekundäre Indizes fehlen](#)
- [AWS DMS erstellt keine CloudWatch Protokolle](#)
- [Bei der Verbindung mit Amazon RDS treten Probleme auf](#)
- [Es treten Netzwerkprobleme auf](#)
- [CDC ist nach Volllast hängen geblieben](#)
- [Fehler durch Primärschlüsselverletzung beim Neustart einer Aufgabe](#)
- [Erstes Laden des Schemas fehlgeschlagen](#)
- [Aufgaben mit unbekanntem Fehler fehlgeschlagen](#)
- [Bei erneutem Laden der Aufgabe werden Tabellen von Beginn an geladen](#)

- [Die Anzahl der Tabellen pro Aufgabe verursacht Probleme](#)
- [Aufgaben schlagen fehl, wenn ein Primärschlüssel in der LOB-Spalte erstellt wird](#)
- [Doppelte Datensätze in einer Zieltabelle ohne Primärschlüssel](#)
- [Quellendpunkte fallen in den reservierten IP-Bereich](#)
- [Zeitstempel sind in Amazon-Athena-Abfragen unleserlich](#)
- [Fehlersuche bei Verwendung von Oracle](#)
- [Fehlersuche bei Verwendung von MySQL](#)
- [Fehlersuche bei Verwendung von PostgreSQL](#)
- [Fehlersuche bei Verwendung von Microsoft SQL Server](#)
- [Fehlersuche bei Verwendung von Amazon Redshift](#)
- [Fehlersuche bei Verwendung von Amazon Aurora MySQL](#)
- [Fehlersuche bei Verwendung von SAP ASE](#)
- [Fehlersuche bei Verwendung von IBM Db2](#)
- [Fehlersuche bei Latenzproblemen in AWS Database Migration Service](#)
- [Arbeiten mit Diagnoseunterstützungsskripts in AWS DMS](#)
- [Arbeiten mit dem AWS DMS-Diagnoseunterstützungs-AMI](#)

Migrationsaufgaben werden langsam ausgeführt

Verschiedene Probleme können dazu führen, dass eine Migrationsaufgabe langsam ausgeführt wird oder dass nachfolgende Aufgaben langsamer ausgeführt werden als die erste Aufgabe.

Der häufigste Grund für ein langsames Ausführen von Migrationsaufgaben ist, dass der AWS DMS-Replikations-Instance unzureichende Ressourcen zugewiesen sind. Überprüfen Sie die Nutzung von CPU, Arbeitsspeicher, Swap-Dateien und IOPS durch Ihre Replikations-Instance, um sicherzustellen, dass die Instance über ausreichend Ressourcen für die ausgeführte Aufgabe verfügt. So ist beispielsweise die Ausführung mehrerer Aufgaben mit Amazon Redshift als Endpunkt recht E/A-intensiv. Sie können die IOPS für Ihre Replikations-Instance erhöhen oder Ihre Aufgaben über mehrere Replikations-Instances verteilen, um die Migration effizienter zu gestalten.

Weitere Informationen zur Bestimmung der Größe Ihrer Replikations-Instance finden Sie unter [Auswahl der besten Größe für eine Replikations-Instance](#).

Sie können die Geschwindigkeit eines ersten Ladevorgangs bei einer Migration wie folgt erhöhen:

- Wenn Ihr Ziel eine DB-Instance von Amazon RDS ist, stellen Sie sicher, dass Multi-AZ nicht für die DB-Ziel-Instance aktiviert ist.
- Deaktivieren Sie alle automatischen Backups oder Protokollierungen für die Zieldatenbank während des Ladevorgangs und aktivieren Sie sie wieder, wenn die Migration abgeschlossen ist.
- Verwenden Sie bereitgestellte IOPS, sofern dieses Feature für Ihr Ziel verfügbar ist.
- Wenn Ihre Migrationsdaten LOBs enthalten, stellen Sie sicher, dass die Aufgabe für eine LOB-Migration optimiert ist. Weitere Informationen zur Optimierung für LOBs finden Sie unter [Ziel-Metadaten-Aufgabeneinstellungen](#).

Die Statusleiste der Aufgabe bewegt sich nicht

Die Aufgabenstatusleiste bietet eine Schätzung des Fortschritts der Aufgabe. Die Qualität dieser Schätzung hängt von der Qualität der Tabellenstatistik der Quelldatenbank ab. Je besser die Tabellenstatistik, desto genauer die Schätzung.

Wenn eine Aufgabe nur eine Tabelle ohne geschätzte Zeilenstatistik enthält, kann AWS DMS den prozentualen Fortschritt nicht berechnen. Überprüfen Sie in diesem Fall anhand des Aufgabenstatus und der Angabe der geladenen Zeilen, ob die Aufgabe tatsächlich ausgeführt wird und Fortschritte macht.

Die Aufgabe ist abgeschlossen, aber es wurde nichts migriert

Gehen Sie wie folgt vor, wenn nach Abschluss Ihrer Aufgabe nichts migriert wurde.

- Überprüfen Sie, ob der Benutzer, der den Endpunkt erstellt hat, Lesezugriff auf die Tabelle hat, die Sie migrieren möchten.
- Überprüfen Sie, ob es sich bei dem Objekt, das Sie migrieren möchten, um eine Tabelle handelt. Wenn es sich um eine Ansicht handelt, aktualisieren Sie die Tabellenzuordnungen und geben Sie den Objekt-Locator als „Ansicht“ oder „Alle“ an. Weitere Informationen finden Sie unter [Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole](#).

Fremdschlüssel und sekundäre Indizes fehlen

AWS DMS erstellt Tabellen, Primärschlüssel und in einigen Fällen eindeutige Indizes, jedoch keine anderen Objekte, die für eine effiziente Migration der Daten von der Quelle nicht erforderlich sind.

So erstellt DMS z. B. keine sekundären Indizes, Nicht-Primärschlüssel-Beschränkungen oder Datenstandardwerte.

Um sekundäre Objekte von der Datenbank zu migrieren, verwenden Sie die nativen Tools der Datenbank, wenn Sie die Migration auf derselben Datenbank-Engine wie Ihre Quelldatenbank durchführen. Verwenden Sie das AWS Schema Conversion Tool (AWS SCT), wenn Sie die Migration auf eine andere Datenbank-Engine durchführen als die, die von Ihrer Quelldatenbank für die Migration sekundärer Objekte verwendet wird.

AWS DMS erstellt keine CloudWatch Protokolle

Wenn Ihre Replikationsaufgabe keine CloudWatch Protokolle erstellt, stellen Sie sicher, dass Ihr Konto über die `dms-cloudwatch-logs-role` Rolle verfügt. Wenn diese Rolle nicht vorhanden ist, erstellen Sie sie wie folgt:

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie die Registerkarte Rollen aus. Wählen Sie Rolle erstellen aus.
3. Wählen Sie im Abschnitt Typ der vertrauenswürdigen Entität auswählen die Option AWS-Service aus.
4. Wählen Sie im Abschnitt Wählen Sie einen Anwendungsfall aus DMS aus.
5. Wählen Sie Weiter: Berechtigungen aus.
6. Geben Sie **AmazonDMSCloudWatchLogsRole** in das Suchfeld ein und aktivieren Sie das Kontrollkästchen neben AmazonDMSCloudWatchLogsRole. Dadurch werden AWS DMS Berechtigungen für den Zugriff auf erteilt CloudWatch.
7. Wählen Sie Next: Tags (Weiter: Tags) aus.
8. Klicken Sie auf Weiter: Prüfen.
9. Geben Sie für Rollennamen **dms-cloudwatch-logs-role** ein. Bei diesem Namen wird zwischen Groß- und Kleinschreibung unterschieden.
10. Wählen Sie Rolle erstellen aus.

Bei der Verbindung mit Amazon RDS treten Probleme auf

Es kann verschiedene Gründe dafür geben, dass Sie keine Verbindung mit einer DB-Instance von Amazon RDS herstellen können, die Sie als Quelle oder Ziel festgelegt haben. Folgende Punkte sollten Sie überprüfen:

- Stellen Sie sicher, dass die Kombination aus Benutzername und Passwort korrekt ist.
- Stellen Sie sicher, dass der Endpunktwert, der in der Amazon-RDS-Konsole für die Instance angezeigt wird, mit der Endpunktkennung übereinstimmt, die Sie zum Erstellen des AWS DMS-Endpunkts verwendet haben.
- Stellen Sie sicher, dass der in der Amazon-RDS-Konsole für die Instance angezeigte Port-Wert mit dem Port übereinstimmt, der dem AWS DMS-Endpunkt zugewiesen ist.
- Stellen Sie sicher, dass die der Amazon RDS-DB-Instance zugewiesene Sicherheitsgruppe Verbindungen von der AWS DMS-Replikations-Instance zulässt.
- Wenn sich die AWS DMS-Replikations-Instance und die DB-Instance von Amazon RDS nicht in derselben Virtual Private Cloud (VPC) befinden, vergewissern Sie sich, dass die DB-Instance öffentlich zugänglich ist.

Fehlermeldung: Falsche Thread-Verbindungszeichenfolge: Falscher Thread-Wert 0

Dieser Fehler kann häufig auftreten, wenn Sie die Verbindung zu einem Endpunkt testen. Diese Fehlermeldung weist auf einen Fehler in der Verbindungszeichenfolge hin. Ein Beispiel ist ein Leerzeichen nach der Host-IP-Adresse. Ein anderes Beispiel ist ein fehlerhaftes Zeichen, das in die Verbindungszeichenfolge kopiert wurde.

Es treten Netzwerkprobleme auf

Das häufigste Netzwerkproblem betrifft die VPC-Sicherheitsgruppe, die von der AWS DMS-Replikations-Instance verwendet wird. Standardmäßig verfügt diese Sicherheitsgruppe über Regeln, die ausgehenden Datenverkehr zu 0.0.0.0/0 auf allen Ports zulassen. In vielen Fällen ändern Sie diese Sicherheitsgruppe oder verwenden Ihre eigene Sicherheitsgruppe. Wenn dies der Fall ist, stellen Sie zumindest sicher, dass die Quell- und Zielendpunkte über ihre jeweiligen Datenbankports ausgehenden Datenverkehr erhalten.

Weitere Probleme im Zusammenhang mit der Konfiguration können Folgendes umfassen:

- Replikations-Instance sowie Quell- und Zielendpunkt in derselben VPC – Die von den Endpunkten verwendete Sicherheitsgruppe muss eingehenden Datenverkehr am Datenbankport von der Replikations-Instance zulassen. Stellen Sie sicher, dass die von der Replikations-Instance verwendete Sicherheitsgruppe Zugang zu den Endpunkten hat. Sie können auch eine Regel in der von den Endpunkten verwendeten Sicherheitsgruppe erstellen, die der privaten IP-Adresse der Replikations-Instance den Zugriff erlaubt.
- Der Quellendpunkt befindet sich außerhalb der von der Replikations-Instance verwendeten VPC (über Internet-Gateway). – Die VPC-Sicherheitsgruppe muss über Routing-Regeln verfügen, die Datenverkehr, der nicht für die VPC bestimmt ist, an das Internet-Gateway senden. In dieser Konfiguration scheint es, dass die Verbindung zum Endpunkt von der öffentlichen IP-Adresse auf der Replikations-Instance kommt.
- Der Quellendpunkt befindet sich außerhalb der von der Replikations-Instance verwendeten VPC (über NAT-Gateway). – Sie können ein Network Address Translation (NAT)-Gateway mit einer einzelnen Elastic-IP-Adresse konfigurieren, die an eine einzelne Elastic-Network-Schnittstelle gebunden ist. Dieses NAT-Gateway empfängt eine NAT-Kennung (nat-#####).

Unter Umständen enthält die VPC eine Standardroute zu diesem NAT-Gateway anstelle des Internet-Gateways. In solchen Fällen scheint die Replikations-Instance stattdessen den Datenbankendpunkt über die öffentliche IP-Adresse des NAT-Gateways zu kontaktieren. Dann muss der Eingang zum Datenbankendpunkt außerhalb der VPC den Eingang von der NAT-Adresse anstatt von der öffentlichen IP-Adresse der Replikations-Instance erlauben.

Informationen zur Verwendung Ihres eigenen On-Premises-Nameservers finden Sie unter [Verwenden Ihres eigenen Vor-Ort-Nameservers](#).

CDC ist nach Vollast hängen geblieben

Langsame oder hängen gebliebene Replikationsänderungen können nach einem vollständigen Migrationsladevorgang auftreten, wenn mehrere AWS DMS-Einstellungen miteinander in Konflikt stehen.

Nehmen wir beispielsweise an, dass der Parameter Zieltabellen-Vorbereitungsmodus auf Nichts unternehmen oder Kürzen gesetzt ist. In diesem Fall haben Sie AWS DMS angewiesen, keine Einstellungen an den Zieltabellen vorzunehmen. Dies umfasst die Erstellung primärer und eindeutiger Indizes. Wenn Sie keine primären oder eindeutigen Schlüssel für die Zieltabellen erstellt haben, führt AWS DMS eine vollständige Tabellenprüfung für jede Aktualisierung durch. Dadurch kann die Leistung erheblich beeinträchtigt werden.

Fehler durch Primärschlüsselverletzung beim Neustart einer Aufgabe

Dieser Fehler kann auftreten, wenn Daten von einer vorherigen Migrationsaufgabe in der Zieldatenbank verbleiben. Wenn der Parameter Zieltabellen-Vorbereitungsmodus auf Nichts unternehmen festgelegt ist, führt AWS DMS keine Vorbereitung für die Zieltabelle durch. Dies umfasst auch die Bereinigung von Daten aus einer vorherigen Aufgabe.

Um Ihre Aufgabe erneut zu starten und diese Fehler zu vermeiden, müssen Sie die in die Zieltabellen eingefügten Zeilen aus der vorherigen Aufgabenausführung entfernen.

Erstes Laden des Schemas fehlgeschlagen

Unter Umständen schlägt das erste Laden Ihrer Schemas möglicherweise mit dem Fehler `Operation:getSchemaListDetails:errType=, status=0, errMessage=, errDetails=` fehl.

In solchen Fällen verfügt das Benutzerkonto, das AWS DMS für die Verbindung mit dem Quellendpunkt verwendet, nicht über die erforderlichen Berechtigungen.

Aufgaben mit unbekanntem Fehler fehlgeschlagen

Unbekannte Fehler können verschiedene Ursachen haben. Oftmals lässt sich das Problem jedoch darauf zurückführen, dass der Replikations-Instance von AWS DMS unzureichende Ressourcen zugewiesen sind.

Überprüfen Sie die Nutzung von CPU, Arbeitsspeicher, Swap-Dateien und IOPS durch Ihre Replikations-Instance, um sicherzustellen, dass Ihre Instance über ausreichend Ressourcen für die Migration verfügt. Weitere Informationen zur Überwachung finden Sie unter [AWS Database Migration Service-Metriken](#).

Bei erneutem Laden der Aufgabe werden Tabellen von Beginn an geladen

AWS DMS startet den Ladevorgang von Tabellen neu von vorn, wenn der erste Ladevorgang einer Tabelle noch nicht abgeschlossen ist. Wenn eine Aufgabe neu gestartet wird, lädt AWS DMS die Tabellen neu von vorn, wenn der erste Ladevorgang nicht abgeschlossen wurde.

Die Anzahl der Tabellen pro Aufgabe verursacht Probleme

Es gibt keine festgelegte Begrenzung für die Anzahl der Tabellen pro Replikationsaufgabe. Als Faustregel empfehlen wir jedoch, die Anzahl der Tabellen in einer Aufgabe auf weniger als 60 000 zu begrenzen. Die Ressourcennutzung wird oft zum Engpass, wenn eine einzige Aufgabe mehr als 60.000 Tabellen verwendet.

Aufgaben schlagen fehl, wenn ein Primärschlüssel in der LOB-Spalte erstellt wird

Im Modus FULL LOB oder LIMITED LOB unterstützt AWS DMS keine Replikation von Primärschlüsseln des Datentyps LOB.

DMS migriert zunächst eine Zeile mit einer LOB-Spalte als Null und aktualisiert später die LOB-Spalte. Wenn der Primärschlüssel in einer LOB-Spalte erstellt wird, schlägt daher die anfängliche Einfügung fehl, da der Primärschlüssel nicht null sein darf. Fügen Sie zur Umgehung des Problems eine weitere Spalte als Primärschlüssel hinzu und entfernen Sie den Primärschlüssel aus der LOB-Spalte.

Doppelte Datensätze in einer Zieltabelle ohne Primärschlüssel

Das Ausführen einer Volllast- und CDC-Aufgabe kann zu doppelten Datensätzen in Zieltabellen führen, die keinen Primärschlüssel oder eindeutigen Index haben. Um das Duplizieren von Datensätzen in Zieltabellen bei Volllast- und CDC-Aufgaben zu vermeiden, stellen Sie sicher, dass die Zieltabellen über einen Primärschlüssel oder einen eindeutigen Index verfügen.

Quellendpunkte fallen in den reservierten IP-Bereich

Wenn eine AWS DMS-Quelldatenbank eine IP-Adresse im reservierten IP-Bereich 192.168.0.0/24 verwendet, schlägt der Test der Quellendpunktverbindung fehl. Die folgenden Schritte stellen eine mögliche Umgehung des Problems dar:

1. Suchen Sie eine Amazon-EC2-Instance, die sich nicht im reservierten Bereich befindet und unter 192.168.0.0/24 mit der Quelldatenbank kommunizieren kann.
2. Installieren Sie einen Socat-Proxy und führen Sie ihn aus. Es folgt ein Beispiel.

```
yum install socat
```

```
socat -d -d -llocal2 tcp4-listen:database_port,bind=0.0.0.0,reuseaddr,fork
tcp4:source_database_ip_address:database_port
&
```

Verwenden Sie die IP-Adresse der Amazon-EC2-Instance und den oben angegebenen Datenbank-Port für den AWS DMS-Endpunkt. Stellen Sie sicher, dass der Endpunkt über die Sicherheitsgruppe verfügt, mit der AWS DMS auf den Datenbank-Port zugreifen kann. Beachten Sie, dass der Proxy für die Dauer der Ausführung Ihrer DMS-Aufgabe aktiv sein muss. Je nach Anwendungsfall müssen Sie möglicherweise das Proxy-Setup automatisieren.

Zeitstempel sind in Amazon-Athena-Abfragen unleserlich

Wenn Zeitstempel in Athena-Abfragen verschleiert werden, verwenden Sie die [ModifyEndpoint](#) Aktion AWS Management Console oder , um den `parquetTimestampInMillisecond` Wert für Ihren Amazon S3-Endpunkt auf `festzulegen true`. Weitere Informationen finden Sie unter [S3Settings](#).

Fehlersuche bei Verwendung von Oracle

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit Oracle-Datenbanken.

Themen

- [Abrufen von Daten aus Ansichten](#)
- [Migrieren von LOBs aus Oracle 12c](#)
- [Wechseln zwischen Oracle LogMiner und Binary Reader](#)
- [Fehler: Oracle CDC angehalten 122301 Oracle CDC maximale Wiederholversuche überschritten.](#)
- [Automatisches Hinzufügen der zusätzlichen Protokollierung zu einem Oracle-Quellendpunkt](#)
- [LOB-Änderungen werden nicht erfasst](#)
- [Fehler: ORA-12899: Wert zu groß für Spalte column-name](#)
- [Datentyp NUMBER wird nicht richtig interpretiert](#)
- [Datensätze fehlen bei Volllast](#)
- [Table Error](#)
- [Fehler: Cannot retrieve Oracle archived Redo log destination ids](#)

- [Bewertung der Leseleistung von Oracle-Redo- oder -Archivprotokollen](#)

Abrufen von Daten aus Ansichten

Sie können Daten aus einer Ansicht einmal abrufen; Sie können sie nicht für die laufende Replikation verwenden. Um Daten aus Ansichten extrahieren zu können, müssen Sie den folgenden Code im Abschnitt Endpunkteinstellungen auf der Seite zum Oracle-Quellendpunkt hinzufügen. Wenn Sie Daten aus einer Ansicht extrahieren, wird die Ansicht im Zielschema als Tabelle angezeigt.

```
"ExposeViews": true
```

Migrieren von LOBs aus Oracle 12c

AWS DMS kann zwei Methoden verwenden, um Änderungen an einer Oracle-Datenbank zu erfassen: Binary Reader und Oracle LogMiner. Standardmäßig AWS DMS verwendet Oracle LogMiner um Änderungen zu erfassen. Unter Oracle 12c unterstützt Oracle LogMiner jedoch keine LOB-Spalten. Um Änderungen an LOB-Spalten unter Oracle 12c zu erfassen, verwenden Sie Binary Reader.

Wechseln zwischen Oracle LogMiner und Binary Reader

AWS DMS kann zwei Methoden verwenden, um Änderungen an einer Oracle-Quelldatenbank zu erfassen: Binary Reader und Oracle LogMiner. Oracle LogMiner ist die Standardeinstellung. Um Binary Reader zur Erfassung von Änderungen verwenden, führen Sie die folgenden Schritte aus:

So verwenden Sie Binary Reader für die Erfassung von Änderungen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Endpunkte aus.
3. Wählen Sie den Oracle-Quellendpunkt, für den Sie Binary Reader verwenden möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie Erweitert aus und fügen Sie dann den folgenden Code unter Extra Verbindungsattribute hinzu:

```
useLogminerReader=N
```

6. Verwenden Sie ein Oracle-Entwickler-Tool wie z. B. SQL-Plus, um dem AWS DMS-Benutzerkonto, über das die Verbindung zum Oracle-Endpunkt hergestellt wird, die folgenden zusätzlichen Berechtigungen zu erteilen.

```
SELECT ON V_$TRANSPORTABLE_PLATFORM
```

Fehler: Oracle CDC angehalten 122301 Oracle CDC maximale Wiederholversuche überschritten.

Dieser Fehler tritt auf, wenn die erforderlichen Oracle-Archivprotokolle von Ihrem Server entfernt wurden, bevor AWS DMS anhand dieser Protokolle Änderungen erfassen konnte. Erhöhen Sie die Richtlinien für die Aufbewahrung von Protokollen auf Ihrem Datenbankserver. Führen Sie für eine Amazon RDS-Datenbank die folgenden Schritte aus, um die Aufbewahrungsfrist für Protokolle zu erhöhen. Im folgenden Beispiel wird durch den Code die Aufbewahrung für Protokolle für eine Amazon RDS-DB-Instance auf 24 Stunden erhöht.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
```

Automatisches Hinzufügen der zusätzlichen Protokollierung zu einem Oracle-Quellendpunkt

Standardmäßig ist in AWS DMS die zusätzliche Protokollierung deaktiviert. Um die zusätzliche Protokollierung für einen Oracle-Quellendpunkt automatisch zu aktivieren, führen Sie die folgenden Schritte aus:

So fügen Sie einem Oracle-Quellendpunkt die zusätzliche Protokollierung hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Endpunkte aus.

3. Wählen Sie den Oracle-Quellendpunkt, dem Sie die zusätzliche Protokollierung hinzufügen möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie Erweitert aus und fügen Sie dann den folgenden Code im Textfeld Extra Verbindungsattribute hinzu:

```
addSupplementalLogging=Y
```

6. Wählen Sie Ändern aus.

LOB-Änderungen werden nicht erfasst

Derzeit muss eine Tabelle über einen Primärschlüssel für AWS DMS verfügen, um LOB-Änderungen erfassen zu können. Wenn eine Tabelle mit LOBs über keinen Primärschlüssel verfügt, können Sie mehrere Aktionen zur Erfassung von LOB-Änderungen durchführen:

- Fügen Sie der Tabelle einen Primärschlüssel hinzu. Dazu fügen Sie einfach eine ID-Spalte hinzu und füllen diese mit einer Sequenz unter Verwendung eines Auslösers.
- Erstellen Sie eine materialisierte Ansicht der Tabelle, die eine vom System generierte ID als Primärschlüssel enthält, und migrieren Sie dann die materialisierte Ansicht anstatt der Tabelle.
- Erstellen Sie einen logische Standby, fügen Sie der Tabelle einen Primärschlüssel hinzu, und migrieren Sie vom logischen Standby.

Fehler: ORA-12899: Wert zu groß für Spalte *column-name*

Der Fehler „ORA-12899: Wert zu groß für Spalte *column-name*“ wird häufig durch verschiedene Probleme verursacht.

Eins besteht darin, dass die von der Quell- und Zieldatenbank verwendeten Zeichensätze nicht übereinstimmen.

Ein anderes besteht darin, dass sich die National Language Support (NLS)-Einstellungen zwischen den beiden Datenbanken unterscheiden. Ein häufiger Grund für diesen Fehler ist, wenn der NLS_LENGTH_SEMANTICS-Parameter der Quelldatenbank auf CHAR und der NLS_LENGTH_SEMANTICS-Parameter der Zieldatenbank auf BYTE festgelegt sind.

Datentyp NUMBER wird nicht richtig interpretiert

Der Oracle-Datentyp NUMBER wird in verschiedene AWS DMS-Datentypen konvertiert, je nach Genauigkeit und Skalierung von NUMBER. Diese Konvertierungen sind hier dokumentiert: [Quelldatentypen für Oracle](#). Die Art und Weise, wie der Datentyp NUMBER konvertiert wird, kann auch von der Verwendung von Endpunkteinstellungen für den Oracle-Quellendpunkt abhängen. Diese Endpunkteinstellungen sind unter [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#) dokumentiert.

Datensätze fehlen bei Volllast

Beim Ausführen von Volllastaufgaben sucht AWS DMS nach offenen Transaktionen auf Datenbankebene und wartet, bis die Transaktion bestätigt wird. Basierend auf der Aufgabeneinstellung `TransactionConsistencyTimeout=600` wartet AWS DMS beispielsweise 10 Minuten, auch wenn sich die offene Transaktion auf eine Tabelle bezieht, die nicht in der Tabellenzuordnung enthalten ist. Wenn sich die offene Transaktion jedoch auf eine Tabelle bezieht, die in der Tabellenzuordnung enthalten ist, und die Transaktion nicht rechtzeitig bestätigt wird, führt dies zu fehlenden Datensätzen in der Zieltabelle.

Sie können die Aufgabeneinstellung `TransactionConsistencyTimeout` ändern und die Wartezeit erhöhen, wenn Sie wissen, dass das Bestätigen offener Transaktionen länger dauert.

Beachten Sie außerdem, dass der Standardwert für die Aufgabeneinstellung `FailOnTransactionConsistencyBreached` `false` lautet. Das bedeutet, dass AWS DMS weiterhin andere Transaktionen anwendet, offene Transaktionen jedoch ausgelassen werden. Falls Sie möchten, dass die Aufgabe fehlschlägt, wenn offene Transaktionen nicht rechtzeitig abgeschlossen werden, können Sie `FailOnTransactionConsistencyBreached` auf `true` setzen.

Table Error

`Table Error` wird während der Replikation in Tabellenstatistiken angezeigt, wenn eine `WHERE`-Klausel nicht auf eine Primärschlüsselspalte verweist und die zusätzliche Protokollierung nicht für alle Spalten verwendet wird.

Um dieses Problem zu beheben, aktivieren Sie die zusätzliche Protokollierung für alle Spalten der Tabelle, auf die verwiesen wird. Weitere Informationen finden Sie unter [Einrichten der zusätzlichen Protokollierung](#).

Fehler: Cannot retrieve Oracle archived Redo log destination ids

Dieser Fehler tritt auf, wenn für Ihre Oracle-Quelle keine Archivprotokolle generiert wurden oder wenn V\$ARCHIVED_LOG leer ist. Sie können den Fehler beheben, indem Sie manuell zwischen den Protokollen wechseln.

Führen Sie für eine Amazon-RDS-Datenbank die folgenden Schritte aus, um die Protokolldateien zu wechseln. Die Prozedur `switch_logfile` hat keine Parameter.

```
exec rdsadmin.rdsadmin_util.switch_logfile;
```

Verwenden Sie für eine selbstverwaltete Oracle-Quelldatenbank den folgenden Befehl, um einen Protokollwechsel zu erzwingen.

```
ALTER SYSTEM SWITCH LOGFILE ;
```

Bewertung der Leseleistung von Oracle-Redo- oder -Archivprotokollen

Wenn Sie Leistungsprobleme bei Ihrer Oracle-Quelle feststellen, können Sie die Leseleistung Ihrer Oracle-Redo- oder -Archivprotokolle bewerten, um Möglichkeiten zur Leistungssteigerung zu finden. Verwenden Sie das [Diagnose-Amazon-Machine-Image \(AMI\) von AWS DMS](#), um die Leseleistung des Redo- oder -Archivprotokolls zu testen.

Sie können das Diagnose-AMI von AWS DMS für folgende Zwecke verwenden:

- Verwenden Sie die bFile-Methode, um die Leistung von Redo-Protokolldateien zu bewerten.
- Verwenden Sie die - LogMiner Methode, um die Leistung der Redo-Protokolldatei zu bewerten.
- Verwenden Sie die PL/SQL-Methode (`dbms_lob.read`), um die Leistung von Redo-Protokolldateien zu bewerten.
- Verwenden Sie Single-Thread, um die Leseleistung von ASMFile zu bewerten.
- Verwenden Sie Multi-Threads, um die Leseleistung von ASMFile zu bewerten.
- Verwenden Sie die Direct-OS-Windows-Funktion `Readfile()` oder die Linux-Funktion `Pread64`, um die Redo-Protokolldatei zu bewerten.

Anschließend können Sie auf der Grundlage der Ergebnisse Korrekturmaßnahmen ergreifen.

So testen Sie die Leseleistung von Oracle-Redo- oder -Archivprotokolldateien

1. Erstellen Sie eine Amazon-EC2-Instance mit AWS DMS-Diagnose-AMI und stellen Sie eine Verbindung damit her.

Weitere Informationen finden Sie unter [Working with the AWS DMS diagnostic AMI](#).

2. Führen Sie den Befehl `awsreplperf` aus.

```
$ awsreplperf
```

Der Befehl zeigt die Optionen des AWS DMS-Dienstprogramms zur Oracle-Leseleistung an.

```
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. Wählen Sie eine Option in der Liste aus.
4. Geben Sie die folgenden Informationen zu Datenbankverbindung und Archivprotokoll ein.

```
Oracle user name [system]:
Oracle password:

Oracle connection name [orcllx]:
Connection format hostname:port/instance

Oracle event trace? [N]:
Default N = No or Y = Yes

Path to redo or archive log file []:
```

5. Untersuchen Sie die angezeigte Ausgabe auf relevante Informationen zur Leseleistung. Das folgende Beispiel zeigt eine Ausgabe, die sich aus der Auswahl von Optionsnummer 2, Lesen mit LogMiner ergeben kann.


```

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
1198000
Elapsed Time : 7044.83973 sec
Read speed in : 0.088575 MB/sec
LogMinerRead: counted 1198389 redo log rows, total undo / redo size : 655073562

```

6. Geben Sie 0 (Null) ein, um das Dienstprogramm zu beenden.

Nächste Schritte

- Wenn die Ergebnisse zeigen, dass die Lesegeschwindigkeit unter einem akzeptablen Schwellenwert liegt, führen Sie das [Oracle Diagnostic Support Script](#) auf dem Endpunkt aus und überprüfen Sie die Abschnitte zu Wartezeit, Lastprofil und E/A-Profil. Passen Sie dann alle ungewöhnlichen Konfigurationen an, die die Leseleistung verbessern könnten. Wenn Ihre Redo-Protokolldateien beispielsweise bis zu 2 GB groß sind, versuchen Sie, LOG_BUFFER auf 200 MB zu erhöhen, um die Leistung zu verbessern.
- Sehen Sie sich [AWS DMS Best Practices](#) an, um sicherzustellen, dass Ihre Replikations-Instance, Aufgabe und Endpunkte in DMS optimal konfiguriert sind.

Fehlersuche bei Verwendung von MySQL

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit MySQL-Datenbanken.

Themen

- [CDC-Aufgabe schlägt für Amazon RDS-DB-Instance-Endpunkt aufgrund deaktivierter binärer Protokollierung fehl](#)
- [Verbindungen mit einer MySQL-Ziel-Instance werden während einer Aufgabe getrennt](#)
- [Hinzufügen von Autocommit zu einem MySQL-kompatiblen Endpunkt](#)
- [Deaktivieren von Fremdschlüsseln auf einem MySQL-kompatiblen Zielendpunkt](#)
- [Durch Fragezeichen ersetzte Zeichen](#)
- [„Bad event“-Protokolleinträge](#)

- [Change Data Capture \(CDC\) mit MySQL 5.5](#)
- [Erhöhen der Aufbewahrungszeit für binäre Protokolle für Amazon RDS-DB-Instances](#)
- [Protokollmeldung: Einige Änderungen von der Quelldatenbank hatten bei Anwendung auf die Zieldatenbank keine Auswirkungen.](#)
- [Fehler: Bezeichner zu lang](#)
- [Fehler: Felddatenumwandlung schlägt aufgrund nicht unterstützten Zeichensatzes fehl](#)
- [Fehler: Codeseite 1252 zu UTF8 \[120112\] eine Felddatenkonvertierung ist fehlgeschlagen](#)
- [Indizes, Fremdschlüssel oder kaskadierende Aktualisierungen oder Löschungen wurden nicht migriert](#)

CDC-Aufgabe schlägt für Amazon RDS-DB-Instance-Endpoint aufgrund deaktivierter binärer Protokollierung fehl

Dieses Problem tritt bei Amazon RDS-DB-Instances auf, wenn automatische Sicherungen deaktiviert sind. Aktivieren Sie automatische Sicherungen, indem Sie den Aufbewahrungszeitraum für Backups auf einen Wert größer als null festlegen.

Verbindungen mit einer MySQL-Ziel-Instance werden während einer Aufgabe getrennt

Wenn eine Aufgabe mit LOBs von einem MySQL-Ziel getrennt wird, werden möglicherweise die folgenden Arten von Fehlern im Aufgabenprotokoll angezeigt.

```
[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.
```

In diesem Fall müssen Sie möglicherweise einige Aufgabeneinstellungen anpassen.

Um das Problem einer Aufgabe zu lösen, die von einem MySQL-Ziel getrennt wird, führen Sie die folgenden Schritte aus:

- Stellen Sie sicher, dass Ihre Datenbankvariable `max_allowed_packet` auf einen ausreichend großen Wert festgelegt ist, um Ihre größten LOBs zu speichern.
- Stellen Sie sicher, dass die folgenden Variablen auf einen hohen Zeitüberschreitungswert festgelegt sind. Wir empfehlen, einen Wert von mindestens 5 Minuten für jede dieser Variablen zu verwenden.
 - `net_read_timeout`
 - `net_write_timeout`
 - `wait_timeout`

Informationen zum Festlegen von MySQL-Systemvariablen finden Sie unter [Server System Variables](#) in der [MySQL-Dokumentation](#).

Hinzufügen von Autocommit zu einem MySQL-kompatiblen Endpunkt

So fügen Sie Autocommit zu einem MySQL-kompatiblen Zielendpunkt hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Endpunkte aus.
3. Wählen Sie den MySQL-kompatiblen Zielendpunkt aus, dem Sie Autocommit hinzufügen möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie Erweitert aus und fügen Sie dann den folgenden Code im Textfeld Extra Verbindungsattribute hinzu:

```
Initstmt= SET AUTOCOMMIT=1
```

6. Wählen Sie Ändern aus.

Deaktivieren von Fremdschlüsseln auf einem MySQL-kompatiblen Zielendpunkt

Sie können Fremdschlüsselprüfungen für MySQL deaktivieren, indem Sie Extra Verbindungsattribute im Abschnitt Erweitert des Zielendpunkts von MySQL, MariaDB oder der Amazon-Aurora-MySQL-kompatiblen Edition hinzufügen.

So deaktivieren Sie Fremdschlüssel auf einem MySQL-kompatiblen Zielendpunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS DMSKonsole unter <https://console.aws.amazon.com/dms/v2/>.
2. Wählen Sie Endpunkte aus.
3. Wählen Sie den MySQL-, Aurora-MySQL- oder MariaDB-Zielendpunkt aus, für den Sie Fremdschlüssel deaktivieren möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie Erweitert aus und fügen Sie dann den folgenden Code im Textfeld Extra Verbindungsattribute hinzu:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0
```

6. Wählen Sie Ändern aus.

Durch Fragezeichen ersetzte Zeichen

Dieses Problem tritt am häufigsten auf, wenn die Zeichen des Quellendpunkts von einem Zeichensatz kodiert wurden, der nicht von AWS DMS unterstützt wird.

„Bad event“-Protokolleinträge

„Bad event“-Einträge in den Migrationsprotokollen weisen in der Regel darauf hin, dass auf dem Endpunkt der Quelldatenbank ein nicht unterstützter Data Definition Language (DDL)-Vorgang versucht wurde. Nicht unterstützte DDL-Vorgänge führen zu einem Ereignis, das die Replikations-Instance nicht überspringen kann, sodass ein fehlerhaftes Ereignis protokolliert wird.

Starten Sie die Aufgabe von Anfang an neu, um dieses Problem zu beheben. Dadurch werden die Tabellen neu geladen und die Änderungen werden ab einem Zeitpunkt erfasst, nachdem der nicht unterstützte DDL-Vorgang ausgeführt wurde.

Change Data Capture (CDC) mit MySQL 5.5

Change Data Capture (CDC) von AWS DMS für MySQL-kompatible Amazon-RDS-Datenbanken erfordert eine zeilenbasierte binäre Protokollierung mit vollständigem Abbild. Dies wird in MySQL Version 5.5 oder früher nicht unterstützt. Um CDC von AWS DMS zu verwenden, müssen Sie Ihre Amazon RDS-DB-Instance auf MySQL-Version 5.6 aktualisieren.

Erhöhen der Aufbewahrungszeit für binäre Protokolle für Amazon RDS-DB-Instances

AWS DMS erfordert die Aufbewahrung von binären Protokolldateien für die Erfassung geänderter Daten. Um die Aufbewahrungsfrist für Protokolle für eine Amazon RDS-DB-Instance zu erhöhen, gehen Sie wie folgt vor. Beim folgenden Beispiel wird die Aufbewahrungszeit für binäre Protokolle auf 24 Stunden erhöht.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Protokollmeldung: Einige Änderungen von der Quelldatenbank hatten bei Anwendung auf die Zieldatenbank keine Auswirkungen.

Wenn AWS DMS den Wert einer MySQL-Datenbankspalte auf seinen vorhandenen Wert aktualisiert, wird von MySQL die Meldung `zero rows affected` zurückgegeben. Dieses Verhalten unterscheidet sich von anderen Datenbank-Engines wie Oracle und SQL Server. Diese Engines aktualisieren eine Zeile, auch wenn der ersetzende Wert mit dem aktuellen identisch ist.

Fehler: Bezeichner zu lang

Der folgende Fehler tritt auf, wenn ein Bezeichner zu lang ist:

```
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)
```

In Einzelfällen legen Sie fest, dass AWS DMS die Tabellen und Primärschlüssel in der Zieldatenbank erstellen soll. In solchen Fällen verwendet DMS derzeit nicht die gleichen Namen für die Primärschlüssel, die in der Quelldatenbank verwendet wurden. Stattdessen erstellt DMS die Primärschlüsselnamen basierend auf den Tabellennamen. Bei langen Tabellennamen kann es vorkommen, dass der automatisch generierte Bezeichner die zulässigen Grenzwerte für MySQL überschreitet.

Der aktuelle Ansatz zum Beheben dieses Problems besteht darin, zunächst die Tabellen und Primärschlüssel in der Zieldatenbank vorab zu erstellen. Verwenden Sie dann eine Aufgabe, bei der die Aufgabeneinstellung Zieltabellen-Vorbereitungsmodus auf Nichts unternehmen oder Kürzen gesetzt ist, um die Zieltabellen zu befüllen.

Fehler: Felddatenumwandlung schlägt aufgrund nicht unterstützten Zeichensatzes fehl

Der folgende Fehler tritt auf, wenn ein nicht unterstützter Zeichensatz dazu führt, dass eine Felddatenumwandlung fehlschlägt:

```
"[SOURCE_CAPTURE ]E: Column 'column-name' uses an unsupported character set [120112]
A field data conversion failed. (mysql_endpoint_capture.c:2154)
```

Überprüfen Sie die Parameter Ihrer Datenbank bezüglich Verbindungen. Mit dem folgenden Befehl können Sie diese Parameter festlegen:

```
SHOW VARIABLES LIKE '%char%';
```

Fehler: Codeseite 1252 zu UTF8 [120112] eine Felddatenkonvertierung ist fehlgeschlagen

Die folgenden Fehler kann während einer Migration auftreten, wenn Sie andere als Codepage-1252-Zeichen in der MySQL-Quelldatenbank haben.

```
[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)
```

Als Abhilfe können Sie das zusätzliche Verbindungsattribut `CharsetMapping` mit Ihrem MySQL-Quellendpunkt verwenden, um die Zeichensatzzuordnung festzulegen. Möglicherweise müssen Sie die AWS DMS-Migrationsaufgabe von Anfang an neu starten, wenn Sie diese Endpunkteinstellung hinzufügen.

Die folgende Endpunkteinstellung könnte beispielsweise für einen MySQL-Quellendpunkt verwendet werden, bei dem der Quellzeichensatz `Utf8` oder `latin1` ist. `65001` ist die UTF8-Codepage-ID.

```
CharsetMapping=utf8,65001  
CharsetMapping=latin1,65001
```

Indizes, Fremdschlüssel oder kaskadierende Aktualisierungen oder Löschungen wurden nicht migriert

AWS DMS unterstützt die Migration von sekundären Objekten wie Indizes und Fremdschlüsseln nicht. Zum Replizieren von Änderungen, die durch eine kaskadierende Aktualisierung oder Löschung an untergeordneten Tabellen vorgenommen wurden, muss die auslösende Fremdschlüsseleinschränkung für die Zieltabelle aktiv sein. Erstellen Sie den Fremdschlüssel manuell in der Zieltabelle, um diese Einschränkung zu umgehen. Erstellen Sie dann entweder eine einzelne Aufgabe für Volllast und CDC oder zwei separate Aufgaben für Volllast und CDC, wie im Folgenden beschrieben:

Eine einzelne Aufgabe erstellen, die Volllast und CDC unterstützt

In diesem Verfahren wird beschrieben, wie Fremdschlüssel und Indizes mithilfe einer einzelnen Aufgabe für Volllast und CDC migriert werden.

Eine Aufgabe für Volllast und CDC erstellen

1. Erstellen Sie manuell die Tabellen mit Fremdschlüsseln und Indizes auf dem Ziel, die mit den Quelltabellen übereinstimmen.
2. Fügen Sie dem AWS DMS-Zielendpunkt das folgende ECA hinzu:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

- Erstellen Sie die AWS DMS-Aufgabe mit der Einstellung `DO_NOTHING` für `TargetTablePrepMode`.
- Setzen Sie die Einstellung `Stop task after full load completes` auf `StopTaskCachedChangesApplied`.
- Starten Sie die Aufgabe. AWS DMS beendet die Aufgabe nach Abschluss der Volllast automatisch und wendet alle zwischengespeicherten Änderungen an.
- Entfernen Sie das zuvor hinzugefügte ECA `SET FOREIGN_KEY_CHECKS`.
- Setzen Sie die Aufgabe fort. Die Aufgabe geht in die CDC-Phase über und wendet die laufenden Änderungen aus der Quelldatenbank auf das Ziel an.

Aufgaben für Volllast und CDC separat erstellen

In diesen Verfahren wird beschrieben, wie Fremdschlüssel und Indizes mithilfe separater Aufgaben für Volllast und CDC migriert werden.

Eine Aufgabe für Volllast erstellen

- Erstellen Sie manuell die Tabellen mit Fremdschlüsseln und Indizes auf dem Ziel, die mit den Quelltabellen übereinstimmen.
- Fügen Sie dem AWS DMS-Zielendpunkt das folgende ECA hinzu:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

- Erstellen Sie die AWS DMS-Aufgabe mit der Einstellung `DO_NOTHING` für den Parameter `TargetTablePrepMode` und `FALSE` für `EnableValidation`.
- Starten Sie die Aufgabe. AWS DMS beendet die Aufgabe nach Abschluss der Volllast automatisch und wendet alle zwischengespeicherten Änderungen an.
- Notieren Sie sich nach Aufgabenabschluss die Startzeit der Aufgabe für Volllast in UTC oder den Namen und die Position der Binärprotokolldatei, um die Nur-CDC-Aufgabe zu starten. In den Protokollen finden Sie den Zeitstempel in UTC ab der ersten Startzeit der Volllast.

Eine Nur-CDC-Aufgabe erstellen

- Entfernen Sie das zuvor festgelegte ECA `SET FOREIGN_KEY_CHECKS`.

- Erstellen Sie die Nur-CDC-Aufgabe und legen Sie die Startposition auf die im vorherigen Schritt notierte Startzeit der Aufgabe für Volllast fest. Alternativ können Sie auch die im vorigen Schritt notierte Position der Binärprotokolldatei verwenden. Setzen Sie die Einstellung `TargetTablePrepMode` auf `DO_NOTHING`. Aktivieren Sie die Datenvalidierung, indem Sie die Einstellung `EnableValidation` auf `TRUE` setzen, wenn erforderlich.
- Starten Sie die Nur-CDC-Aufgabe und überwachen Sie die Protokolle auf Fehler.

Note

Diese Problemumgehung gilt nur für Migrationen von MySQL zu MySQL. Sie können diese Methode nicht mit dem Feature für Stapelanwendungen verwenden, da dieses voraussetzt, dass Zieltabellen keine aktiven Fremdschlüssel haben.

Fehlersuche bei Verwendung von PostgreSQL

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit PostgreSQL-Datenbanken.

Themen

- [Verkürzte JSON-Datentypen](#)
- [Spalten eines benutzerdefinierten Datentyps werden nicht korrekt migriert](#)
- [Fehler: Kein Schema zum Erstellen ausgewählt](#)
- [Lösch- und Aktualisierungsvorgänge für eine Tabelle werden bei Verwendung von CDC nicht repliziert](#)
- [Truncate-Anweisungen werden nicht ordnungsgemäß propagiert](#)
- [Verhindern, dass PostgreSQL DDL erfasst](#)
- [Auswahl des Schemas, in dem Datenbankobjekte für die DDL-Erfassung erstellt werden](#)
- [Oracle-Tabellen fehlen nach Migration zu PostgreSQL](#)
- [ReplicationSlotDiskUsage erhöht und restart_Isn stoppt bei langen Transaktionen wie ETL-Workloads](#)
- [Für Aufgabe, die Ansicht als Quelle verwendet, wurden keine Zeilen kopiert](#)

Verkürzte JSON-Datentypen

AWS DMS verarbeitet den JSON-Datentyp in PostgreSQL als LOB-Datentypspalte. Das bedeutet, dass die LOB-Größenbeschränkung für JSON-Daten gilt, wenn Sie den beschränkten LOB-Modus verwenden.

Nehmen wir beispielsweise an, dass der beschränkte LOB-Modus auf 4 096 KB eingestellt ist. In diesem Fall werden alle JSON-Daten, die größer als 4 096 KB sind, bei der Begrenzung von 4 096 KB gekürzt und der Validierungstest in PostgreSQL schlägt fehl.

Die folgenden Protokollinformationen zeigen einen JSON-Code, der aufgrund der Einstellungen des beschränkten LOB-Modus und fehlgeschlagener Validierung gekürzt wurde.

```
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)
```

Spalten eines benutzerdefinierten Datentyps werden nicht korrekt migriert

Bei der Replikation von einer PostgreSQL-Quelle erstellt AWS DMS die Zieltabelle mit denselben Datentypen für alle Spalten, mit Ausnahme der Spalten mit benutzerdefinierten Datentypen. In solchen Fällen wird der Datentyp als "character varying" im Ziel erstellt.

Fehler: Kein Schema zum Erstellen ausgewählt

In einigen Fällen wird möglicherweise der Fehler „SQL_ERROR SqlState: 3F000 NativeError: 7 Message: ERROR: no schema has selected to create in“ angezeigt.

Dieser Fehler kann auftreten, wenn Ihre JSON-Tabellenzuordnung einen Platzhalterwert für das Schema enthält, die Quelldatenbank diesen Wert jedoch nicht unterstützt.

Lösch- und Aktualisierungsvorgänge für eine Tabelle werden bei Verwendung von CDC nicht repliziert

Lösch- und Aktualisierungsvorgänge während der Erfassung von Datenänderungen (Change Data Capture, CDC) werden ignoriert, wenn die Quelltablelle keinen Primärschlüssel aufweist. AWS DMS unterstützt die CDC für PostgreSQL-Tabellen mit Primärschlüsseln.

Wenn eine Tabelle keinen Primärschlüssel aufweist, enthalten die Write-Ahead (WAL)-Protokolle kein Vorher-Image der Datenbankzeile. In diesem Fall kann AWS DMS die Tabelle nicht aktualisieren. Erstellen Sie einen Primärschlüssel für die Quelltablelle, damit Löschvorgänge repliziert werden.

Truncate-Anweisungen werden nicht ordnungsgemäß propagiert

Bei der Verwendung von CDC werden TRUNCATE-Operationen von AWS DMS nicht unterstützt.

Verhindern, dass PostgreSQL DDL erfasst

Sie können verhindern, dass ein PostgreSQL-Zielendpunkt DDL-Anweisungen erfasst, indem Sie die folgende Endpunkteinstellung-Anweisung hinzufügen.

```
"CaptureDDLs": "N"
```

Auswahl des Schemas, in dem Datenbankobjekte für die DDL-Erfassung erstellt werden

Sie können steuern, in welchem Schema die Datenbankobjekte im Zusammenhang mit der Erfassung von DDL erstellt werden. Fügen Sie die folgende Endpunkteinstellung-Anweisung hinzu. Der Parameter Endpunkteinstellung ist auf der Registerkarte des Quellendpunkts verfügbar.

```
"DdlArtifactsSchema: "xyzddlschema"
```

Oracle-Tabellen fehlen nach Migration zu PostgreSQL

In diesem Fall sind Ihre Tabellen und Daten im Allgemeinen weiterhin verfügbar.

Bei Oracle werden Tabellennamen standardmäßig in Großbuchstaben, bei PostgreSQL in Kleinbuchstaben geschrieben. Für Migrationen von Oracle zu PostgreSQL empfehlen wir, bestimmte Transformationsregeln im Abschnitt zur Tabellenzuordnung Ihrer Aufgabe anzugeben. Mit diesen Transformationsregeln wird die Groß- und Kleinschreibung Ihrer Tabellennamen umgewandelt.

Falls Sie Ihre Tabellen ohne Transformationsregeln für die Umwandlung der Groß-/Kleinschreibung der Tabellennamen migriert haben, setzen Sie die Tabellennamen in Anführungszeichen, wenn Sie darauf verweisen.

ReplicationSlotDiskUsage erhöht und restart_lsn stoppt bei langen Transaktionen wie ETL-Workloads

Wenn die logische Replikation aktiviert ist, beträgt die maximale Anzahl von Änderungen, die pro Transaktion im Arbeitsspeicher gespeichert werden, 4 MB. Danach werden die Änderungen auf den Datenträger übertragen. Daher nimmt der Wert für `ReplicationSlotDiskUsage` zu und `restart_lsn` schreitet erst voran, wenn die Transaktion abgeschlossen/abgebrochen und das Rollback beendet ist. Da es sich um eine lange Transaktion handelt, kann das Rollback lange dauern.

Vermeiden Sie daher lang andauernde Transaktionen, wenn die logische Replikation aktiviert ist. Versuchen Sie stattdessen, die Transaktion in mehrere kleinere Transaktionen aufzuteilen.

Für Aufgabe, die Ansicht als Quelle verwendet, wurden keine Zeilen kopiert

Setzen Sie zum Migrieren einer Ansicht `table`-type auf `all` oder `view`. Weitere Informationen finden Sie unter [Festlegen der Tabellenauswahl und der Transformationsregeln über die Konsole](#).

Zu den Quellen, die Ansichten unterstützen, gehören folgende.

- Oracle
- Microsoft SQL Server

- MySQL
- PostgreSQL
- IBM Db2 (LUW)
- SAP Adaptive Server Enterprise (ASE)

Fehlersuche bei Verwendung von Microsoft SQL Server

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit Datenbanken von Microsoft SQL Server.

Themen

- [Fehler bei Erfassung von Änderungen für SQL Server-Datenbank](#)
- [Fehlende Identitätsspalten](#)
- [Fehler: SQL Server unterstützt keine Publikationen](#)
- [Änderungen werden im Ziel nicht angezeigt](#)
- [Uneinheitliche Tabelle, die über Partitionen hinweg zugeordnet ist](#)

Fehler bei Erfassung von Änderungen für SQL Server-Datenbank

Fehler während CDC weisen häufig darauf hin, dass eine der Voraussetzungen nicht erfüllt war. Zum Beispiel ist eine der am häufigsten übersehenen Voraussetzungen eine vollständige Datenbanksicherung. Das Aufgabenprotokoll gibt dieses Versäumnis mit dem folgenden Fehler an:

```
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)
```

Überprüfen Sie die Voraussetzungen für die Verwendung von SQL Server als Quelle unter [Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS](#).

Fehlende Identitätsspalten

AWS DMS unterstützt bei der Erstellung eines Zielschemas keine Identitätsspalten. Sie müssen sie hinzufügen, nachdem der erste Ladevorgang abgeschlossen ist.

Fehler: SQL Server unterstützt keine Publikationen

Die folgende Fehlermeldung wird angezeigt, wenn Sie SQL Server Express als Quellendpunkt verwenden:

```
RetCode: SQL_ERROR SqlState: HY000 NativeError: 21106  
Message: This edition of SQL Server does not support publications.
```

AWS DMS unterstützt derzeit nicht SQL Server Express als Quelle oder Ziel.

Änderungen werden im Ziel nicht angezeigt

AWS DMS erfordert, dass eine SQL Server-Quelldatenbank entweder im Datenwiederherstellungsmodell "FULL" oder "BULK LOGGED" operiert, um Änderungen konsistent erfassen zu können. Das Modell „SIMPLE“ wird nicht unterstützt.

Beim Wiederherstellungsmodell SIMPLE werden minimale Informationen protokolliert, die erforderlich sind, damit Benutzer ihre Datenbank wiederherstellen können. Alle inaktiven Protokolleinträge werden automatisch abgeschnitten, wenn ein Prüfpunkt eintritt.

Alle Operationen werden weiterhin protokolliert. Sobald jedoch ein Prüfpunkt eintritt, wird das Protokoll automatisch gekürzt. Diese Kürzung bedeutet, dass das Protokoll wiederverwendet werden und ältere Protokolleinträge überschrieben werden können. Wenn Protokolleinträge überschrieben werden, können Änderungen nicht erfasst werden. Dieses Problem ist der Grund dafür, dass AWS DMS das Datenwiederherstellungsmodell SIMPLE nicht unterstützt. Informationen zu weiteren erforderlichen Voraussetzungen für die Verwendung von SQL Server als Quelle finden Sie unter [Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS](#).

Uneinheitliche Tabelle, die über Partitionen hinweg zugeordnet ist

Während CDC wird die Migration einer Tabelle mit einer spezialisierten Struktur unterbrochen, wenn AWS DMS CDC nicht ordnungsgemäß für die Tabelle ausführen kann. Nachrichten wie diese werden ausgegeben:

```
[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)
```

Beim Ausführen von CDC in SQL Server-Tabellen werden die SQL Server-Tlogs von AWS DMS analysiert. AWS DMS analysiert auf jedem Tlog-Datensatz hexadezimale Werte, die Daten für Spalten enthalten, die während einer Änderung eingefügt, aktualisiert oder gelöscht wurden.

Um den Hexadezimaldatensatz zu analysieren, liest AWS DMS die Tabellenmetadaten aus den SQL Server-Systemtabellen. Diese Systemtabellen identifizieren, was die speziell strukturierten Tabellenspalten sind und zeigen einige ihrer internen Eigenschaften, wie „xoffset“ und „null bit position“.

AWS DMS erwartet, dass Metadaten für alle unformatierten Partitionen der Tabelle identisch sind. In Einzelfällen haben speziell strukturierte Tabellen jedoch nicht auf allen Partitionen die gleichen Metadaten. In diesen Fällen kann AWS DMS CDC für diese Tabelle unterbrechen, um zu verhindern, dass Änderungen falsch analysiert und dem Ziel falsche Daten bereitgestellt werden. Es gibt u. a. folgende Möglichkeiten zur Umgehung des Problems:

- Wenn die Tabelle über einen gruppierten Index verfügt, führen Sie einen Indexneuaufbau durch.
- Wenn die Tabelle keinen gruppierten Index hat, fügen Sie der Tabelle einen gruppierten Index hinzu (Sie können ihn später löschen, wenn Sie möchten).

Fehlersuche bei Verwendung von Amazon Redshift

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit Amazon-Redshift-Datenbanken.

Themen

- [Laden in einen Amazon-Redshift-Cluster in einer anderen AWS-Region](#)
- [Fehler: Beziehung "attrep_apply_exceptions" bereits vorhanden](#)
- [Fehler mit Tabellen, deren Name mit „awsdms_changes“ beginnt](#)
- [Anzeige von Tabellen in Clustern mit Namen wie dms.awsdms_changes000000000XXXX](#)
- [Berechtigungen für die Verwendung mit Amazon Redshift erforderlich](#)

Laden in einen Amazon-Redshift-Cluster in einer anderen AWS-Region

Das Laden in einen Amazon-Redshift-Cluster in einer anderen AWS-Region als die, in der sich die AWS DMS-Replikations-Instance befindet, ist nicht möglich. DMS setzt voraus, dass sich die Replikations-Instance und der Amazon-Redshift-Cluster in derselben Region befinden.

Fehler: Beziehung "attrep_apply_exceptions" bereits vorhanden

Der Fehler „Relation 'awsdms_apply_exceptions' already exists“ tritt häufig auf, wenn ein Redshift-Endpunkt als PostgreSQL-Endpunkt angegeben wird. Um dieses Problem zu beheben, ändern Sie den Endpunkt und ändern Sie die Target engine in "redshift".

Fehler mit Tabellen, deren Name mit „awsdms_changes“ beginnt

Fehlermeldungen zu Tabellen mit Namen, die mit „awsdms_changes“ beginnen, können auftreten, wenn zwei Aufgaben gleichzeitig ausgeführt werden, die versuchen, Daten in den gleichen Amazon Redshift-Cluster zu laden. Aufgrund der Art und Weise, wie temporäre Tabellen benannt sind, können gleichzeitige Aufgaben in Konflikt geraten, wenn die gleiche Tabelle aktualisiert wird.

Anzeige von Tabellen in Clustern mit Namen wie dms.awsdms_changes000000000XXXX

AWS DMS erstellt temporäre Tabellen, wenn Daten von in Amazon S3 gespeicherten Dateien geladen werden. Die Namen dieser temporären Tabellen weisen das Präfix `dms.awsdms_changes` auf. Diese Tabellen sind erforderlich, damit AWS DMS Daten speichern kann, wenn sie zum ersten Mal geladen und bevor sie in der endgültigen Zieltabelle abgelegt werden.

Berechtigungen für die Verwendung mit Amazon Redshift erforderlich

Um AWS DMS mit Amazon Redshift verwenden zu können, muss das Benutzerkonto, mit dem Sie auf Amazon Redshift zugreifen, über die folgenden Berechtigungen verfügen:

- CRUD (Auswählen, Einfügen, Aktualisieren, Löschen)
- Massenladevorgang
- Erstellen, Ändern, Löschen (falls gemäß Aufgabendefinition erforderlich)

Sie finden die Voraussetzungen für die Verwendung von Amazon Redshift als Ziel unter [Verwenden einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service](#).

Fehlersuche bei Verwendung von Amazon Aurora MySQL

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit Datenbanken von Amazon Aurora MySQL.

Themen

- [Fehler: CHARACTER SET UTF8-Felder beendet durch ',' umschlossen von "" Zeilen beendet durch '\n'](#)

Fehler: CHARACTER SET UTF8-Felder beendet durch ',' umschlossen von "" Zeilen beendet durch '\n'

Wenn Sie Amazon Aurora MySQL als Ziel verwenden, wird in den Protokollen möglicherweise ein Fehler wie der folgende angezeigt. Dieser Fehlertyp weist in der Regel darauf hin, dass ANSI_QUOTES Teil des SQL_MODE-Parameters ist. Die Verwendung von ANSI_QUOTES im SQL_MODE-Parameter sorgt dafür, dass doppelte Anführungszeichen wie einfache Anführungszeichen verwendet werden, was bei Ausführung einer Aufgabe zu Problemen führen kann.

Um diesen Fehler zu beheben, entfernen Sie ANSI_QUOTES aus dem SQL_MODE-Parameter.

```
2016-11-02T14:23:48 [TARGET_LOAD ]E: Load data sql statement. load data local infile
"/rdsdbdata/data/tasks/7X04FJHCVON7TYTLQ6RX3CQHDX/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)
```

Fehlersuche bei Verwendung von SAP ASE

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit SAP-ASE-Datenbanken.

Fehler: LOB-Spalten haben NULL-Werte, wenn die Quelle einen zusammengesetzten eindeutigen Index mit NULL-Werten hat

Wenn Sie SAP ASE als Quelle mit Tabellen verwenden, die mit einem zusammengesetzten eindeutigen Index konfiguriert sind, der NULL-Werte zulässt, werden LOB-Werte während der laufenden Replikation möglicherweise nicht migriert. Dieses Verhalten ist in der Regel darauf zurückzuführen, dass ANSI_NULL auf dem Client der DMS-Replikations-Instance standardmäßig auf 1 gesetzt ist.

Um sicherzustellen, dass LOB-Felder korrekt migriert werden, fügen Sie dem AWS DMS-Quellendpunkt für die Aufgabe die Endpunkteinstellung 'AnsiNull=0' hinzu.

Fehlersuche bei Verwendung von IBM Db2

Im Folgenden erfahren Sie mehr über die Fehlersuche bei Verwendung von AWS DMS mit IBM-Db2-Datenbanken.

Fehler: Fortsetzen ab Zeitstempel wird nicht unterstützt

Wenn Sie bei einer laufenden Replikation (CDC) die Replikation ab einem bestimmten Zeitstempel starten möchten, setzen Sie das Verbindungsattribut `StartFromContext` auf den erforderlichen Zeitstempel. Weitere Informationen finden Sie unter [Endpoint settings when using Db2 LUW](#). Indem Sie `StartFromContext` auf den erforderlichen Zeitstempel setzen, wird folgender Fehler verhindert:

```
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.
```

Fehlersuche bei Latenzproblemen in AWS Database Migration Service

Dieser Abschnitt stellt einen Überblick über die häufigsten Ursachen für die Latenz von AWS DMS-Aufgaben während der laufenden Replikation (CDC) bereit. AWS DMS repliziert Daten asynchron. Latenz ist die Verzögerung zwischen dem Zeitpunkt, an dem eine Änderung an der Quelle

freigeschaltet wurde, und dem Zeitpunkt, an dem die Änderung auf das Ziel repliziert wurde. Latenz kann durch Fehlkonfigurationen von Replikationskomponenten wie beispielsweise den folgenden verursacht werden:

- Quellendpunkt oder Datenquelle
- Zielpunkt oder Datenquelle
- Replikations-Instances
- Das Netzwerk zwischen diesen Komponenten

Es wird empfohlen, eine Testmigration als Machbarkeitsnachweis zu verwenden, um Informationen über Ihre Replikation zu erfassen. Sie können diese Informationen dann verwenden, um Ihre Replikationskonfiguration zu optimieren und die Latenz zu minimieren. Informationen zur Ausführung einer Migration als Machbarkeitsnachweis finden Sie unter [Ausführen eines Machbarkeitsnachweises](#).

Themen

- [CDC-Latenztypen](#)
- [Häufige Ursachen für CDC-Latenz](#)
- [Fehlersuche bei Latenzproblemen](#)

CDC-Latenztypen

In diesem Abschnitt werden die Typen von Replikationslatenz beschrieben, die während CDC auftreten können.

Quelllatenz

Die Verzögerung zwischen dem Commit des letzten vom Quellendpunkt erfassten Ereignisses und dem aktuellen Systemzeitstempel der Replikations-Instance in Sekunden. Sie können die Latenz zwischen der Datenquelle und Ihrer Replikations-Instance mithilfe der `-CDCLatencySource` CloudWatch Metrik überwachen. Eine hohe `CDCLatencySource`-Metrik weist darauf hin, dass der Prozess der Erfassung von Änderungen aus der Quelle verzögert wird. Wenn Ihre Anwendung beispielsweise um 10:00 Uhr eine Einfügung in die Quelle freischaltet und AWS DMS die Änderung um 10:02 Uhr verarbeitet, beträgt die Metrik `CDCLatencySource` 120 Sekunden.

Weitere Informationen zu CloudWatch Metriken für AWS DMS finden Sie unter [Metriken für die Replikationsaufgabe](#).

Ziellatenz

Die Verzögerung zwischen dem Commit des ersten Ereignisses in der Quelle, das für das Ziel freigeschaltet werden soll, und dem aktuellen Zeitstempel der DMS-Replikations-Instance in Sekunden. Sie können die Latenz zwischen Commits in der Datenquelle und Ihrem Datenziel mithilfe der `-CDCLatencyTarget` CloudWatch Metrik überwachen. Dies bedeutet, dass `CDCLatencyTarget` auch Verzögerungen beim Lesen aus der Quelle berücksichtigt. Daher ist `CDCLatencyTarget` immer größer als oder gleich `CDCLatencySource`.

Wenn Ihre Anwendung beispielsweise um 10:00 Uhr eine Einfügung in die Quelle freischaltet und AWS DMS diese um 10:02 Uhr verarbeitet und um 10:05 Uhr ins Ziel schreibt, beträgt die Metrik `CDCLatencyTarget` 300 Sekunden.

Häufige Ursachen für CDC-Latenz

In diesem Abschnitt werden die Ursachen von Latenz beschrieben, die während CDC bei Ihrer Replikation auftreten können.

Themen

- [Endpunktressourcen](#)
- [Ressourcen für Replikations-Instances](#)
- [Netzwerkgeschwindigkeit und -bandbreite](#)
- [DMS-Konfiguration](#)
- [Replikationsszenarien](#)

Endpunktressourcen

Die folgenden Faktoren haben erhebliche Auswirkungen auf die Replikationsleistung und -latenz:

- Konfigurationen der Quell- und Zieldatenbank
- Instance-Größe
- Zu wenig bereitgestellter oder falsch konfigurierter Quell- oder Zieldatenspeicher

Überwachen Sie die folgenden CloudWatch Metriken, um Latenzursachen zu ermitteln, die durch Endpunktprobleme für von AWSgehostete Quellen und Ziele verursacht werden:

- `FreeMemory`

- CPUUtilization
- Durchsatz- und E/A-Metriken wie beispielsweise WriteIOPS, WriteThroughput oder ReadLatency
- Metriken zum Transaktionsvolumen wie beispielsweise CDCIncomingChanges

Informationen zur Überwachung von CloudWatch Metriken finden Sie unter [AWS Database Migration Service-Metriken](#).

Ressourcen für Replikations-Instances

Die Ressourcen für Replikations-Instances sind für die Replikation entscheidend. Sie sollten sicherstellen, dass es keine Ressourcenengpässe gibt, da diese sowohl Quell- als auch Ziellatenz verursachen können.

Überprüfen Sie Folgendes, um Ressourcenengpässe für Ihre Replikations-Instance zu identifizieren:

- Kritische CloudWatch Metriken wie CPU, Arbeitsspeicher, E/A pro Sekunde und Speicher weisen keine Spitzen oder konstant hohe Werte auf.
- Ihre Replikations-Instance ist für Ihren Workload angemessen dimensioniert. Informationen zur Ermittlung der richtigen Größe einer Replikations-Instance finden Sie unter [Auswahl der besten Größe für eine Replikations-Instance](#).

Netzwerkgeschwindigkeit und -bandbreite

Die Netzwerkbandbreite ist ein Faktor, der sich auf die Datenübertragung auswirkt. Führen Sie einen der folgenden Schritte aus, um die Netzwerkleistung Ihrer Replikation zu analysieren:

- Überprüfen Sie die Metriken ReadThroughput und WriteThroughput auf Instance-Ebene. Informationen zur Überwachung von CloudWatch Metriken finden Sie unter [AWS Database Migration Service-Metriken](#).
- Verwenden Sie das AWS DMS-Diagnosesupport-AMI. Wenn das Diagnosesupport-AMI in Ihrer Region nicht verfügbar ist, können Sie es aus jeder unterstützten Region herunterladen und in Ihre Region kopieren, um eine Netzwerkanalyse vorzunehmen. Informationen zum Diagnosesupport-AMI finden Sie unter [Arbeiten mit dem AWS DMS-Diagnoseunterstützungs-AMI](#).

CDC in AWS DMS ist ein Single-Thread-Prozess, um die Datenkonsistenz sicherzustellen. Daher können Sie das Datenvolumen bestimmen, das Ihr Netzwerk unterstützen kann, indem Sie Ihre

Single-Thread-Datenübertragungsrate berechnen. Wenn Ihre Aufgabe beispielsweise über ein Netzwerk mit 100 Mbit/s (Megabit pro Sekunde) eine Verbindung mit ihrer Quelle herstellt, hat Ihre Replikation theoretisch eine maximale Bandbreitenzuweisung von 12,5 MB/s (Megabyte pro Sekunde). Das entspricht 45 Gigabit pro Stunde. Wenn die Rate der Generierung von Transaktionsprotokollen auf der Quelle mehr als 45 Gigabit pro Stunde beträgt, bedeutet dies, dass bei der Aufgabe CDC-Latenz besteht. Bei einem Netzwerk mit 100 Mbit/s handelt es sich bei diesen Raten um theoretische Höchstwerte. Andere Faktoren wie beispielsweise Netzwerkverkehr und Ressourcenaufwand für Quelle und Ziel reduzieren die tatsächlich verfügbare Bandbreite.

DMS-Konfiguration

Dieser Abschnitt enthält empfohlene Replikationskonfigurationen, die zur Verringerung der Latenz beitragen können.

- **Endpunkteinstellungen:** Ihre Einstellungen für den Quell- und Zielendpunkt können Leistungseinbußen bei Ihrer Replikations-Instance verursachen. Endpunkteinstellungen, die ressourcenintensive Features aktivieren, wirken sich auf die Leistung aus. Beispielsweise verbessert das Deaktivieren LogMiner und Verwenden von Binary Reader für einen Oracle-Endpunkt die Leistung, da ressourcenintensiv LogMiner ist. Die folgende Endpunkteinstellung verbessert die Leistung eines Oracle-Endpunkts:

```
useLogminerReader=N;useBfile=Y
```

Weitere Informationen zu Endpunkteinstellungen finden Sie in der Dokumentation zu Ihrer Quell- und Ziel-Endpunkt-Engine im Thema [Arbeiten mit AWS-DMS-Endpunkten](#).

- **Aufgabeneinstellungen:** Einige Aufgabeneinstellungen für Ihr spezielles Replikationsszenario können Leistungseinbußen bei Ihrer Replikations-Instance verursachen. Beispielsweise verwendet AWS DMS standardmäßig den transaktionalen Anwendungsmodus (`BatchApplyEnabled=false`) für CDC für alle Endpunkte außer Amazon Redshift. Bei Quellen mit einer großen Anzahl von Änderungen lässt sich die Leistung jedoch möglicherweise verbessern, wenn `BatchApplyEnabled` auf `true` gesetzt wird.

Weitere Informationen zu den Aufgabeneinstellungen finden Sie unter [Angeben von Aufgabeneinstellungen für Aufgaben des AWS Database Migration Service](#).

- **Startposition einer reinen CDC-Aufgabe:** Wird eine reine CDC-Aufgabe von einer Position oder einem Zeitstempel in der Vergangenheit aus gestartet, wird die Aufgabe mit einer erhöhten CDC-Quelllatenz gestartet. Je nach Umfang der Änderungen an der Quelle dauert es einige Zeit, bis die Aufgabenlatenz nachlässt.

- LOB-Einstellungen: Datentypen mit großen Objekten können aufgrund der Art und Weise, wie AWS DMS große Binärdaten repliziert, die Replikationsleistung beeinträchtigen. Weitere Informationen finden Sie unter den folgenden Themen:
 - [Einstellung der LOB-Unterstützung für Quelldatenbanken in einer Aufgabe AWS DMS](#)
 - [Migrieren von Binary Large Objects \(LOBs\)](#).

Replikationsszenarien

In diesem Abschnitt werden spezifische Replikationsszenarien und deren mögliche Auswirkungen auf die Latenz beschrieben.

Themen

- [Anhalten einer Aufgabe für einen längeren Zeitraum](#)
- [Zwischengespeicherte Änderungen](#)
- [Regionsübergreifende Replikation](#)

Anhalten einer Aufgabe für einen längeren Zeitraum

Wenn Sie eine Aufgabe anhalten, speichert AWS DMS die Position des letzten Transaktionsprotokolls, das aus der Quelle gelesen wurde. Wenn Sie die Aufgabe fortsetzen, versucht DMS, von derselben Position des Transaktionsprotokolls aus mit dem Lesevorgang fortzufahren. Wird eine Aufgabe nach mehreren Stunden oder Tagen fortgesetzt, erhöht sich die CDC-Quelllatenz, bis DMS den Transaktionsrückstand abgearbeitet hat.

Zwischengespeicherte Änderungen

Zwischengespeicherte Änderungen sind Änderungen, die Ihre Anwendung in die Datenquelle schreibt, während AWS DMS die Volllast-Replikationsphase ausführt. DMS wendet diese Änderungen erst an, wenn die Volllastphase abgeschlossen ist und die CDC-Phase beginnt. Bei einer Quelle mit einer großen Anzahl von Transaktionen dauert das Übernehmen zwischengespeicherter Änderungen länger. Daher nimmt die Quelllatenz zu, wenn die CDC-Phase beginnt. Wir empfehlen, die Volllastphase auszuführen, wenn das Transaktionsvolumen gering ist, um die Anzahl der zwischengespeicherten Änderungen zu minimieren.

Regionsübergreifende Replikation

Wenn Sie Ihre DMS-Endpunkte oder Ihre Replikations-Instance in verschiedenen AWS-Regionen einrichten, erhöht sich die Netzwerklatenz. Dadurch erhöht sich auch die Replikationslatenz. Um eine

optimale Leistung zu erzielen, sollten Sie den Quell- und den Zielpunkt sowie die Replikations-Instance in derselben AWS-Region einrichten.

Fehlersuche bei Latenzproblemen

Dieser Abschnitt enthält Schritte zur Fehlerbehebung bei der Replikationslatenz.

Gehen Sie wie folgt vor, um Latenzprobleme zu beheben:

- Ermitteln Sie zunächst die Art und den Umfang der Latenz für die Aufgabe. Überprüfen Sie in der DMS-Konsole oder der CLI den Abschnitt Tabellenstatistik der Aufgabe. Wenn sich die Zähler ändern, wird gerade eine Datenübertragung ausgeführt. Überprüfen Sie die Metriken `CDCLatencySource` und `CDCLatencyTarget` zusammen, um zu ermitteln, ob es während CDC einen Engpass gibt.
- Wenn hohe `CDCLatencySource`- oder `CDCLatencyTarget`-Metriken auf einen Engpass in Ihrer Replikation hinweisen, überprüfen Sie Folgendes:
 - Wenn `CDCLatencySource` hoch und `CDCLatencyTarget` mit `CDCLatencySource` identisch ist, deutet dies darauf hin, dass ein Engpass in Ihrem Quellendpunkt vorliegt und AWS DMS die Daten reibungslos in das Ziel schreibt. Weitere Informationen finden Sie im Nachfolgenden unter [Fehlersuche bei Problemen mit der Quelllatenz](#).
 - Wenn `CDCLatencySource` niedrig und `CDCLatencyTarget` hoch ist, deutet dies darauf hin, dass ein Engpass in Ihrem Zielpunkt vorliegt und AWS DMS die Daten reibungslos aus der Quelle liest. Weitere Informationen finden Sie im Nachfolgenden unter [Fehlersuche bei Problemen mit der Ziellatenz](#).
 - Wenn `CDCLatencySource` hoch und `CDCLatencyTarget` deutlich höher als `CDCLatencySource` ist, deutet dies auf Engpässe sowohl beim Lesen aus der Quelle als auch beim Schreiben in das Ziel hin. Untersuchen Sie zuerst die Quell- und dann die Ziellatenz.

Informationen zur Überwachung von Aufgabenmetriken in DMS finden Sie unter [Überwachen von AWS-DMS-Aufgaben](#).

Fehlersuche bei Problemen mit der Quelllatenz

In den folgenden Themen werden spezifische Replikationsszenarien für Quellendpunkttypen beschrieben.

Themen

- [Fehlersuche bei Oracle-Endpunkten](#)

- [Fehlersuche bei MySQL-Endpunkten](#)
- [Fehlersuche bei PostgreSQL-Endpunkten](#)
- [Fehlersuche bei SQL-Server-Endpunkten](#)

Fehlersuche bei Oracle-Endpunkten

Dieser Abschnitt enthält spezifische Replikationsszenarien für Oracle.

Das Lesen aus der Quelle wurde angehalten

In den folgenden Szenarien unterbricht AWS DMS das Lesen aus einer Oracle-Quelle. Dieses Verhalten ist beabsichtigt. Sie können die Ursachen dafür mithilfe des Aufgabenprotokolls untersuchen. Suchen Sie im Aufgabenprotokoll nach Meldungen, die den folgenden ähneln. Weitere Informationen zur Arbeit mit dem Aufgabenprotokoll finden Sie unter [Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen](#).

- SORTER-Meldung: Dies weist darauf hin, dass DMS Transaktionen auf der Replikations-Instance zwischenspeichert. Weitere Informationen finden Sie unter [SORTER-Meldung im Aufgabenprotokoll](#).
- Aufgabenprotokolle debuggen: Wenn DMS den Lesevorgang unterbricht, schreibt Ihre Aufgabe wiederholt die folgende Meldung in die Debug-Aufgabenprotokolle, ohne Änderungen am Kontextfeld oder Zeitstempel:
 - Binary Reader:

```
[SOURCE_CAPTURE ]T: Produce CTI event:
context '00000020.f23ec6e5.00000002.000a.00.0000:190805.3477731.16'
xid [00000000001e0018] timestamp '2021-07-19 06:57:55'
thread 2 (oradcdc_oralog.c:817)
```

- Logminer:

```
[SOURCE_CAPTURE ]T: Produce INSERT event:
object id 1309826 context
'000000000F2CECAA010000010005A8F500000275016C0000000000000F2CEC58'
xid [000014e06411d996] timestamp '2021-08-12 09:20:32' thread 1
(oradcdc_reader.c:2269)
```

- AWS DMS protokolliert die folgende Meldung für jeden neuen Redo- oder archivierten Protokollvorgang.

```
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)
```

Wenn die Quelle neue Redo- oder archivierte Protokollvorgänge aufweist und AWS DMS diese Meldungen nicht in das Protokoll schreibt, bedeutet dies, dass die Aufgabe keine Ereignisse verarbeitet.

Hohe Redo-Generierung

Wenn Ihre Aufgabe Redo- oder archivierte Protokolle verarbeitet, die Quelllatenz jedoch hoch bleibt, versuchen Sie, die Redo-Protokollgenerierungsrate und -generierungsmuster zu ermitteln. Eine hohe Redo-Protokollgenerierungsrate führt zu einer höheren Quelllatenz, da Ihre Aufgabe alle Redo- und Archivierungsprotokolle liest, um Änderungen im Zusammenhang mit den replizierten Tabellen abzurufen.

Verwenden Sie die folgenden Abfragen, um die Redo-Generierungsrate zu ermitteln.

- Redo-Generierungsrate pro Tag:

```
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;
```

- Redo-Generierungsrate pro Stunde:

```
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 ;
```

Überprüfen Sie Folgendes, um Latenzprobleme in diesem Szenario zu beheben:

- Überprüfen Sie die Netzwerkbandbreite und die Single-Thread-Leistung Ihrer Replikation, um sicherzustellen, dass Ihr zugrunde liegendes Netzwerk die Redo-Generierungsrate der Quelle unterstützen kann. Informationen zu den möglichen Auswirkungen der Netzwerkbandbreite auf die Replikationsleistung finden Sie unter [Netzwerkgeschwindigkeit und -bandbreite](#).
- Überprüfen Sie, ob Sie die zusätzliche Protokollierung richtig eingerichtet haben. Vermeiden Sie zusätzliche Protokollierung an der Quelle, wie beispielsweise die Aktivierung der Protokollierung für alle Spalten einer Tabelle. Weitere Informationen zum Einrichten der zusätzlichen Protokollierung finden Sie unter [Einrichten der zusätzlichen Protokollierung](#).
- Stellen Sie sicher, dass Sie die richtige API verwenden, um die Redo- oder archivierten Protokolle zu lesen. Sie können entweder Oracle LogMiner oder AWS DMS Binary Reader verwenden. Während die Online-Redo-Protokolle und archivierten Redo-Protokolldateien LogMiner liest, liest und analysiert Binary Reader die unformatierten Redo-Protokolldateien direkt. Daher ist Binary Reader leistungsfähiger. Wir empfehlen, Binary Reader zu verwenden, wenn die Redo-Protokollgenerierung mehr als 10 GB/Stunde umfasst. Weitere Informationen finden Sie unter [Verwenden von Oracle LogMiner oder AWS DMS Binary Reader für CDC](#).
- Überprüfen Sie, ob Sie `ArchivedLogsOnly` auf `Y` gesetzt haben. Wenn diese Endpunkteinstellung festgelegt ist, liest AWS DMS aus den archivierten Redo-Protokollen. Dadurch erhöht sich die Quelllatenz, da AWS DMS erst mit dem Lesevorgang beginnt, nachdem das Online-Redo-Protokoll archiviert wurde. Weitere Informationen finden Sie unter [ArchivedLogsOnly](#).
- Wenn Ihre Oracle-Quelle Automatic Storage Management (ASM) verwendet, erhalten Sie unter [REDO wird auf Oracle ASM gespeichert, wenn Oracle als Quelle für verwendet wird AWS DMS](#) Informationen zur ordnungsgemäßen Konfiguration Ihres Datenspeichers. Möglicherweise können Sie die Leseleistung auch weiter optimieren, wenn Sie das zusätzliche Verbindungsattribut (Extra Connection Attribute, ECA) `asmUsePLSQLArray` verwenden. Für weitere Informationen zur Nutzung von `asmUsePLSQLArray` siehe [Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS](#).

Fehlersuche bei MySQL-Endpunkten

Dieser Abschnitt enthält spezifische Replikationsszenarien für MySQL. AWS DMS scannt das MySQL-Binärprotokoll regelmäßig, um Änderungen zu replizieren. Dieser Prozess kann die Latenz in den folgenden Szenarien erhöhen:

Themen

- [Lang laufende Transaktion in der Quelle](#)
- [Hoher Workload in der Quelle](#)

- [Binärprotokoll-Konflikt](#)

Lang laufende Transaktion in der Quelle

Da MySQL nur freigeschaltete Transaktionen in das Binärprotokoll schreibt, verursachen lang laufende Transaktionen Latenzspitzen, die proportional zur Abfrageausführungszeit sind.

Verwenden Sie die folgende Abfrage oder das Slow-Query-Protokoll, um lang laufende Transaktionen zu identifizieren:

```
SHOW FULL PROCESSLIST;
```

Informationen zur Verwendung des Slow-Query-Protokolls finden Sie unter [The Slow Query Log](#) in der [MySQL-Dokumentation](#).

Strukturieren Sie Ihre Quelltransaktionen neu, um entweder die Abfrageausführungszeit zu verringern oder die Häufigkeit der Commits zu erhöhen und dadurch Latenzspitzen aufgrund lang laufender Transaktionen zu vermeiden.

Hoher Workload in der Quelle

Da es sich bei CDC in DMS um einen Single-Thread-Prozess handelt, kann eine große Anzahl von Transaktionen zu einer höheren Quelllatenz führen. Vergleichen Sie die Anzahl und Größe der während der Latenzzeit generierten Binärprotokolle mit den vor der Latenzzeit generierten Protokollen, um zu ermitteln, ob die Quelllatenz auf einen hohen Workload zurückzuführen ist. Verwenden Sie die folgenden Abfragen, um die Binärprotokolle und den Status des DMS-CDC-Threads zu überprüfen:

```
SHOW BINARY LOGS;  
SHOW PROCESSLIST;
```

Weitere Informationen zu den Status von CDC-Binärprotokoll-Dump-Threads finden Sie unter [Replication Source Thread States](#).

Sie können die Latenz ermitteln, indem Sie die letzte Binärprotokoll-Position, die in der Quelle generiert wurde, mit dem Ereignis vergleichen, das DMS gerade verarbeitet. Gehen Sie wie folgt vor, um das neueste Binärprotokoll der Quelle zu identifizieren:

- Aktivieren Sie Debug-Protokolle für die Komponente SOURCE_CAPTURE.

- Rufen Sie das Binärprotokoll zur DMS-Verarbeitung und die Positionsdetails aus den Debug-Protokollen der Aufgabe ab.
- Verwenden Sie die folgende Abfrage, um das neueste Binärprotokoll der Quelle zu identifizieren:

```
SHOW MASTER STATUS;
```

Optimieren Sie den Wert für `EventsPollInterval`, um die Leistung weiter zu verbessern. Standardmäßig fragt DMS das Binärprotokoll alle 5 Sekunden ab. Sie können die Leistung jedoch verbessern, indem Sie diesen Wert reduzieren. Weitere Informationen zur `EventsPollInterval`-Einstellung finden Sie unter [Endpunkteinstellungen bei Verwendung von MySQL als Quelle für AWS DMS](#).

Binärprotokoll-Konflikt

Für die Migration mehrerer Tabellen mit einer großen Datenmenge empfehlen wir, die Tabellen in separate Aufgaben für MySQL 5.7.2 oder höher aufzuteilen. In den MySQL-Versionen 5.7.2 und höher erzeugt der Master-Dump-Thread weniger Sperrkonflikte und verbessert den Durchsatz. Daher sperrt der Dump-Thread das Binärprotokoll nicht mehr, wenn er ein Ereignis liest. Dies bedeutet, dass mehrere Dump-Threads die Binärprotokolldatei gleichzeitig lesen können. Außerdem bedeutet dies, dass Dump-Threads das Binärprotokoll lesen können, während Clients darin schreiben. Weitere Informationen zu Dump-Threads finden Sie unter [Replication Threads](#) und in den [Versionshinweisen zu MySQL 5.7.2](#).

Zum Verbessern der Replikationsleistung für MySQL-Quellversionen vor 5.7.2 sollten Sie versuchen, Aufgaben mit CDC-Komponenten zusammenzufassen.

Fehlersuche bei PostgreSQL-Endpunkten

Dieser Abschnitt enthält spezifische Replikationsszenarien für PostgreSQL.

Themen

- [Lang laufende Transaktion in der Quelle](#)
- [Hoher Workload in der Quelle](#)
- [Hohen Netzwerkdurchsatz](#)
- [Überlaufdateien in Aurora PostgreSQL](#)

Lang laufende Transaktion in der Quelle

Wenn in der Quelldatenbank lang laufende Transaktionen vorhanden sind, z. B. einige Tausend Einfügungen in einer einzelnen Transaktion, steigen die DMS-CDC-Ereignis- und Transaktionszähler erst an, wenn die Transaktion abgeschlossen ist. Diese Verzögerung kann Latenzprobleme verursachen, die Sie anhand der Metrik `CDCLatencyTarget` messen können.

Führen Sie einen der folgenden Schritte aus, um lang laufende Transaktionen zu überprüfen:

- Verwenden Sie die Ansicht `pg_replication_slots`. Wenn der Wert `restart_lsn` nicht aktualisiert wird, kann PostgreSQL wahrscheinlich aufgrund von lang laufenden aktiven Transaktionen keine Write-Ahead-Protokolle (WALs) veröffentlichen. Informationen zur Ansicht `pg_replication_slots` finden Sie unter [pg_replication_slots](#) in der [Dokumentation zu PostgreSQL 15.4](#).
- Verwenden Sie die folgende Abfrage, um eine Liste aller aktiven Abfragen in der Datenbank zusammen mit den zugehörigen Informationen zurückzugeben:

```
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;
```

Das Feld `age` in den Abfrageergebnissen zeigt die aktive Dauer der einzelnen Abfragen an, anhand derer Sie lang laufende Abfragen identifizieren können.

Hoher Workload in der Quelle

Wenn Ihre PostgreSQL-Quelle einen hohen Workload aufweist, überprüfen Sie Folgendes, um die Latenz zu reduzieren:

- Wenn Sie das Plug-in `test_decoding` während der Migration einer Teilmenge von Tabellen aus der Quelldatenbank mit einem hohen TPS-Wert (Transaktionen pro Sekunde) verwenden, kann es zu einer hohen Latenz kommen. Dies liegt daran, dass das Plug-in `test_decoding` alle Datenbankänderungen an die Replikations-Instance sendet, die DMS dann basierend auf der Tabellenzuordnung der Aufgabe filtert. Ereignisse für Tabellen, die nicht Teil der Tabellenzuordnung der Aufgabe sind, können die Quelllatenz erhöhen.
- Überprüfen Sie den TPS-Durchsatz anhand einer der folgenden Methoden:
 - Verwenden Sie für Aurora-PostgreSQL-Quellen die `-CommitThroughput` CloudWatch Metrik.

- Verwenden Sie für PostgreSQL, das in Amazon RDS oder On-Premises ausgeführt wird, die folgende Abfrage mit einem PSQL-Client der Version 11 oder höher (drücken Sie während der Abfrage **enter**, um die Ergebnisse weiterzuleiten):

```
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";
```

- Um die Latenz bei der Verwendung des Plug-ins `test_decoding` zu reduzieren, sollten Sie stattdessen das Plug-in `pglogical` verwenden. Im Gegensatz zum Plug-in `test_decoding` filtert das Plug-in `pglogical` Änderungen an Write-Ahead-Protokollen (WALs) an der Quelle und sendet nur relevante Änderungen an die Replikations-Instance. Informationen zur Verwendung des Plug-ins `pglogical` mit AWS DMS finden Sie unter [Konfigurieren des Plug-ins pglogical](#).

Hohen Netzwerkdurchsatz

Bei Ihrer Replikation wird möglicherweise eine hohe Netzwerkbandbreite beansprucht, wenn Sie das Plug-in `test_decoding` verwenden, insbesondere bei Transaktionen mit hohem Volumen. Dies liegt daran, dass das Plug-in `test_decoding` Änderungen verarbeitet und in ein für Menschen lesbares Format umwandelt, das größer als das ursprüngliche Binärformat ist.

Zum Verbessern der Leistung sollten Sie stattdessen das Plug-in `pglogical` verwenden, bei dem es sich um ein binäres Plug-in handelt. Im Gegensatz zum Plug-in `test_decoding` generiert das Plug-in `pglogical` eine Ausgabe im Binärformat, was zu komprimierten Änderungen des Write-Ahead-Protokoll (WAL)-Streams führt.

Überlaufdateien in Aurora PostgreSQL

In PostgreSQL Version 13 und höher bestimmt der `logical_decoding_work_mem` Parameter die Speicherzuweisung für die Dekodierung und das Streaming. Weitere Informationen zum `logical_decoding_work_mem` Parameter finden Sie unter [Ressourcennutzung in PostgreSQL](#) in der [PostgreSQL 13.13-Dokumentation](#).

Die logische Replikation akkumuliert Änderungen für alle Transaktionen im Speicher, bis diese Transaktionen festgeschrieben werden. Wenn die Menge der in allen Transaktionen gespeicherten Daten die durch den Datenbankparameter angegebene Menge überschreitet `logical_decoding_work_mem`, übergibt DMS die Transaktionsdaten auf die Festplatte, um Speicher für neue Dekodierungsdaten freizugeben.

Lang andauernde Transaktionen oder viele Untertransaktionen können dazu führen, dass DMS mehr logischen Dekodierungsspeicher verbraucht. Dieser erhöhte Speicherverbrauch führt dazu, dass DMS Spill-Dateien auf der Festplatte erstellt, was zu einer hohen Quelllatenz während der Replikation führt.

Gehen Sie wie folgt vor, um die Auswirkungen einer Erhöhung der Quell-Workload zu reduzieren:

- Reduzieren Sie lang andauernde Transaktionen.
- Reduzieren Sie die Anzahl der Teiltransaktionen.
- Vermeiden Sie das Ausführen von Operationen, die einen großen Burst von Protokolldatensätzen erzeugen, z. B. das Löschen oder Aktualisieren einer gesamten Tabelle in einer einzigen Transaktion. Führen Sie stattdessen Operationen in kleineren Stapeln aus.

Sie können die folgenden CloudWatch Metriken verwenden, um den Workload auf der Quelle zu überwachen:

- `TransactionLogsDiskUsage`: Die Anzahl der Bytes, die derzeit vom logischen WAL belegt werden. Dieser Wert steigt monoton, wenn logische Replikationsslots nicht mit dem Tempo neuer Schreibvorgänge Schritt halten können oder wenn lang andauernde Transaktionen die Garbage Collection älterer Dateien verhindern.
- `ReplicationSlotDiskUsage`: Der Speicherplatz, den die logischen Replikations-Slots derzeit verwenden.

Sie können die Quelllatenz reduzieren, indem Sie den `logical_decoding_work_mem` Parameter optimieren. Der Standardwert für diesen Parameter ist 64 MB. Dieser Parameter begrenzt die Speichermenge, die von jeder logischen Streaming-Replikationsverbindung verwendet wird. Wir empfehlen, den `logical_decoding_work_mem` Wert deutlich höher als den `work_mem` Wert festzulegen, um die Menge der dekodierten Änderungen zu reduzieren, die DMS auf die Festplatte schreibt.

Wir empfehlen Ihnen, regelmäßig nach Spill-Dateien zu suchen, insbesondere in Zeiten hoher Migrationsaktivität oder Latenz. Wenn DMS eine erhebliche Anzahl von Spill-Dateien erstellt, bedeutet dies, dass die logische Dekodierung nicht effizient funktioniert, was die Latenz erhöhen kann. Um dies zu minimieren, erhöhen Sie den `logical_decoding_work_mem` Parameterwert.

Sie können den aktuellen Transaktionsüberlauf mit der `aurora_stat_file` Funktion überprüfen. Weitere Informationen finden Sie unter [Anpassen des Arbeitsspeichers für die logische Dekodierung](#) im Amazon Relational Database Service-Entwicklerhandbuch.

Fehlersuche bei SQL-Server-Endpunkten

Dieser Abschnitt enthält spezifische Replikationsszenarien für SQL Server. Um zu ermitteln, welche Änderungen von SQL Server repliziert werden sollen, liest AWS DMS die Transaktionsprotokolle und führt regelmäßige Scans der Quelldatenbank aus. Die Replikationslatenz ist in der Regel darauf zurückzuführen, dass SQL Server diese Scans aufgrund von Ressourcenbeschränkungen drosselt. Sie kann auch auf einen deutlichen Anstieg der Anzahl von Ereignissen zurückzuführen sein, die innerhalb kurzer Zeit in das Transaktionsprotokoll geschrieben werden.

Themen

- [Index-Neuerstellungen](#)
- [Große Transaktionen](#)
- [Falsch konfiguriertes MS-CDC-Abfrageintervall für Amazon RDS SQL Server](#)
- [Replikation mehrerer CDC-Aufgaben aus derselben Quelldatenbank](#)

Index-Neuerstellungen

Wenn SQL Server einen großen Index neu erstellt, wird eine einzige Transaktion verwendet. Dadurch werden viele Ereignisse generiert und möglicherweise viel Protokollspeicher beansprucht, wenn SQL Server mehrere Indizes gleichzeitig neu erstellt. In diesem Fall können Sie mit kurzen Replikationsspitzen rechnen. Wenn Ihre SQL-Server-Quelle anhaltende Protokollspitzen aufweist, überprüfen Sie Folgendes:

- Überprüfen Sie zunächst den Zeitraum der Latenzspitzen mithilfe der `CDCLatencySource` CloudWatch Metriken `CDCLatencySource` und oder durch Überprüfen der Durchsatzüberwachungsmeldungen in den Aufgabenprotokollen. Weitere Informationen zu CloudWatch Metriken für finden Sie AWS DMSunter [Metriken für die Replikationsaufgabe](#).
- Überprüfen Sie, ob die Größe der aktiven Transaktionsprotokolle oder Protokoll-Backups während der Latenzspitze zugenommen hat. Überprüfen Sie außerdem, ob während dieser Zeit ein Wartungsauftrag oder eine Neuerstellung ausgeführt wurde. Informationen zur Überprüfung der Größe des Transaktionsprotokolls finden Sie unter [Monitor log space use](#) in der [technischen Dokumentation zu SQL Server](#).

- Stellen Sie sicher, dass Ihr Wartungsplan den bewährten Methoden für SQL Server entspricht. Informationen zu den bewährten Methoden für die Wartung von SQL Server finden Sie unter [Index maintenance strategy](#) in der [technischen Dokumentation zu SQL Server](#).

Versuchen Sie Folgendes, um Latenzprobleme während Index-Neuerstellungen zu beheben:

- Verwenden Sie das Wiederherstellungsmodell BULK_LOGGED für Offline-Neuerstellungen, um die Anzahl der Ereignisse zu reduzieren, die eine Aufgabe verarbeiten muss.
- Beenden Sie die Aufgabe während der Index-Neuerstellung, wenn möglich. Versuchen Sie alternativ, Index-Neuerstellungen außerhalb der Spitzenzeiten zu planen, um die Auswirkungen einer Latenzspitze zu mildern.
- Versuchen Sie, Ressourcenengpässe zu identifizieren und zu beheben, die DMS-Lesevorgänge verlangsamen, z. B. Festplattenlatenz oder E/A-Durchsatz.

Große Transaktionen

Transaktionen mit vielen Ereignissen oder lang laufende Transaktionen lassen das Transaktionsprotokoll anwachsen. Dadurch dauern DMS-Lesevorgänge länger, was zu Latenz führt. Dies ist vergleichbar mit den Auswirkungen von Index-Neuerstellungen auf die Replikationsleistung.

Möglicherweise haben Sie Schwierigkeiten, dieses Problem zu identifizieren, wenn Sie mit dem typischen Workload in der Quelldatenbank nicht vertraut sind. Gehen Sie wie folgt vor, um dieses Problem zu beheben:

- Identifizieren Sie zunächst die Zeit, zu der die Latenz entweder mithilfe der WriteThroughput CloudWatch Metriken ReadThroughput und oder durch Überprüfen der Durchsatzüberwachungsmeldungen in den Aufgabenprotokollen stark ansteigte.
- Überprüfen Sie, ob es in der Quelldatenbank lang laufende Abfragen während der Latenzspitze gibt. Informationen zu lang laufenden Abfragen finden Sie unter [Troubleshoot slow-running queries in SQL Server](#) in der [technischen Dokumentation zu SQL Server](#).
- Überprüfen Sie, ob die Größe der aktiven Transaktionsprotokolle oder Protokoll-Backups zugenommen hat. Weitere Informationen finden Sie unter [Monitor log space use](#) in der [technischen Dokumentation zu SQL Server](#).

Führen Sie einen der folgenden Schritte aus, um dieses Problem zu beheben:

- Die beste Lösung besteht darin, Ihre Transaktionen auf der Anwendungsseite so umzustrukturieren, dass sie schnell abgeschlossen werden.
- Wenn Sie Ihre Transaktionen nicht umstrukturieren können, besteht eine kurzfristige Lösung darin, nach Ressourcenengpässen wie Festplattenwartezeiten oder CPU-Konflikten zu suchen. Wenn Sie Engpässe in Ihrer Quelldatenbank feststellen, können Sie die Latenz reduzieren, indem Sie die Festplatten-, CPU- und Speicherressourcen für die Quelldatenbank erhöhen. Dadurch werden Konflikte um Systemressourcen reduziert, sodass DMS-Abfragen schneller abgeschlossen werden können.

Falsch konfiguriertes MS-CDC-Abfrageintervall für Amazon RDS SQL Server

Eine falsch konfigurierte Einstellung für das Abfrageintervall auf Amazon-RDS-Instances kann das Transaktionsprotokoll anwachsen lassen. Das liegt daran, dass die Replikation die Protokollkürzung verhindert. Während laufende Aufgaben möglicherweise weiterhin mit minimaler Latenz repliziert werden, kann das Anhalten und Fortsetzen von Aufgaben oder das Starten von reinen CDC-Aufgaben Aufgabenfehler verursachen. Diese sind auf Zeitüberschreitungen während des Scannens des großen Transaktionsprotokolls zurückzuführen.

Gehen Sie wie folgt vor, um Probleme mit falsch konfigurierten Abfrageintervallen zu beheben:

- Überprüfen Sie, ob die Größe des aktiven Transaktionsprotokolls zunimmt und ob die Protokollnutzung nahezu 100 Prozent beträgt. Weitere Informationen finden Sie unter [Monitor log space use](#) in der [technischen Dokumentation zu SQL Server](#).
- Überprüfen Sie, ob die Protokollkürzung mit dem `log_reuse_wait_desc` value-Wert REPLICATION verzögert wird. Weitere Informationen finden Sie unter [The Transaction Log \(SQL Server\)](#) in der [technischen Dokumentation zu SQL Server](#).

Wenn Sie Probleme mit einem der Elemente in der vorherigen Liste feststellen, passen Sie das MS-CDC-Abfrageintervall an. Informationen zur Optimierung des Abfrageintervalls finden Sie unter [Empfohlene Einstellungen bei Verwendung von Amazon RDS für SQL Server als Quelle für AWS DMS](#).

Replikation mehrerer CDC-Aufgaben aus derselben Quelldatenbank

Während der Vollastphase empfehlen wir, Tabellen auf mehrere Aufgaben aufzuteilen, um die Leistung zu verbessern, abhängige Tabellen logisch voneinander zu trennen und die Auswirkungen eines Aufgabenfehlers zu minimieren. Während der CDC-Phase empfehlen wir jedoch, die Aufgaben zusammenzufassen, um die Anzahl der DMS-Scans zu minimieren. Während der CDC-Phase scannt

jede DMS-Aufgabe die Transaktionsprotokolle mehrmals pro Minute auf neue Ereignisse. Da jede Aufgabe unabhängig ausgeführt wird, scannt jede Aufgabe jedes Transaktionsprotokoll einzeln. Dies erhöht die Festplatten- und CPU-Auslastung in der SQL-Server-Quelldatenbank. Infolgedessen kann eine große Anzahl von parallel ausgeführten Aufgaben dazu führen, dass SQL Server DMS-Lesevorgänge drosselt, was eine erhöhte Latenz verursacht.

Möglicherweise haben Sie Schwierigkeiten, dieses Problem zu identifizieren, wenn mehrere Aufgaben schrittweise gestartet werden. Das häufigste Symptom dieses Problems besteht darin, dass die meisten Aufgaben-Scans länger dauern. Dies verursacht eine höhere Latenz bei diesen Scans. SQL Server priorisiert einige Aufgaben-Scans, sodass einige Aufgaben eine normale Latenz aufweisen. Überprüfen Sie die Metrik `CDCLatencySource` für alle Ihre Aufgaben, um dieses Problem zu beheben. Wenn einige der Aufgaben einen zunehmenden Wert für `CDCLatencySource` und andere einen niedrigen Wert für `CDCLatencySource` aufweisen, ist es wahrscheinlich, dass SQL Server Ihre DMS-Lesevorgänge für einige Ihrer Aufgaben drosselt.

Wenn SQL Server Ihre Aufgabenlesevorgänge während CDC drosselt, fassen Sie Ihre Aufgaben zusammen, um die Anzahl der DMS-Scans zu minimieren. Die maximale Anzahl von Aufgaben, die eine Verbindung mit Ihrer Quelldatenbank herstellen können, ohne dass Konflikte entstehen, hängt von Faktoren wie der Kapazität der Quelldatenbank, der Wachstumsrate des Transaktionsprotokolls oder der Anzahl der Tabellen ab. Testen Sie die Replikation in einer Testumgebung, die Ihrer Produktionsumgebung ähnelt, um die ideale Anzahl von Aufgaben für Ihr Replikationsszenario zu ermitteln.

Fehlersuche bei Problemen mit der Ziellatenz

Dieser Abschnitt enthält Szenarien, die zur Ziellatenz beitragen können.

Themen

- [Probleme bei der Indizierung](#)
- [SORTER-Meldung im Aufgabenprotokoll](#)
- [Sperrungen von Datenbanken](#)
- [Langsame LOB-Lookups](#)
- [Multi-AZ, Prüfungsprotokollierung und Backups](#)

Probleme bei der Indizierung

Während der CDC-Phase repliziert AWS DMS Änderungen an der Quelle, indem DML-Anweisungen (Insert, Update und Delete) auf dem Ziel ausgeführt werden. Bei heterogenen Migrationen mit DMS

können Unterschiede bei den Indexoptimierungen für die Quelle und das Ziel dazu führen, dass das Schreiben in das Ziel länger dauert. Dies führt zu Problemen mit der Ziellatenz und -leistung.

Gehen Sie wie folgt vor, um Probleme bei der Indizierung zu beheben: Die Verfahren unterscheiden sich je nach Datenbank-Engine.

- Überwachen Sie die Abfragezeit für Ihre Zieldatenbank. Ein Vergleich der Abfrageausführungszeit für Ziel und Quelle kann Aufschluss darüber geben, welche Indizes optimiert werden müssen.
- Aktivieren Sie die Protokollierung für langsam laufende Abfragen.

Gehen Sie wie folgt vor, um Probleme bei der Indizierung für lange laufende Replikationen zu beheben:

- Passen Sie die Indizes Ihrer Quell- und Zieldatenbanken so an, dass die Abfrageausführungszeit für die Quelle und das Ziel ähnlich ist.
- Vergleichen Sie die in DML-Abfragen verwendeten sekundären Indizes für die Quelle und das Ziel. Stellen Sie sicher, dass die DML-Leistung auf dem Ziel mit der DML-Leistung auf der Quelle vergleichbar oder besser als diese ist.

Beachten Sie, dass das Verfahren zur Optimierung von Indizes spezifisch für Ihre Datenbank-Engine ist. Es gibt kein DMS-Feature zur Optimierung von Quell- und Zielindizes.

SORTER-Meldung im Aufgabenprotokoll

Wenn ein Zielendpunkt mit dem Volumen an Änderungen, die AWS DMS in ihn schreibt, nicht Schritt halten kann, speichert die Aufgabe die Änderungen vorläufig in der Replikations-Instance. Wenn der Cache einen internen Schwellenwert überschreitet, liest die Aufgabe keine weiteren Änderungen aus der Quelle. Auf diese Weise verhindert DMS, dass der Speicherplatz auf der Replikations-Instance knapp wird oder dass die Aufgabe beim Lesen einer großen Menge ausstehender Ereignisse hängen bleibt.

Um dieses Problem zu beheben, überprüfen Sie die CloudWatch Protokolle auf eine Meldung, die einer der folgenden ähnelt:

```
[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)
```

Wenn Ihre Protokolle eine Meldung enthalten, die der ersten Meldung ähnelt, deaktivieren Sie die Ablaufprotokollierung für die Aufgabe und erhöhen Sie den Speicher der Replikations-Instance. Informationen zum Erhöhen des Speichers der Replikations-Instance finden Sie unter [Ändern einer Replikations-Instance](#).

Wenn Ihre Protokolle eine Meldung enthalten, die der zweiten Meldung ähnelt, gehen Sie wie folgt vor:

- Verschieben Sie Tabellen mit zahlreichen Transaktionen oder lang laufenden DML-Operationen in eine separate Aufgabe, wenn sie keine Abhängigkeiten von anderen Tabellen in der Aufgabe aufweisen.
- Erhöhen Sie die Einstellungen `MemoryLimitTotal` und `MemoryKeepTime`, um die Transaktion für einen längeren Zeitraum im Speicher zu halten. Dies hilft nicht, wenn die Latenz dauerhaft ist, aber es kann dazu beitragen, die Latenz bei kurzen Ausbrüchen des Transaktionsvolumens gering zu halten. Weitere Informationen zu diesen Aufgabeneinstellungen finden Sie unter [Einstellungen für die Optimierung der Verarbeitung von Änderungen](#).
- Prüfen Sie, ob Sie die Stapelanwendung für Ihre Transaktion verwenden können, indem Sie `BatchApplyEnabled` auf `true` setzen. Informationen zur Einstellung `BatchApplyEnabled` finden Sie unter [Ziel-Metadaten-Aufgabeneinstellungen](#).

Sperren von Datenbanken

Wenn eine Anwendung auf eine Datenbank zugreift, die AWS DMS als Replikationsziel verwendet, sperrt die Anwendung möglicherweise eine Tabelle, auf die DMS zuzugreifen versucht. Dadurch entsteht ein Sperrkonflikt. Da DMS Änderungen in der Reihenfolge in die Zieldatenbank schreibt, in der sie in der Quelle aufgetreten sind, führen Verzögerungen beim Schreiben in eine Tabelle aufgrund von Sperrkonflikten zu Verzögerungen beim Schreiben in alle Tabellen.

Fragen Sie zum Beheben dieses Problems die Zieldatenbank ab, um zu überprüfen, ob ein Sperrkonflikt DMS-Schreibtransaktionen blockiert. Wenn die Zieldatenbank DMS-Schreibtransaktionen blockiert, führen Sie einen oder mehrere der folgenden Schritte aus:

- Strukturieren Sie Ihre Abfragen neu, sodass Änderungen häufiger freigeschaltet werden.
- Ändern Sie Ihre Einstellungen für das Sperr-Timeout.
- Partitionieren Sie Ihre Tabellen, um Sperrkonflikte zu minimieren.

Beachten Sie, dass das Verfahren zur Optimierung von Sperrkonflikten spezifisch für Ihre Datenbank-Engine ist. Es gibt kein DMS-Feature zur Optimierung von Sperrkonflikten.

Langsame LOB-Lookups

Wenn AWS DMS eine Spalte mit großen Objekten (Large Object, LOB) repliziert, wird ein Lookup in der Quelle durchgeführt, kurz bevor Änderungen in das Ziel geschrieben werden. Dieser Lookup verursacht in der Regel keine Latenz auf dem Ziel. Wenn die Quelldatenbank den Lookup jedoch aufgrund von Sperren verzögert, kann dies zu einem Anstieg der Ziellatenz führen.

Dieses Problem ist in der Regel schwer zu diagnostizieren. Aktivieren Sie das detaillierte Debugging in den Aufgabenprotokollen und vergleichen Sie die Zeitstempel der LOB-Lookup-Aufrufe in DMS, um dieses Problem zu beheben. Informationen zur Aktivierung des detaillierten Debuggings finden Sie unter [Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen](#).

Versuchen Sie Folgendes, um dieses Problem zu beheben:

- Verbessern Sie die Leistung der SELECT-Abfrage in der Quelldatenbank.
- Passen Sie die DMS-LOB-Einstellungen an. Informationen zum Anpassen der LOB-Einstellungen finden Sie unter [Migrieren von Binary Large Objects \(LOBs\)](#).

Multi-AZ, Prüfungsprotokollierung und Backups

Bei Amazon-RDS-Zielen kann sich die Ziellatenz in den folgenden Fällen erhöhen:

- Sicherungen
- Nach der Aktivierung mehrerer Availability Zones (Multi-AZ)
- Nach der Aktivierung der Datenbankprotokollierung, wie beispielsweise Prüfungs- oder Slow-Query-Protokolle.

Diese Probleme sind in der Regel schwer zu diagnostizieren. Überwachen Sie die Latenz auf periodische Spitzenwerte während der Wartungsfenster von Amazon RDS oder bei hoher Datenbanklast, um diese Probleme zu beheben.

Versuchen Sie Folgendes, um diese Probleme zu beheben:

- Wenn möglich, deaktivieren Sie während einer kurzfristigen Migration Multi-AZ, Backups oder die Protokollierung.
- Verschieben Sie Ihre Wartungsfenster auf Zeiträume mit geringer Aktivität.

Arbeiten mit Diagnoseunterstützungsskripts in AWS DMS

Wenn Sie bei der Arbeit mit AWS DMS auf ein Problem stoßen, benötigt Ihr Support-Techniker möglicherweise weitere Informationen zu Ihrer Quell- oder Zieldatenbank. Wir möchten sicherstellen, dass der AWS-Support so viele der erforderlichen Informationen wie möglich in kürzester Zeit erhält. Aus diesem Grund haben wir Skripts entwickelt, um diese Informationen für mehrere der wichtigsten relationalen Datenbank-Engines abzufragen.

Wenn ein Unterstützungsskript für Ihre Datenbank verfügbar ist, können Sie es über den Link im entsprechenden Skriptthema im folgenden Abschnitt herunterladen. Nachdem Sie das Skript überprüft haben (wie nachfolgend beschrieben), können Sie es gemäß dem im Skriptthema beschriebenen Verfahren ausführen. Wenn die Ausführung des Skripts abgeschlossen ist, können Sie seine Ausgabe in Ihren AWS-Support-Fall hochladen (ebenfalls unten beschrieben).

Bevor Sie das Skript ausführen, können Sie alle Fehler erkennen, die möglicherweise beim Herunterladen oder Speichern des Unterstützungsskripts aufgetreten sind. Vergleichen Sie dazu die Prüfsumme für die Skriptdatei mit einem von AWS bereitgestellten Wert. AWS verwendet den SHA256-Algorithmus für die Prüfsumme.

So prüfen Sie die Unterstützungsskriptdatei anhand einer Prüfsumme

1. Öffnen Sie die neueste Prüfsummendatei, die zur Prüfung dieser Unterstützungsskripts zur Verfügung gestellt wurde, unter <https://d2pwp9zz55emqw.cloudfront.net/sha256Check.txt>. Beispielsweise könnte die Datei einen Inhalt wie den folgenden haben.

```
MYSQL dfafd0d511477c699f96c64693ad0b1547d47e74d5c5f2f2025b790b1422e3c8
ORACLE 6c41ebcfc99518cfa8a10cb2ce8943b153b2cc7049117183d0b5de3d551bc312
POSTGRES 6ccd274863d14f6f3146fbd8bba43f2d8d4c6a4c25380d7b41c71883aa4f9790
SQL_SERVER 971a6f2c46aec8d083d2b3b6549b1e9990af3a15fe4b922e319f4fdd358debe7
```

2. Führen Sie den SHA256-Validierungsbefehl für Ihr Betriebssystem in dem Verzeichnis mit der Unterstützungsdatei aus. Auf dem macOS-Betriebssystem können Sie beispielsweise den folgenden Befehl in einem Oracle-Unterstützungsskript ausführen, das später in diesem Thema beschrieben wird.

```
shasum -a 256 awsdms_support_collector_oracle.sql
```

3. Vergleichen Sie die Ergebnisse des Befehls mit dem Wert, der in der zuletzt geöffneten sha256Check.txt-Datei angezeigt wird. Die beiden Werte sollten übereinstimmen. Wenn dies

nicht der Fall ist, informieren Sie Ihren Support-Techniker darüber, um zu erfahren, wie Sie eine saubere Unterstützungsskriptdatei erhalten können.

Wenn Sie über eine saubere Unterstützungsskriptdatei verfügen, stellen Sie vor der Ausführung des Skripts sicher, dass Sie die SQL sowohl aus Leistungs- als auch aus Sicherheitsperspektive gelesen und verstanden haben. Wenn Sie mit der Ausführung von SQL in diesem Skript nicht vertraut sind, können Sie das problematische SQL auskommentieren oder entfernen. Sie können sich auch an Ihren Support-Techniker wenden, um mögliche Behelfslösungen zu besprechen.

Nach erfolgreichem Abschluss und sofern nicht anders angegeben, gibt das Skript die Ausgabe in einem lesbaren HTML-Format zurück. Das Skript ist so konzipiert, dass alle Daten oder Sicherheitsdetails, die Ihr Unternehmen gefährden könnten, aus diesem HTML-Code ausgeschlossen werden. Es nimmt auch keine Änderungen an Ihrer Datenbank oder ihrer Umgebung vor. Wenn Sie jedoch Informationen im HTML-Code finden, die Sie nicht weitergeben möchten, können Sie die Probleminformationen entfernen, bevor Sie den HTML-Code hochladen. Wenn der HTML-Code akzeptabel ist, laden Sie ihn mithilfe der Anlagen in den Falldetails Ihres Support-Falls hoch.

In jedem der folgenden Themen werden die für eine unterstützte AWS DMS-Datenbank verfügbaren Skripts und deren Ausführung beschrieben. Ihr Support-Techniker wird Sie zu einem bestimmten Skript weiterleiten, das im Folgenden dokumentiert ist.

Themen

- [Oracle-Diagnoseunterstützungsskripts](#)
- [SQL-Server-Diagnoseunterstützungsskripts](#)
- [Diagnoseunterstützungsskripts für MySQL-kompatible Datenbanken](#)
- [PostgreSQL-Diagnoseunterstützungsskripts](#)

Oracle-Diagnoseunterstützungsskripts

Im Folgenden finden Sie die Diagnoseunterstützungsskripts, die für die Analyse einer On-Premises- oder einer Amazon-RDS-für-Oracle-Datenbank in Ihrer AWS DMS-Migrationskonfiguration verfügbar sind. Diese Skripts funktionieren entweder mit einem Quell- oder Zielendpunkt. Die Skripts wurden alle so geschrieben, dass sie im Befehlszeilen-Dienstprogramm SQL*Plus ausgeführt werden können. Weitere Informationen zur Verwendung dieses Dienstprogramms finden Sie unter [Verwendung von SQL Command Line](#) in der Oracle-Dokumentation.

Bevor Sie das Skript ausführen, stellen Sie sicher, dass das von Ihnen verwendete Benutzerkonto über die erforderlichen Berechtigungen für den Zugriff auf Ihre Oracle-Datenbank verfügt. Bei den angezeigten Berechtigungseinstellungen wird davon ausgegangen, dass ein Benutzer wie folgt erstellt wurde.

```
CREATE USER script_user IDENTIFIED BY password;
```

Legen Sie für eine On-Premises-Datenbank die Mindestberechtigungen wie folgt für *script_user* fest.

```
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;
```

Legen Sie für eine Amazon-RDS-Datenbank die Mindestberechtigungen wie folgt fest.

```
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');
```

Im Folgenden finden Sie Beschreibungen, wie Sie jedes für Oracle verfügbare SQL*Plus-Unterstützungsskript herunterladen, überprüfen und ausführen können. Dort erfahren Sie auch, wie Sie die Ausgabe überprüfen und in Ihren AWS-Support-Fall hochladen können.

Themen

- [Das Skript awsdms_support_collector_oracle.sql](#)

Das Skript awsdms_support_collector_oracle.sql

Laden Sie das [awsdms_support_collector_oracle.sql](#)-Skript herunter.

Dieses Skript erfasst Informationen über Ihre Oracle-Datenbankkonfiguration. Denken Sie daran, die Prüfsumme des Skripts zu überprüfen. Wenn die Prüfsumme verifiziert wurde, überprüfen Sie den SQL-Code in dem Skript, um den Code auszukommentieren, dessen Ausführung Sie nicht wünschen. Wenn Sie mit der Integrität und dem Inhalt des Skripts zufrieden sind, können Sie es ausführen.

So führen Sie das Skript aus und laden die Ergebnisse in Ihren Support-Fall hoch

1. Führen Sie das Skript in Ihrer Datenbankumgebung mit der folgenden SQL*Plus-Befehlszeile aus.

```
SQL> @awsdms_support_collector_oracle.sql
```

<result>

Das Skript zeigt eine kurze Beschreibung und eine Aufforderung an, die Ausführung entweder fortzusetzen oder abubrechen. Betätigen Sie zum Fortfahren die Eingabetaste.

</result>

2. Geben Sie bei der folgenden Eingabeaufforderung nur den Namen eines der Schemata ein, die Sie migrieren möchten.
3. Geben Sie bei der folgenden Eingabeaufforderung den Namen des Benutzers (*script_user*) ein, den Sie für die Verbindung mit der Datenbank definiert haben.
4. Geben Sie bei der folgenden Eingabeaufforderung die Anzahl der Tage an Daten ein, die Sie untersuchen möchten, oder akzeptieren Sie die Standardeinstellung. Das Skript erfasst dann die angegebenen Daten aus Ihrer Datenbank.

<result>

Nach Abschluss des Skripts wird der Name der HTML-Ausgabedatei angezeigt, beispielsweise `dms_support_oracle-2020-06-22-13-20-39-0RCL.html`. Das Skript speichert diese Datei in Ihrem Arbeitsverzeichnis.

`</result>`

- Überprüfen Sie diese HTML-Datei und entfernen Sie alle Informationen, die Sie nicht weitergeben möchten. Wenn Sie den HTML-Code weitergeben können, laden Sie die Datei in Ihren AWS-Support-Fall hoch. Weitere Informationen zum Hochladen dieser Datei finden Sie unter [Arbeiten mit Diagnoseunterstützungsskripten in AWS DMS](#).

SQL-Server-Diagnoseunterstützungsskripts

Im Folgenden finden Sie die Diagnoseunterstützungsskripts, die für die Analyse einer On-Premises-Datenbank oder einer Amazon-RDS-für-SQL-Server-Datenbank in Ihrer AWS DMS-Migrationskonfiguration verfügbar sind. Diese Skripts funktionieren entweder mit einem Quell- oder Zielpunkt. Führen Sie diese Skript für eine On-Premises-Datenbank im Befehlszeilen-Dienstprogramm `sqlcmd` aus. Weitere Informationen zur Verwendung dieses Dienstprogramms finden Sie unter [sqlcmd – Verwendung des Hilfsprogramms](#) in der Microsoft-Dokumentation.

Bei einer Amazon-RDS-Datenbank können Sie mit dem Befehlszeilen-Dienstprogramm `sqlcmd` keine Verbindung herstellen. Führen Sie diese Skripts stattdessen mit einem beliebigen Client-Tool aus, das eine Verbindung zu Amazon RDS SQL Server herstellt.

Bevor Sie das Skript ausführen, stellen Sie sicher, dass das von Ihnen verwendete Benutzerkonto über die erforderlichen Berechtigungen für den Zugriff auf Ihre SQL-Server-Datenbank verfügt. Sowohl für eine On-Premises- als auch für eine Amazon-RDS-Datenbank können Sie dieselben Berechtigungen verwenden, die Sie für den Zugriff auf Ihre SQL-Server-Datenbank ohne die `SysAdmin`-Rolle verwenden.

Themen

- [Einrichtung von Mindestberechtigungen für eine On-Premises-SQL-Server-Datenbank](#)
- [Einrichtung von Mindestberechtigungen für eine Amazon-RDS-SQL-Server-Datenbank](#)
- [Einrichtung der fortlaufenden Replikation auf einem eigenständigen SQL Server: Ohne Sysadmin-Rolle](#)
- [Einrichtung einer laufenden Replikation auf einem SQL-Server in einer Availability-Group-Umgebung: Ohne Sysadmin-Rolle](#)
- [SQL-Server-Unterstützungsskripts](#)

Einrichtung von Mindestberechtigungen für eine On-Premises-SQL-Server-Datenbank

So richten Sie Mindestberechtigungen für eine On-Premises-SQL-Server-Datenbank ein

1. Erstellen Sie unter Verwendung von SQL Server Management Studio (SSMS) ein neues SQL-Server-Konto mit Passwort-Authentifizierung, zum Beispiel *on-prem-user*.
2. Wählen Sie im Abschnitt Benutzerzuweisungen von SSMS die Datenbanken MSDB und MASTER aus (wodurch öffentliche Berechtigungen erteilt werden) und weisen Sie der Datenbank, die Sie für die fortlaufende Replikation verwenden möchten, die Rolle DB_OWNER zu.
3. Öffnen Sie das Kontextmenü (rechte Maustaste) für das neue Konto, wählen Sie Sicherheit aus, um ausdrücklich die Connect SQL-Berechtigung zu erteilen.
4. Führen Sie die folgenden Befehle zum Erteilen der Berechtigung aus.

```
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;
```

Einrichtung von Mindestberechtigungen für eine Amazon-RDS-SQL-Server-Datenbank

So richten Sie Mindestberechtigungen für eine Amazon-RDS-SQL-Server-Datenbank ein

1. Erstellen Sie unter Verwendung von SQL Server Management Studio (SSMS) ein neues SQL-Server-Konto mit Passwort-Authentifizierung, zum Beispiel *rds-user*.
2. Wählen Sie im Abschnitt Benutzerzuweisungen von SSMS die Datenbank MSDB aus (die öffentliche Zugriffsrechte gewährt) und weisen Sie die DB_OWNER-Rolle der Datenbank zu, in der Sie das Skript ausführen möchten.
3. Öffnen Sie das Kontextmenü (rechte Maustaste) für das neue Konto, wählen Sie Sicherheit aus, um ausdrücklich die Connect SQL-Berechtigung zu erteilen.
4. Führen Sie die folgenden Befehle zum Erteilen der Berechtigung aus.

```
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;
```

Einrichtung der fortlaufenden Replikation auf einem eigenständigen SQL Server: Ohne Sysadmin-Rolle

Dieser Abschnitt beschreibt die Einrichtung der fortlaufenden Replikation für eine eigenständige SQL-Server-Datenbankquelle, für die das Benutzerkonto keine Sysadmin-Berechtigungen benötigt.

Note

Nachdem Sie die Schritte in diesem Abschnitt ausgeführt haben, hat der DMS-Benutzer, der kein Systemadministrator ist, die Berechtigung, um Folgendes zu tun:

- Lesen der Änderungen aus der Protokolldatei für Online-Transaktionen
- Festplattenzugriff zum Lesen von Änderungen aus Transaktionsprotokoll-Backup-Dateien
- Hinzufügen oder Ändern der von DMS verwendeten Publikation
- Hinzufügen von Artikeln zu der Publikation

1. Richten Sie Microsoft SQL Server für die Replikation ein, wie unter [Erfassen von Datenänderungen für selbstverwaltete SQL-Server-Quellen \(On-Premises oder in Amazon EC2\)](#) beschrieben.
2. Aktivieren Sie MS-REPLICATION in der Quelldatenbank. Dies kann entweder manuell oder durch einmaliges Ausführen der Aufgabe als Sysadmin-Benutzer erfolgen.
3. Erstellen Sie das awsdms-Schema in der Quelldatenbank mit dem folgenden Skript:

```
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

@dmlm 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 @dmlm_pos integer;

declare @dmlm_len integer;

set @dmlm_len = len(@dmlm);

while (charindex(@dmlm,@plist)>0)

begin

set @dmlm_pos = charindex(@dmlm,@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. Erstellen Sie die [awsdms].[rtm_dump_dblog]-Prozedur in der Master-Datenbank mit dem folgenden Skript:

```
use [MASTER]

go

if (object_id('[awsdms].[rtm_dump_dblog]','P')) is not null drop procedure
[awsdms].[rtm_dump_dblog];
go

set ansi_nulls on
go

set quoted_identifier on
GO

CREATE procedure [awsdms].[rtm_dump_dblog]

(
```

```
@start_lsn varchar(32),

@seqno integer,

@filename varchar(260),

@partition_list varchar(8000), – A comma delimited list: P1,P2,... Pn

@programmed_filtering integer,

@minPartition bigint,

@maxPartition bigint

)

as begin

declare @start_lsn_cmp varchar(32); – Stands against the GT comparator

SET NOCOUNT ON – – Disable "rows affected display"

set @start_lsn_cmp = @start_lsn;

if (@start_lsn_cmp) is null

set @start_lsn_cmp = '00000000:00000000:0000';

if (@partition_list is null)

begin

RAISERROR ('Null partition list waspassed',16,1);

return

end

if (@start_lsn) is not null

set @start_lsn = '0x'+@start_lsn;

if (@programmed_filtering=0)
```

```
SELECT

[Current LSN],

[operation],

[Context],

[Transaction ID],

[Transaction Name],

[Begin Time],

[End Time],

[Flag Bits],

[PartitionID],

[Page ID],

[Slot ID],

[RowLog Contents 0],

[Log Record],

[RowLog Contents 1]

FROM

fn_dump_dblog (

@start_lsn, NULL, N'DISK', @seqno, @filename,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,
```

```
default, default, default, default, default, default, default,
default, default, default, default, default, default, default,
default, default, default, default, default, default, default,
default, default, default, default, default, default, default,
default, default, default, default, default, default, default,
default, default, default, default, default, default, default)

where [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp collate
    SQL_Latin1_General_CP1_CI_AS

and

(

( [operation] in ('LOP_BEGIN_XACT', 'LOP_COMMIT_XACT', 'LOP_ABORT_XACT') )

or

( [operation] in ('LOP_INSERT_ROWS', 'LOP_DELETE_ROWS', 'LOP_MODIFY_ROW')

and

( ( [context] in ('LCX_HEAP', 'LCX_CLUSTERED', 'LCX_MARK_AS_GHOST') ) or ([context] =
    'LCX_TEXT_MIX' and (datalength([RowLog Contents 0]) in (0,1))))

and [PartitionID] in ( select * from master.awsdms.split_partition_list
    (@partition_list, ','))

)

or

([operation] = 'LOP_HOBT_DDL')

)

else
```

```
SELECT

[Current LSN],

[operation],

[Context],

[Transaction ID],

[Transaction Name],

[Begin Time],

[End Time],

[Flag Bits],

[PartitionID],

[Page ID],

[Slot ID],

[RowLog Contents 0],

[Log Record],

[RowLog Contents 1] – After Image

FROM

fn_dump_dblog (

@start_lsn, NULL, N'DISK', @seqno, @filename,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,
```

```
default, default, default, default, default, default, default,
default, default, default, default, default, default, default,
default, default, default, default, default, default, default,
default, default, default, default, default, default, default,
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. Erstellen Sie das Zertifikat in der Master-Datenbank mit dem folgenden Skript:

```
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. Erstellen Sie die Anmeldung von dem Zertifikat mit dem folgenden Skript:

```
Use [master]
```

```
Go
```

```
CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE  
[awsdms_rtm_dump_dblog_cert];
```

7. Fügen Sie die Anmeldung mithilfe des folgenden Skripts zur Sysadmin-Serverrolle hinzu:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```


8. Fügen Sie die Signatur zu [Master].[awsdms].[rtm_dump_dblog] unter Verwendung des Zertifikats mit dem folgenden Skript hinzu:

```
Use [master]
```

```
GO
```

```
ADD SIGNATURE
```

```
TO [master].[awsdms].[rtm_dump_dblog] BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]  
WITH PASSWORD = '@5trongpassword';
```

 Note

Wenn Sie die gespeicherte Prozedur neu erstellen, müssen Sie die Signatur erneut hinzufügen.

9. Erstellen Sie [awsdms].[rtm_position_1st_timestamp] in der Master-Datenbank mithilfe des folgenden Skripts:

```

use [master]
if object_id('[awsdms].[rtm_position_1st_timestamp]','P') is not null
DROP PROCEDURE [awsdms].[rtm_position_1st_timestamp];
go
create procedure [awsdms].[rtm_position_1st_timestamp]
(
  @dbname          sysname,          -- Database name
  @seqno           integer,          -- Backup set sequence/position number
within file
  @filename         varchar(260),    -- The backup filename
  @1stTimeStamp    varchar(40)      -- The timestamp to position by
)
as begin

SET NOCOUNT ON          -- Disable "rows affected display"

declare @firstMatching table
(
  cLsn varchar(32),
  bTim datetime
)

declare @sql nvarchar(4000)
declare @nl          char(2)
declare @tb          char(2)
declare @fnameVar   nvarchar(254) = 'NULL'

set @nl = char(10); -- New line
set @tb = char(9)  -- Tab separator

if (@filename is not null)
set @fnameVar = '''+@filename +''''

set @sql='use ['+@dbname+'];'+@nl+
'select top 1 [Current LSN],[Begin Time]'+@nl+
'FROM fn_dump_dblog (NULL, NULL, NULL, '+ cast(@seqno as varchar(10))+','+
@fnameVar+', '+@nl+
@tb+'default, default, default, default, default, default, default, '+@nl+
@tb+'default, default, default, default, default, default, default, '+@nl+
@tb+'default, default, default, default, default, default, default, '+@nl+

```



```

@tb+'default, default, default, default, default, default, default, default,'+@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,'+@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,'+@nl+
'where operation='LOP_BEGIN_XACT'' '+@nl+
'and [Begin Time]>= cast('+'''+@1stTimeStamp+'''+ ' as datetime)'+@nl

--print @sql
delete from @firstMatching
insert into @firstMatching exec sp_executesql @sql -- Get them all

select top 1 cLsn as [matching LSN],convert(varchar,bTim,121) as [matching
Timestamp] from @firstMatching;

SET NOCOUNT OFF -- Re-enable "rows affected display"

end
GO

```

10. Erstellen Sie das Zertifikat in der Master-Datenbank mit dem folgenden Skript:

```

Use [master]
Go
CREATE CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
ENCRYPTION BY PASSWORD = '@5trongpassword'
WITH SUBJECT = N'Certificate for FN_POSITION_1st_TIMESTAMP Permissions';

```

11. Erstellen Sie die Anmeldung von dem Zertifikat mit dem folgenden Skript:

```

Use [master]
Go
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
[awsdms_rtm_position_1st_timestamp_cert];

```

12. Fügen Sie die Anmeldedaten mithilfe des folgenden Skripts zur sysadmin-Rolle hinzu:

```

ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_position_1st_timestamp_login];

```


13. Fügen Sie die Signatur zu [master].[awsdms].[rtm_position_1st_timestamp] unter Verwendung des Zertifikats mit dem folgenden Skript hinzu:

```
Use [master]
GO
ADD SIGNATURE
TO [master].[awsdms].[rtm_position_1st_timestamp]
BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
WITH PASSWORD = '@5trongpassword';
```

14. Gewähren Sie dem DMS-Benutzer mithilfe des folgenden Skripts Ausführungszugriff auf die neue gespeicherte Prozedur:

```
use master
go
GRANT execute on [awsdms].[rtm_position_1st_timestamp] to dms_user;
```

15. Erstellen Sie in jeder der folgenden Datenbanken einen Benutzer mit den folgenden Berechtigungen und Rollen:

 Note

Sie sollten das Benutzerkonto dmsnosysadmin mit derselben SID für jedes Replikat erstellen. Die folgende SQL-Abfrage kann dabei helfen, den SID-Wert des dmsnosysadmin-Kontos auf jedem Replikat zu überprüfen. Weitere Informationen zum Erstellen eines Benutzers finden Sie unter [BENUTZER ERSTELLEN \(Transact-SQL\)](#) in der [Microsoft-SQL-Server-Dokumentation](#). Weitere Informationen zum Erstellen von SQL-Benutzerkonten für die Azure-SQL-Datenbank finden Sie unter [Aktive Georeplikation](#).

```
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]
```

Führen Sie den folgenden Befehl für die Quelldatenbank aus:

```
EXEC sp_addrolemember N'db_owner', N'DMS_user'
use Source_DB
go
```

16. Fügen Sie abschließend dem SQL-Server-Endpunkt ein zusätzliches Verbindungsattribut (ECA) hinzu:

```
enableNonSysadminWrapper=true;
```

Einrichtung einer laufenden Replikation auf einem SQL-Server in einer Availability-Group-Umgebung: Ohne Sysadmin-Rolle

Dieser Abschnitt beschreibt die Einrichtung der fortlaufenden Replikation für eine eigenständige SQL-Server-Datenbankquelle, für die das Benutzerkonto keine sysadmin-Berechtigungen benötigt.


Note

Nachdem Sie die Schritte in diesem Abschnitt ausgeführt haben, hat der DMS-Benutzer, der kein Systemadministrator ist, die Berechtigung, um Folgendes zu tun:

- Lesen der Änderungen aus der Protokolldatei für Online-Transaktionen
- Festplattenzugriff zum Lesen von Änderungen aus Transaktionsprotokoll-Backup-Dateien
- Hinzufügen oder Ändern der von DMS verwendeten Publikation
- Hinzufügen von Artikeln zu der Publikation

So richten Sie die fortlaufende Replikation in einer Availability-Group-Umgebung ein, ohne den Sysadmin-Benutzer zu verwenden

1. Richten Sie Microsoft SQL Server für die Replikation ein, wie unter [Erfassen von Datenänderungen für selbstverwaltete SQL-Server-Quellen \(On-Premises oder in Amazon EC2\)](#) beschrieben.
2. Aktivieren Sie MS-REPLICATION in der Quelldatenbank. Dies kann entweder manuell oder durch einmaliges Ausführen der Aufgabe als Sysadmin-Benutzer erfolgen.

 Note

Sie sollten den MS-REPLICATION-Verteiler entweder lokal oder so konfigurieren, dass Benutzer, die keine Systemadministratoren sind, über den zugehörigen Verbindungsserver darauf zugreifen können.

3. Wenn die Option Nur sp_repldone innerhalb eines einzelnen Aufgaben-Endpunkts verwenden aktiviert ist, beenden Sie den MS-REPLICATION-Protokollleseauftrag.
4. Führen Sie auf jedem Replikat die folgenden Schritte aus:

1. Erstellen Sie das [awsdms][awsdms]-Schema in der Master-Datenbank:

```
CREATE SCHEMA [awsdms]
```

2. Erstellen Sie die [awsdms].[split_partition_list]-Tabellenwertfunktion in der Master-Datenbank:

```
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. Erstellen Sie die [awsdms].[rtm_dump_dblog]-Prozedur in der Master-Datenbank:

```

USE [MASTER]
GO

IF (object_id('[awsdms].[rtm_dump_dblog]','P')) is not null
  DROP PROCEDURE [awsdms].[rtm_dump_dblog];
GO

SET ansi_nulls on
GO

SET quoted_identifier on
GO

```

```
CREATE PROCEDURE [awsdms].[rtm_dump_dblog]
(
    @start_lsn          varchar(32),
    @seqno              integer,
    @filename           varchar(260),
    @partition_list    varchar(8000), -- A comma delimited list: P1,P2,... Pn
    @programmed_filtering integer,
    @minPartition      bigint,
    @maxPartition      bigint
)
AS
BEGIN

    DECLARE @start_lsn_cmp varchar(32); -- Stands against the GT comparator

    SET NOCOUNT ON -- Disable "rows affected display"

    SET @start_lsn_cmp = @start_lsn;
    IF (@start_lsn_cmp) is null
        SET @start_lsn_cmp = '00000000:00000000:0000';

    IF (@partition_list is null)
        BEGIN
            RAISERROR ('Null partition list was passed',16,1);
            return
            --set @partition_list = '0,';    -- A dummy which is never matched
        END

    IF (@start_lsn) is not null
        SET @start_lsn = '0x'+@start_lsn;

    IF (@programmed_filtering=0)
        SELECT
            [Current LSN],
            [operation],
            [Context],
            [Transaction ID],
            [Transaction Name],
            [Begin Time],
            [End Time],
            [Flag Bits],
            [PartitionID],
            [Page ID],
            [Slot ID],
```

```

[RowLog Contents 0],
[Log Record],
[RowLog Contents 1] -- After Image
FROM
fn_dump_dblog (
  @start_lsn, NULL, N'DISK', @seqno, @filename,
  default, default, default, default, default, default, default, default,
  default, default, default, default, default, default, default, default,
  default, default, default, default, default, default, default, default,
  default, default, default, default, default, default, default, default,
  default, default, default, default, default, default, default, default,
  default, default, default, default, default, default, default, default,
  default, default, default, default, default, default, default, default,
  default, default, default, default, default, default, default, 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.awsdfs.split_partition_list
(@partition_list,',') )
  )
  OR
    ([operation] = 'LOP_HOBT_DDL')
)
ELSE
SELECT
  [Current LSN],
  [operation],
  [Context],
  [Transaction ID],
  [Transaction Name],
  [Begin Time],
  [End Time],
  [Flag Bits],

```

```

    [PartitionID],
    [Page ID],
    [Slot ID],
    [RowLog Contents 0],
    [Log Record],
    [RowLog Contents 1] -- After Image
FROM
    fn_dump_dblog (
        @start_lsn, NULL, N'DISK', @seqno, @filename,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default,
        default, default, default, default, default, default, default)
    WHERE [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp
    collate SQL_Latin1_General_CP1_CI_AS -- This aims for implementing FN_DBLOG
    based on GT comparator.
    AND
    (
        ( [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )
    OR
        ( [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')
        AND
            ( ( [context] in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') )
        or ([context] = 'LCX_TEXT_MIX') )
            AND ([PartitionID] is not null) and ([PartitionID] >= @minPartition and
            [PartitionID]<=@maxPartition)
        )
    OR
        ([operation] = 'LOP_HOBT_DDL')
    )
    SET NOCOUNT OFF -- Re-enable "rows affected display"
END
GO

```

4. Erstellen Sie ein Zertifikat in der Master-Datenbank:

```

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. Erstellen Sie eine Anmeldung aus dem Zertifikat:


```
USE [master]  
GO  
CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE  
[awsdms_rtm_dump_dblog_cert];
```

6. Fügen Sie den Anmeldenamen zur Sysadmin-Serverrolle hinzu:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

7. Fügen Sie die Signatur zur Prozedur [master].[awsdms].[rtm_dump_dblog] unter Verwendung des Zertifikats hinzu:

```
USE [master]  
GO  
  
ADD SIGNATURE  
TO [master].[awsdms].[rtm_dump_dblog]  
BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]  
WITH PASSWORD = '@hardpassword1';
```

 Note

Wenn Sie die gespeicherte Prozedur neu erstellen, müssen Sie die Signatur erneut hinzufügen.

8. Erstellen Sie die [awsdms].[rtm_position_1st_timestamp]-Prozedur in der Master-Datenbank:

```
USE [master]  
IF object_id('[awsdms].[rtm_position_1st_timestamp]','P') is not null  
DROP PROCEDURE [awsdms].[rtm_position_1st_timestamp];  
GO  
CREATE PROCEDURE [awsdms].[rtm_position_1st_timestamp]  
(  
@dbname  
sysname, -- Database name
```

```

    @seqno                integer,          -- Backup set sequence/position number
    within file
    @filename              varchar(260),    -- The backup filename
    @1stTimeStamp          varchar(40)     -- The timestamp to position by
)
AS
BEGIN
    SET NOCOUNT ON          -- Disable "rows affected display"

    DECLARE @firstMatching table
    (
        cLsn varchar(32),
        bTim datetime
    )
    DECLARE @sql nvarchar(4000)
    DECLARE @nl                char(2)
    DECLARE @tb                char(2)
    DECLARE @fnameVar          sysname = 'NULL'

    SET @nl = char(10); -- New line
    SET @tb = char(9)  -- Tab separator

    IF (@filename is not null)
        SET @fnameVar = '''+@filename +'''
    SET @filename = '''+@filename +'''
    SET @sql='use ['+@dbname+'];'+@nl+
        'SELECT TOP 1 [Current LSN],[Begin Time]'+@nl+
        'FROM fn_dump_dblog (NULL, NULL, NULL, '+ cast(@seqno as varchar(10))+','+
@filename +','+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        @tb+'default, default, default, default, default, default, default,'+@nl+
        'WHERE operation=''LOP_BEGIN_XACT'' '+@nl+
        'AND [Begin Time]>= cast(''+'''+@1stTimeStamp+'''+ as datetime)'+@nl

    --print @sql
    DELETE FROM @firstMatching
    INSERT INTO @firstMatching exec sp_executesql @sql -- Get them all

```

```
SELECT TOP 1 cLsn as [matching LSN],convert(varchar,bTim,121) AS[matching  
Timestamp] FROM @firstMatching;  
  
SET NOCOUNT OFF      -- Re-enable "rows affected display"  
  
END  
GO
```

9. Erstellen Sie ein Zertifikat in der Master-Datenbank:

```
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 Erstellen Sie eine Anmeldung aus dem Zertifikat:

```
USE [master]  
GO  
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE  
  [awsdms_rtm_position_1st_timestamp_cert];
```

11 Fügen Sie den Anmeldenamen zur Sysadmin-Serverrolle hinzu:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER  
  [awsdms_rtm_position_1st_timestamp_login];
```

12 Fügen Sie der [master].[awsdms].[rtm_position_1st_timestamp]-Prozedur mithilfe des Zertifikats die Signatur hinzu:

```
USE [master]  
GO  
ADD SIGNATURE  
  TO [master].[awsdms].[rtm_position_1st_timestamp]  
  BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]  
  WITH PASSWORD = '@hardpassword1';
```

Note

Wenn Sie die gespeicherte Prozedur neu erstellen, müssen Sie die Signatur erneut hinzufügen.

13 Erstellen Sie in jeder der folgenden Datenbanken einen Benutzer mit den folgenden Berechtigungen/Rollen:

Note

Sie sollten das Benutzerkonto `dmsnosysadmin` mit derselben SID für jedes Replikat erstellen. Die folgende SQL-Abfrage kann dabei helfen, den SID-Wert des `dmsnosysadmin`-Kontos auf jedem Replikat zu überprüfen. Weitere Informationen zum Erstellen eines Benutzers finden Sie unter [BENUTZER ERSTELLEN \(Transact-SQL\)](#) in der [Microsoft-SQL-Server-Dokumentation](#). Weitere Informationen zum Erstellen von SQL-Benutzerkonten für die Azure-SQL-Datenbank finden Sie unter [Aktive Georeplikation](#).

```
SELECT @@servername servername, name, sid, create_date, modify_date
FROM sys.server_principals
WHERE name = 'dmsnosysadmin';
```

14 Erteilen Sie für jedes Replikat Berechtigungen für die Master-Datenbank:

```
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 Erteilen Sie für jedes Replikat Berechtigungen für die msdb-Datenbank:

```
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 Fügen Sie die db_owner-Rolle zu dmsnosysadmin in der Quelldatenbank hinzu. Da die Datenbank synchronisiert ist, können Sie die Rolle nur dem primären Replikat hinzufügen.

```
use <source DB>
GO
EXEC sp_addrolemember N'db_owner', N'dmsnosysadmin'
```

SQL-Server-Unterstützungsskripts

In den folgenden Themen wird beschrieben, wie jedes für SQL Server verfügbare Unterstützungsskript heruntergeladen, überprüft und ausgeführt wird. Dort erfahren Sie auch, wie Sie die Skriptausgabe überprüfen und in Ihren AWS-Support-Fall hochladen können.

Themen

- [Das Skript awsdms_support_collector_sql_server.sql](#)

Das Skript awsdms_support_collector_sql_server.sql

Laden Sie das [awsdms_support_collector_sql_server.sql](#)-Skript herunter.

Note

Führen Sie dieses Skript zur Unterstützung der SQL-Server-Diagnose nur auf SQL Server 2014 und höheren Versionen aus.

Dieses Skript erfasst Informationen über Ihre SQL-Server-Datenbankkonfiguration. Denken Sie daran, die Prüfsumme des Skripts zu überprüfen. Wenn die Prüfsumme verifiziert wurde, überprüfen Sie den SQL-Code in dem Skript, um den Code auszukommentieren, dessen Ausführung Sie nicht

wünschen. Wenn Sie mit der Integrität und dem Inhalt des Skripts zufrieden sind, können Sie es ausführen.

So führen Sie das Skript für eine On-Premises-SQL-Server-Datenbank aus

1. Führen Sie das Skript mit der folgenden sqlcmd-Befehlszeile aus.

```
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
```

Zu den angegebenen sqlcmd-Befehlsparametern gehören unter anderem:

- -U – Name des Datenbankbenutzers.
 - -P – Passwort des Datenbankbenutzers.
 - -S – Name des SQL-Server-Datenbankservers.
 - -y – Maximale Breite der vom Hilfsprogramm sqlcmd ausgegebenen Spalten. Ein Wert von 0 gibt Spalten mit unbegrenzter Breite an.
 - -i – Pfad des auszuführenden Unterstützungsskripts, in diesem Fall `awsdms_support_collector_sql_server.sql`.
 - -o – Pfad der HTML-Ausgabedatei mit einem von Ihnen angegebenen Dateinamen, der die erfassten Datenbankkonfigurationsinformationen enthält.
 - -d – Name der SQL-Server-Datenbank.
2. Überprüfen Sie nach Abschluss des Skripts die HTML-Ausgabedatei und entfernen Sie alle Informationen, die Sie nicht weitergeben möchten. Wenn Sie den HTML-Code weitergeben können, laden Sie die Datei in Ihren AWS-Support-Fall hoch. Weitere Informationen zum Hochladen dieser Datei finden Sie unter [Arbeiten mit Diagnoseunterstützungsskripten in AWS DMS](#).

Mit Amazon RDS für SQL Server können Sie keine Verbindung mit dem Befehlszeilen-Dienstprogramm sqlcmd herstellen. Gehen Sie daher wie folgt vor.

So führen Sie das Skript für eine RDS-SQL-Server-Datenbank aus

1. Führen Sie das Skript mit einem beliebigen Client-Tool aus, mit dem Sie als Master-Benutzer eine Verbindung zu RDS SQL Server herstellen und die Ausgabe als HTML-Datei speichern können.

- Überprüfen Sie die Ausgabe-HTML-Datei und entfernen Sie alle Informationen, die Sie nicht weitergeben möchten. Wenn Sie den HTML-Code weitergeben können, laden Sie die Datei in Ihren AWS-Support-Fall hoch. Weitere Informationen zum Hochladen dieser Datei finden Sie unter [Arbeiten mit Diagnoseunterstützungsskripts in AWS DMS](#).

Diagnoseunterstützungsskripts für MySQL-kompatible Datenbanken

Im Folgenden finden Sie die Diagnoseunterstützungsskripts, die für die Analyse einer On-Premises-Datenbank oder einer Amazon-RDS-für-MySQL-kompatiblen Datenbank in Ihrer AWS DMS-Migrationskonfiguration verfügbar sind. Diese Skripts funktionieren entweder mit einem Quell- oder Zielpunkt. Die Skripts wurden alle so geschrieben, dass sie im Befehlszeilen-Dienstprogramm MySQL SQL ausgeführt werden können.

Informationen zum Installieren des MySQL-Clients finden Sie in der MySQL-Dokumentation unter [Installation der MySQL-Shell](#). Informationen zur Verwendung des MySQL-Clients finden Sie in der MySQL-Dokumentation unter [Verwendung von MySQL-Shell-Befehlen](#).

Bevor Sie das Skript ausführen, stellen Sie sicher, dass das von Ihnen verwendete Benutzerkonto über die erforderlichen Berechtigungen für den Zugriff auf Ihre MySQL-kompatible Datenbank verfügt. Gehen Sie wie folgt vor, um ein Benutzerkonto zu erstellen und die Mindestberechtigungen bereitzustellen, die zur Ausführung dieses Skripts erforderlich sind.

So richten Sie ein Benutzerkonto mit den Mindestberechtigungen für die Ausführung dieser Skripts ein

- Erstellen Sie den Benutzer zum Ausführen der Skripts.

```
create user 'username'@'hostname' identified by password;
```

- Erteilen Sie den select-Befehl für Datenbanken, um sie zu analysieren.

```
grant select on database-name.* to username;  
grant replication client on *.* to username;
```

- ```
grant execute on procedure mysql.rds_show_configuration to username;
```

In den folgenden Themen wird beschrieben, wie jedes für eine MySQL-kompatible Datenbank verfügbare Unterstützungsskript heruntergeladen, überprüft und ausgeführt wird. Dort erfahren Sie auch, wie Sie die Skriptausgabe überprüfen und in Ihren AWS-Support-Fall hochladen können.

## Themen

- [Das Skript `awsdms\_support\_collector\_MySQL.sql`](#)

## Das Skript `awsdms_support_collector_MySQL.sql`

Laden Sie das [awsdms\\_support\\_collector\\_MySQL.sql](#)-Skript herunter.

Dieses Skript erfasst Informationen über Ihre MySQL-kompatible Datenbankkonfiguration. Denken Sie daran, die Prüfsumme des Skripts zu überprüfen. Wenn die Prüfsumme verifiziert wurde, überprüfen Sie den SQL-Code in dem Skript, um den Code auszukommentieren, dessen Ausführung Sie nicht wünschen. Wenn Sie mit der Integrität und dem Inhalt des Skripts zufrieden sind, können Sie es ausführen.

Führen Sie das Skript aus, nachdem Sie über die Befehlszeile eine Verbindung zu Ihrer Datenbankumgebung hergestellt haben.

So führen Sie dieses Skript aus und laden die Ergebnisse in Ihren Support-Fall hoch

1. Verbinden Sie sich mit Ihrer Datenbank mithilfe des `mysql`-Befehls.

```
mysql -h hostname -P port -u username database-name
```

2. Führen Sie das Skript mit dem folgenden `mysql-source`-Befehl aus.

```
mysql> source awsdms_support_collector_MySQL_compatible_DB.sql
```

Überprüfen Sie den generierten Bericht und entfernen Sie alle Informationen, die Sie nicht weitergeben möchten. Wenn Sie den Inhalt weitergeben können, laden Sie die Datei in Ihren AWS-Support-Fall hoch. Weitere Informationen zum Hochladen dieser Datei finden Sie unter [Arbeiten mit Diagnoseunterstützungsskripts in AWS DMS](#).



**Note**

- Wenn Sie bereits über ein Benutzerkonto mit den unter [Diagnoseunterstützungsskripts für MySQL-kompatible Datenbanken](#) beschriebenen erforderlichen Berechtigungen verfügen, können Sie das vorhandene Benutzerkonto auch verwenden, um das Skript auszuführen.
- Denken Sie daran, eine Verbindung zu Ihrer Datenbank herzustellen, bevor Sie das Skript ausführen.
- Das Skript generiert seine Ausgabe im Textformat.
- Unter Berücksichtigung der bewährten Sicherheitsmethoden empfehlen wir, dieses Benutzerkonto nach erfolgreicher Ausführung des Skripts zu löschen, wenn Sie ein neues Benutzerkonto nur zur Ausführung dieses MySQL-Diagnoseunterstützungsskripts erstellen.

## PostgreSQL-Diagnoseunterstützungsskripts

Im Folgenden finden Sie die Diagnoseunterstützungsskripts, die für die Analyse einer PostgreSQL-RDBMS (On-Premises, Amazon RDS oder Aurora PostgreSQL) in Ihrer AWS DMS-Migrationskonfiguration verfügbar sind. Diese Skripts funktionieren entweder mit einem Quell- oder Zielpunkt. Die Skripts wurden alle so geschrieben, dass sie im Befehlszeilen-Dienstprogramm `psql` ausgeführt werden können.

Bevor Sie diese Skripts ausführen, stellen Sie sicher, dass das von Ihnen verwendete Benutzerkonto über die erforderlichen Berechtigungen für den Zugriff auf eine PostgreSQL-RDBMS verfügt:

- PostgreSQL 10.x oder höher – Ein Benutzerkonto mit Ausführungsberechtigung für die `pg_catalog.pg_ls_waldir`-Funktion.
- PostgreSQL 9.x oder früher – Ein Benutzerkonto mit Standardberechtigungen.

Wir empfehlen, ein vorhandenes Konto mit den entsprechenden Berechtigungen zu verwenden, um diese Skripts auszuführen.

Wenn Sie ein neues Benutzerkonto erstellen oder einem vorhandenen Konto Berechtigungen zur Ausführung dieser Skripts gewähren müssen, können Sie die folgenden SQL-Befehle für jedes PostgreSQL-RDBMS ausführen, basierend auf der PostgreSQL-Version.

So gewähren Sie Kontoberechtigungen zur Ausführung dieser Skripts für eine PostgreSQL-Datenbank der Version 10.x oder höher

- Führen Sie eine der folgenden Aktionen aus:
  - Führen Sie für ein neues Benutzerkonto Folgendes aus.

```
CREATE USER script_user WITH PASSWORD 'password';
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

- Führen Sie für ein vorhandenes Benutzerkonto Folgendes aus.

```
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

So gewähren Sie Kontoberechtigungen zur Ausführung dieser Skripts für eine PostgreSQL-Datenbank der Version 9.x oder früher

- Führen Sie eine der folgenden Aktionen aus:
  - Führen Sie für ein neues Benutzerkonto Folgendes mit Standardberechtigungen aus.

```
CREATE USER script_user WITH PASSWORD password;
```

- Verwenden Sie für ein vorhandenes Benutzerkonto die vorhandenen Berechtigungen.

#### Note

Diese Skripts unterstützen bestimmte Funktionen im Zusammenhang mit der Ermittlung der WAL-Größe für PostgreSQL-9.x- und frühere Datenbanken nicht. Weitere Informationen erhalten Sie vom AWS-Support.

In den folgenden Themen wird beschrieben, wie Sie jedes für PostgreSQL verfügbare Unterstützungsskript herunterladen, überprüfen und ausführen. Außerdem wird beschrieben, wie Sie die Skriptausgabe überprüfen und in Ihren AWS-Support-Fall hochladen.

Themen

- [Das Skript `awsdms\_support\_collector\_postgres.sql`](#)

## Das Skript `awsdms_support_collector_postgres.sql`

Laden Sie das [awsdms\\_support\\_collector\\_postgres.sql](#)-Skript herunter.

Dieses Skript erfasst Informationen über Ihre PostgreSQL-Datenbankkonfiguration. Denken Sie daran, die Prüfsumme im Skript zu überprüfen. Wenn die Prüfsumme verifiziert wurde, überprüfen Sie den SQL-Code in dem Skript, um den Code auszukommentieren, dessen Ausführung Sie nicht wünschen. Wenn Sie mit der Integrität und dem Inhalt des Skripts zufrieden sind, können Sie es ausführen.

### Note

Sie können dieses Skript mit dem psql-Client Version 10 oder höher ausführen.

Sie können die folgenden Verfahren verwenden, um dieses Skript entweder von Ihrer Datenbankumgebung oder von der Befehlszeile aus auszuführen. In beiden Fällen können Sie Ihre Datei später zum AWS-Support hochladen.

So führen Sie dieses Skript aus und laden die Ergebnisse in Ihren Support-Fall hoch

1. Führen Sie eine der folgenden Aktionen aus:

- Führen Sie das Skript mit der folgenden psql-Befehlszeile aus Ihrer Datenbankumgebung aus.

```
dbname=# \i awsdms_support_collector_postgres.sql
```

Geben Sie bei der folgenden Eingabeaufforderung nur den Namen eines der Schemata ein, die Sie migrieren möchten.

Geben Sie bei der folgenden Eingabeaufforderung den Namen des Benutzers (*script\_user*) ein, den Sie für die Verbindung mit der Datenbank definiert haben.

- Führen Sie das folgende Skript direkt über die Befehlszeile aus. Diese Option vermeidet jegliche Eingabeaufforderungen vor der Ausführung des Skripts.

```
psql -h database-hostname -p port -U script_user -d database-name -f
awsdms_support_collector_postgres.sql
```

2. Überprüfen Sie die Ausgabe-HTML-Datei und entfernen Sie alle Informationen, die Sie nicht weitergeben möchten. Wenn Sie den HTML-Code weitergeben können, laden Sie die Datei in

Ihren AWS-Support-Fall hoch. Weitere Informationen zum Hochladen dieser Datei finden Sie unter [Arbeiten mit Diagnoseunterstützungsskripts in AWS DMS](#).

## Arbeiten mit dem AWS DMS-Diagnoseunterstützungs-AMI

Wenn Sie bei der Arbeit mit AWS DMS auf ein Netzwerkproblem stoßen, benötigt Ihr Support-Techniker möglicherweise weitere Informationen zu Ihrer Netzwerkkonfiguration. Wir möchten sicherstellen, dass der AWS-Support so viele der erforderlichen Informationen wie möglich in kürzester Zeit erhält. Aus diesem Grund haben wir ein vorgefertigtes Amazon-EC2-AMI mit Diagnose-Tools zum Testen Ihrer AWS DMS-Netzwerkumgebung entwickelt.

Die auf dem Amazon Machine Image (AMI) installierten Diagnosetests umfassen Folgendes:

- Virtual Private Cloud (VPC)
- Verlust von Netzwerkpaketen
- Netzwerklatenz
- Größe der Maximum Transmission Unit (MTU)

### Themen

- [Starten Sie eine neue diagnostische AWS DMS-Amazon-EC2-Instance](#)
- [Erstellen einer IAM-Rolle](#)
- [Ausführen von Diagnosetests](#)
- [Nächste Schritte](#)
- [AMI-IDs nach Region](#)

#### Note

Wenn Sie Leistungsprobleme bei Ihrer Oracle-Quelle feststellen, können Sie die Leseleistung Ihrer Oracle-Redo- oder -Archivprotokolle bewerten, um Möglichkeiten zur Leistungssteigerung zu finden. Weitere Informationen finden Sie unter [Bewertung der Leseleistung von Oracle-Redo- oder -Archivprotokollen](#).

## Starten Sie eine neue diagnostische AWS DMS-Amazon-EC2-Instance

In diesem Abschnitt starten Sie eine neue Amazon-EC2-Instance. Informationen zum Starten einer Amazon-EC2-Instance finden Sie unter [Erste Schritte mit Amazon-EC2-Instances für Linux](#) im [Amazon-EC2-Benutzerhandbuch](#).

Starten Sie eine Amazon-EC2-Instance mit den folgenden Einstellungen:

- Suchen Sie für Anwendungs- und BS-Images (Amazon Machine Image) nach dem DMS-DIAG-AMI-AMI. Wenn Sie an der Konsole angemeldet sind, können Sie mit [dieser Abfrage](#) nach dem AMI suchen. Die AMI-ID des AWS-Diagnose-AMI in Ihrer Region finden Sie unter [AMI-IDs nach Region](#).
- Als Instance-Typ empfehlen wir Ihnen, t2.micro zu wählen.
- Wählen Sie für Netzwerkeinstellungen dieselbe VPC aus, die Ihre Replikations-Instance verwendet.

Sobald die Instance aktiv ist, stellen Sie eine Verbindung mit der Instance her. Informationen zum Verbinden mit einer Amazon-EC2-Linux-Instance finden Sie unter [Verbindung mit Ihrer Linux-Instance](#).

## Erstellen einer IAM-Rolle

Wenn Sie die Diagnosetests auf Ihrer Replikations-Instance mit den erforderlichen Mindestberechtigungen ausführen möchten, erstellen Sie eine IAM-Rolle, die die folgende Berechtigungsrichtlinie verwendet:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor0",
 "Effect": "Allow",
 "Action": [
 "dms:DescribeEndpoints",
 "dms:DescribeTableStatistics",
 "dms:DescribeReplicationInstances",
 "dms:DescribeReplicationTasks",
 "secretsmanager:GetSecretValue"
],
 "Resource": "*"
 }
]
}
```

```
}
]
}
```

Fügen Sie die Rolle einem neuen IAM-Benutzer an. Informationen zum Erstellen und Schützen von IAM-Rollen, -Richtlinien und -Benutzern finden Sie in den folgenden Themen im [IAM-Benutzerhandbuch](#):

- [Erste Schritte mit IAM](#)
- [Erstellen von IAM-Rollen](#)
- [Erstellen von IAM-Richtlinien](#)

## Ausführen von Diagnosetests

Nachdem Sie eine Amazon-EC2-Instance erstellt und eine Verbindung zu ihr hergestellt haben, gehen Sie wie folgt vor, um Diagnosetests auf Ihrer Replikations-Instance durchzuführen.

1. Konfigurieren der AWS-CLI:

```
$ aws configure
```

Geben Sie die Zugangsdaten für das AWS-Benutzerkonto ein, das Sie für die Ausführung der Diagnosetests verwenden möchten. Geben Sie die Region für Ihre VPC und Replikations-Instance an.

2. Zeigen Sie die verfügbaren AWS DMS-Aufgaben in Ihrer Region an. Ersetzen Sie Beispielregion durch Ihre Region.

```
$ dms-report -r us-east-1 -l
```

Dieser Befehl zeigt den Status Ihrer Aufgaben an.







Mit diesem Befehl werden Diagnosedaten zur VPC, zur Netzwerkpaketübertragung, zur Netzwerklatenz und zur MTU-Größe (Maximum Transmission Unit) Ihrer Replikations-Instance angezeigt.

```

#####
#
#
AWS DMS Diagnostic
Date: 07-13-2022
#
#
aws region: us-east-2
#
#
#####
==== DMS DIAG Info ====
Public IP: 3.22.101.12
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-ec2-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-ec2-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-ec2-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**

## Nächste Schritte

In den folgenden Abschnitten werden Informationen zur Fehlerbehebung beschrieben, die auf den Ergebnissen der Netzwerkdiagnosetests basieren:

### VPC-Tests

Dieser Test stellt sicher, dass sich die diagnostische Amazon-EC2-Instance in derselben VPC wie die Replikations-Instance befindet. Wenn sich die diagnostische Amazon-EC2-Instance nicht in derselben VPC wie Ihre Replikations-Instance befindet, beenden und erstellen Sie sie erneut in der korrekten VPC. Sie können die VPC einer Amazon-EC2-Instance nicht mehr ändern, nachdem Sie sie erstellt haben.

### Tests auf Netzwerkpaketverlust

Dieser Test sendet 10 Pakete an die folgenden Endpunkte und prüft, ob Pakete verloren gehen:

- Der AWS DMS-Amazon-EC2-Metadaten-Service auf Port 80
- Der Quellendpunkt
- Der Zielendpunkt

Alle Pakete sollten erfolgreich ankommen. Wenn Pakete verloren gehen, wenden Sie sich an einen Netzwerktechniker, um das Problem zu ermitteln und eine Lösung zu finden.

### Netzwerklatenztests

Dieser Test sendet 10 Pakete an dieselben Endpunkte wie im vorherigen Test und prüft die Paketlatenz. Alle Pakete sollten eine Latenz von weniger als 100 Millisekunden haben. Wenn Pakete eine Latenz von mehr als 100 Millisekunden haben, wenden Sie sich an einen Netzwerktechniker, um das Problem zu ermitteln und eine Lösung zu finden.

### Tests der Größe der Maximum Transmission Unit (MTU)

Dieser Test ermittelt die MTU-Größe, indem das Traceroute-Tool auf denselben Endpunkten wie beim vorherigen Test verwendet wird. Alle Pakete in diesem Test sollten dieselbe MTU-Größe haben. Wenn Pakete eine andere MTU-Größe haben, wenden Sie sich an einen Systemspezialisten, um das Problem zu ermitteln und eine Lösung zu finden.

## AMI-IDs nach Region

Die folgende Tabelle zeigt die AMI-ID für das AMS Diagnostic AMI in Ihrer Region:

| Region                     | Image-ID              |
|----------------------------|-----------------------|
| USA Ost (Ohio)             | ami-0f9c6801e8cab9399 |
| USA West (Oregon)          | ami-01ceeab889d966e16 |
| USA Ost (Nord-Virginia)    | ami-0617285c8257e8f6e |
| USA West (Nordkalifornien) | ami-0983b989d0a31ed55 |
| Kanada (Zentral)           | ami-0ba9380b30eee288e |
| Europa (Irland)            | ami-05959fe4fcc448f01 |
| Europa (London)            | ami-0de91dfa18b365c04 |
| Europa (Paris)             | ami-0844ecd26f2cc9073 |
| Europa (Frankfurt)         | ami-09bc69e6da9f3adf4 |
| Europa (Stockholm)         | ami-01da340b09bf5ab78 |
| Europa (Mailand)           | ami-0dcfc51c17cfe7727 |
| Europa (Spanien)           | ami-055982caf649e8b45 |
| Asien-Pazifik (Singapur)   | ami-04150ffe00fe300d6 |
| Asien-Pazifik (Sydney)     | ami-049c1d1a9d72bafce |
| Asien-Pazifik (Jakarta)    | ami-02189a4b8d3b2ee23 |
| Asien-Pazifik (Hongkong)   | ami-0543ae06eff17cb07 |
| Asien-Pazifik (Tokio)      | ami-07b91e8e25d7b7890 |
| Asien-Pazifik (Seoul)      | ami-0657ad8cb9b5cf71c |

| Region                 | Image-ID              |
|------------------------|-----------------------|
| Asien-Pazifik (Osaka)  | ami-01ccf3e974933afdc |
| Asien-Pazifik (Mumbai) | ami-0d9fa24bbb8225e60 |
| Naher Osten (VAE)      | ami-0e3dcf51dfe58929b |
| Naher Osten (Bahrain)  | ami-0faad6f64101607f5 |
| Südamerika (São Paulo) | ami-097dd866777097d96 |

# AWS-DMS-Referenz

In diesem Referenzabschnitt finden Sie zusätzliche Informationen, die Sie bei der Verwendung von AWS Database Migration Service (AWS DMS) möglicherweise benötigen, einschließlich Informationen zur Datentypkonvertierung.

AWS DMS verwaltet Datentypen, wenn Sie eine homogene Datenbankmigration durchführen, bei der Quelle und Ziel denselben Engine-Typ verwenden. Wenn Sie eine heterogene Migration durchführen, bei der von einem Datenbank-Engine-Typ zu einem andersartigen Datenbank-Engine-Typ migriert wird, werden die Datentypen in einen vorübergehenden Datentyp konvertiert. Um festzustellen, wie die Datentypen in der Zieldatenbank angezeigt werden, sehen Sie sich die Datentypentabellen für die Quell- und Ziel-Datenbank-Engines an.

Es gibt einige wichtige Aspekte, die Sie hinsichtlich der Datentypen bei der Datenbankmigration berücksichtigen sollten:

- Der FLOAT-Datentyp ist ein Näherungswert. Wenn Sie einen bestimmten Wert von FLOAT einfügen, wird er in der Datenbank möglicherweise anders dargestellt. Dieser Unterschied entsteht, weil FLOAT kein präziser Datentyp ist, wie beispielsweise ein Dezimaldatentyp wie NUMBER oder NUMBER(p, s). Dies hat zur Folge, dass sich der interne Wert des in der Datenbank gespeicherten FLOAT von dem von Ihnen eingefügten Wert unterscheidet. Daher stimmt der migrierte Wert eines FLOAT möglicherweise nicht genau mit dem Wert in der Quelldatenbank überein.

Weitere Informationen zu diesem Problem finden Sie in den folgenden Artikeln:

- [IEEE-Gleitkommawerte](#) in Wikipedia
- [Darstellung von IEEE-Gleitkommawerten](#) auf Microsoft Learn
- [Warum Gleitkommazahlen ihre Genauigkeit verlieren können](#) auf Microsoft Learn

## Themen


- [Datentypen für den AWS Database Migration Service](#)

## Datentypen für den AWS Database Migration Service

AWS Database Migration Service verwendet integrierte Datentypen für die Migration von Daten eines Datenbank-Engine-Typs zu einem anderen Datenbank-Engine-Typ. In der folgenden Tabelle sind die integrierten Datentypen mit ihren Beschreibungen aufgeführt.

| AWS-DMS-Datentypen | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STRING             | Eine Zeichenfolge                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| WSTRING            | Eine Doppelbyte-Zeichenfolge                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| BOOLEAN            | Ein Boolescher Wert                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| BYTE               | Ein binärer Datenwert                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| DATUM              | Ein Datumswert: Jahr, Monat, Tag                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| TIME               | Ein Zeitwert: Stunden, Minuten, Sekunden                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| DATETIME           | Ein Zeitstempelwert: Jahr, Monat, Tag, Stunde, Minute, Sekunde und Sekundenbruchteil. Die Sekundenbruchteile haben eine maximale Skalierung von 9 Ziffern. Das folgende Format wird unterstützt: JJJJ:MM:TT HH:MM:SS.F(9). Für Amazon S3 Select und Amazon S3 Glacier Select unterscheidet sich das DATETIME-Datentypformat. Weitere Informationen finden Sie in der Beschreibung des primitiven timestamp -Datentyps im Amazon Simple-Storage-Service-Benutzerhandbuch unter <a href="#">Unterstützte Datentypen</a> . |
| INT1               | Eine 1-Byte-Ganzzahl mit Vorzeichen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| INT2               | Eine 2-Byte-Ganzzahl mit Vorzeichen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| INT4               | Eine 4-Byte-Ganzzahl mit Vorzeichen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| INT8               | Eine 8-Byte-Ganzzahl mit Vorzeichen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| NUMERIC            | Ein exakter numerischer Wert mit fester Präzision und Skalierung                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| REAL4              | Ein Gleitkommawert mit einfacher Präzision                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| AWS-DMS-Datentypen | Beschreibung                               |
|--------------------|--------------------------------------------|
| REAL8              | Ein Gleitkommawert mit doppelter Präzision |
| UINT1              | Eine 1-Byte-Ganzzahl ohne Vorzeichen       |
| UINT2              | Eine 2-Byte-Ganzzahl ohne Vorzeichen       |
| UINT4              | Eine 4-Byte-Ganzzahl ohne Vorzeichen       |
| UINT8              | Eine 8-Byte-Ganzzahl ohne Vorzeichen       |
| BLOB               | Binary Large Object.                       |
| CLOB               | Character Large Object                     |
| NCLOB              | Native Character Large Object              |

 Note

AWS DMS kann keinen LOB-Datentyp zu einem Apache Kafka-Endpunkt migrieren.



# AWS DMS-Versionshinweise

Im Folgenden finden Sie Versionshinweise für aktuelle und frühere Versionen von AWS Database Migration Service (AWS DMS).

AWS DMS unterscheidet nicht zwischen Haupt- und Nebenversionen, wenn Sie das automatische Versionsupgrade für Ihre Replikationsinstanz aktivieren. DMS aktualisiert die Version der Replikationsinstanz während des Wartungsfensters automatisch, wenn die Version veraltet ist.

Beachten Sie, dass Sie den Parameter auf setzen müssen, um die Version Ihrer Replikationsinstanz manuell (mithilfe der API oder CLI) von Version 3.4.x auf 3.5.x zu aktualisieren. `AllowMajorVersionUpgrade true` Informationen zu dem `AllowMajorVersionUpgrade` Parameter finden Sie [ModifyReplicationInstance](#) in der DMS-API-Dokumentation.

## Note

Die aktuelle Standard-Engine-Version für AWS DMS ist 3.5.1.

Die folgende Tabelle zeigt die folgenden Daten für aktive DMS-Versionen:

- Das Veröffentlichungsdatum der Version
- Das Datum, nach dem Sie keine neuen Instanzen mit der Version erstellen können
- Das Datum, an dem DMS automatisch Instanzen dieser Version aktualisiert (das EOL-Datum)

| Version | Datum der Veröffentlichung | Kein neues Instanzdatum | EOL-Datum       |
|---------|----------------------------|-------------------------|-----------------|
| 3.5.2   | 29. Oktober 2023           | 30. März 2025           | 29. April 2025  |
| 3.5.1   | 30. Juni 2023              | 30. November 2024       | 30. Januar 2025 |
| 3.4.7   | 31. Mai 2022               | 30. Juli 2024           | 29. August 2024 |
| 3.4.6   | 30. November 2021          | 26. Mai 2024            | 27. Juni 2024   |

## AWS Versionshinweise zu Database Migration Service 3.5.2

### Neue Funktionen in 3.5.2 AWS DMS

| Neue Funktion oder Änderung                                              | Beschreibung                                                                                                                                              |
|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Redshift-Datenvalidierung                                                | AWS DMS unterstützt jetzt die Validierung von Daten in Redshift-Zielen.                                                                                   |
| Unterstützung für Microsoft SQL Server Version 2022 als Quelle und Ziel. | AWS DMS unterstützt jetzt die Verwendung von Microsoft SQL Server Version 2022 als Quelle und Ziel.                                                       |
| IBM Db2 LUW als Ziel                                                     | AWS DMS unterstützt jetzt IBM Db2 LUW als Ziel. Mit dieser AWS DMS Funktion können Sie jetzt Live-Migrationen von IBM Db2 LUW zu IBM Db2 LUW durchführen. |

## AWS Versionshinweise zu Database Migration Service 3.5.1

Die folgende Tabelle zeigt die neuen Features und Verbesserungen, die in AWS Database Migration Service (AWS DMS) Version 3.5.1 eingeführt wurden.

| Neue Funktion oder Änderung                                                 | Beschreibung                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unterstützung für PostgreSQL 15.x                                           | AWS DMS Version 3.5.1 unterstützt PostgreSQL Version 15.x. Weitere Informationen finden Sie unter <a href="#">Verwenden von PostgreSQL als Quelle</a> und <a href="#">Verwenden von PostgreSQL als Ziel</a> .                          |
| Unterstützung für Amazon-DocumentDB-Elastic-Cluster mit Sharded Collections | AWS DMS Version 3.5.1 unterstützt Amazon DocumentDB Elastic Clusters mit Sharded Collections. Weitere Informationen finden Sie unter <a href="#">Verwenden von Amazon DocumentDB als Ziel für den AWS Database Migration Service</a> . |

| Neue Funktion oder Änderung           | Beschreibung                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Redshift Serverless als Ziel          | Unterstützung für Amazon Redshift Serverless als Zielendpunkt. Weitere Informationen finden Sie unter <a href="#">Verwenden einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service</a> .                                                                                                                                           |
| Babelfish-Endpunkt einstellungen      | Verbesserte PostgreSQL-Zielendpunkteinstellungen für die Bereitstellung von Babelfish-Unterstützung. Weitere Informationen finden Sie unter <a href="#">Verwenden einer PostgreSQL-Datenbank als Ziel für AWS Database Migration Service</a> .                                                                                                          |
| Offene Transaktionen in Oracle-Quelle | AWS DMS 3.5.1 verbessert die Methode zur Behandlung offener Transaktionen, wenn eine reine CDC-Aufgabe von der Startposition für eine Oracle-Quelle aus gestartet wird. Weitere Informationen finden Sie unter <code>OpenTransactionWindow</code> im Abschnitt <a href="#">Endpunkteinstellungen bei Verwendung von Oracle als Quelle für AWS DMS</a> . |
| Amazon Timestream als Ziel            | Support für die Verwendung von Amazon Timestream als Zielendpunkt. Weitere Informationen finden Sie unter <a href="#">Verwenden von Amazon Timestream als Ziel für AWS Database Migration Service</a> .                                                                                                                                                 |

AWS DMS Version 3.5.1 enthält die folgenden behobenen Probleme:

| Behobenes Problem                                    | Beschreibung                                                                                                                                                                                                                                             |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Oracle als Quelle für die Zunahme inaktiver Sessions | Es wurde ein Problem für die Oracle-Quelle behoben, bei dem nur CDC-Aufgaben kontinuierlich mehr inaktive Sitzungen aufwiesen, was zu der folgenden Ausnahme führte: <code>ORA-00020: maximum number of processes exceeded on the source database</code> |
| UPDATE-Änderungen auf DocumentDB replizieren         | Es wurde ein Problem für DocumentDB als Ziel behoben, bei dem UPDATE-Anweisungen in einigen Szenarien nicht richtig repliziert wurden.                                                                                                                   |

| Behobenes Problem                                             | Beschreibung                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Aufgabe, die nur für die Validierung bestimmt ist             | Die Fehlerbehandlung für die Datenüberprüfungsfunktion wurde verbessert, sodass die Aufgabe ordnungsgemäß fehlschlägt, wenn die Datenüberprüfung für reine Validierungsaufgaben deaktiviert ist.                                                                                                   |
| Redshift-Replikation nach Verbindungsabbruch                  | Es wurde ein Problem für das Redshift-Ziel behoben, bei dem die DMS-Aufgabe nicht erneut versuchte, Änderungen auf das Ziel anzuwenden, wenn das Ziel nach dem Verbindungsabbruch einen Wert größer als Null <code>ParallelApplyThreads</code> festgelegt hatte, was zu Datenverlust führen würde. |
| MySQL-Text-zu-Mediumtext-Replikation                          | Es wurde ein Problem bei der MySQL-zu-MySQL-Replikation von Mediumtext-Datentypen im Full-LOB-Modus behoben.                                                                                                                                                                                       |
| Die CDC-Aufgabe repliziert sich nicht mit rotiertem Schlüssel | Es wurde ein Problem mit DMS-Aufgaben behoben, bei dem <code>true</code> den <code>BatchApplyEnabled</code> festgelegt war, dass DMS die Replikation von Daten beendet, nachdem Secrets Manager das Passwort geändert hatte.                                                                       |
| Problem mit der MongoDB/DocumentDB-Segmentierung              | Es wurde ein Problem für die MongoDB/DocDB-Quelle behoben, bei dem die Bereichssegmentierung nicht richtig funktionierte, wenn die Primärschlüsselspalte einen großen Wert enthielt.                                                                                                               |
| Oracle-Datenvalidierung ungebundener numerischer Werte        | Es wurde ein Problem für Oracle Target behoben, bei dem DMS <code>STRING</code> während der Datenvalidierung einen Wert mit ungebundenem Datentyp <code>NUMERIC</code> als <code>a</code> erkannte.                                                                                                |
| SQL Server-Datenvalidierung                                   | Es wurde ein Problem für SQL Server-Endpunkte behoben, bei dem bei der DMS-Datenvalidierung eine ungültige SQL-Anweisung erstellt wurde.                                                                                                                                                           |
| MongoDB-Auto-Segmentierung                                    | Die Funktionalität der automatischen Partitionierung von Daten bei der parallelen Migration von Dokumenten aus MongoDB als Quelle wurde verbessert.                                                                                                                                                |

| Behobenes Problem                                           | Beschreibung                                                                                                                                                                               |
|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon-S3-Apache-Parquet-Format                             | Es wurde ein Problem behoben, so dass Apache-Parquet-Dateien, die in S3 als Ziel geschrieben wurden, mit Python mit Apache Arrow C++ angezeigt werden können.                              |
| Umgang mit PostgreSQL als Quell-DDL                         | Es wurde ein Problem mit der PostgreSQL als Quelle behoben, bei dem nicht unterstützte DDL-Operationen nicht korrekt ignoriert werden.                                                     |
| PostgreSQL-timestamp tz -Datenfehler                        | Es wurde ein Problem mit PostgreSQL-zu-PostgreSQL-Migrationen behoben, bei dem der Zeitstempel mit Zeitzonendaten nicht richtig migriert wird, wenn Batch Apply während CDC aktiviert ist. |
| Fehler bei der Validierung von Oracle zu PostgreSQL         | Es wurde ein Problem mit Migrationen von Oracle zu PostgreSQL behoben, bei dem die Datenvalidierung für den Datentyp NUMERIC(38,30) fehlschlägt.                                           |
| Fehler bei erweiterten Oracle-Datentypen                    | Es wurde ein Problem mit Oracle als Quelle behoben, bei dem der erweiterte Varchar-Datentyp gekürzt wird.                                                                                  |
| Kombinieren von Filteroperatoren                            | Es wurde ein Problem mit der Spaltenfilterfunktion behoben, bei dem der Nullspaltenoperator nicht mit anderen Operatortypen kombiniert werden kann.                                        |
| CDC-Latenz aufgrund übermäßiger Protokollierung             | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem eine übermäßige Protokollierung von Pglogical-Plug-in-Warnungen zu einer Latenz der Quell-CDC führt.                       |
| Umgang mit bidirektionaler Replikation von Create Table DDL | Es wurde ein Problem bei der bidirektionalen Replikation von PostgreSQL zu PostgreSQL behoben, bei dem die DDL-Änderung „Create Table DDL“ nicht korrekt repliziert wird.                  |
| CDC-Fehler bei der Verwendung von Filtern                   | Es wurde ein Problem mit dem Filter-Feature behoben, durch das die CDC-Replikation fehlschlug.                                                                                             |

| Behobenes Problem                                                         | Beschreibung                                                                                                                                                                                                     |
|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Validierung des Host-Namens der Zertifizierungsstelle für Kafka-Endpunkte | Die Funktionalität von Kafka-Endpunkten wurde um die Option erweitert, die Host-Namen-Validierung der Zertifizierungsstelle zu deaktivieren ( <code>SslEndpointIdentificationAlgorithm</code> ).                 |
| IBM-Db2-LUW-Validierung                                                   | Es wurde ein Problem behoben, durch das die Datentypen Datum, Zeitstempel und Uhrzeit der Db2-LUW-Quelle bei der Datenvalidierung nicht richtig behandelt wurden.                                                |
| S3-Validierung                                                            | Es wurde ein Problem mit Migrationen von Db2 LUW zu S3 behoben, bei dem die Validierungsfunktion den Datentyp timestamp (0) nicht richtig behandelt.                                                             |
| Fehler beim Neustart einer DMS-Aufgabe                                    | Es wurde ein Problem mit der PostgreSQL-Quelle behoben, bei dem die AWS DMS Aufgabe nicht neu gestartet werden konnte und bei Verwendung des pglogical-Plugins keine relationalen Ereignisse verarbeiten konnte. |
| SQL-Server-Validierung des HIERARCHY-Datentyps                            | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem die Validierung des HIERARCHY-Datentyps fehlschlägt.                                                                                             |
| SQL-Server-Zeichenfolgen mit Steuerzeichen                                | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem Zeichenketten mit Steuerzeichen nicht korrekt repliziert werden.                                                                                 |
| Redshift mit Secrets Manager                                              | Es wurde ein Problem mit Redshift als Ziel behoben, bei dem das Testen des Endpunkts fehlschlug, wenn Secrets Manager verwendet wird.                                                                            |
| Inkonsistenz ParallelLoadThreads bei MySQL-Einstellungen                  | Es wurde ein Problem mit MySQL als Ziel behoben, bei dem die <code>ParallelLoadThreads</code> -Einstellung nach Änderungen der Aufgabeneinstellungen nicht korrekt beibehalten wird.                             |
| Fehler bei der Zuordnung von PostgreSQL zu Oracle-Datentypen              | Es wurde ein Problem mit Migrationen von PostgreSQL zu Oracle behoben, bei dem die Aufgabe fehlschlägt, wenn vom Datentyp TEXT zum Datentyp VARCHAR2(2000) repliziert wurde.                                     |

| Behobenes Problem                                          | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Datenvalidierung von Oracle zu PostgreSQL                  | Es wurde ein Problem mit Migrationen von Oracle zu PostgreSQL behoben, bei dem bei der Datenvalidierung falsch positive Ergebnisse gemeldet werden, wenn NULL-Zeichen als SPACE-Zeichen repliziert werden.                                                                                                                                                                                                                                                                                |
| SQL Server-Quelle in der Konfiguration AlwaysOn            | Es wurde ein Problem mit der SQL Server-Quelle in der AlwaysOn Konfiguration behoben, bei dem die AWS DMS Aufgabe fehlschlug, wenn der Replikatname nicht genau mit dem tatsächlichen Servernamen übereinstimmte.                                                                                                                                                                                                                                                                         |
| Fehler beim Oracle-Quellentest                             | Es wurde ein Problem mit der Oracle-Quelle behoben, bei dem der AWS DMS Endpunkt-Verbindungstest aufgrund unzureichender Rechte beim Abrufen der Oracle-Sitzungs-ID (SID) fehlschlug.                                                                                                                                                                                                                                                                                                     |
| CDC nimmt keine neuen Tabellen auf                         | Es wurde ein Problem mit Nur-CDC-Aufgaben behoben, bei dem Tabellen, die nach dem Start der Aufgabe auf der Quelle erstellt wurden, in einigen Fällen nicht repliziert werden.                                                                                                                                                                                                                                                                                                            |
| Offene Transaktionen in Oracle als Quelle                  | Die Methode zum Umgang mit offenen Transaktionen, wenn eine Nur-CDC-Aufgabe von der Startposition für eine Oracle-Quelle aus gestartet wird, wurde verbessert.                                                                                                                                                                                                                                                                                                                            |
| Problem mit fehlenden Daten                                | Es wurde ein Problem behoben, bei dem Daten bei der Fortsetzung einer Aufgabe fehlen, wenn diese beendet wird, nachdem zwischengespeicherte Änderungen angewendet wurden ( <code>StopTaskCachedChangesApplied</code> -Option auf „true“ gesetzt). Dieses Problem konnte selten auftreten, wenn aufgrund einer großen Anzahl AWS DMS von Änderungen an der Quelle zwischengespeicherte Änderungen auf der Festplatte der AWS DMS Replikationsinstanz dauerhaft zwischengespeichert wurden. |
| Problem bei der Datenvalidierung beim erweiterten Datentyp | Es wurde ein Problem bei der Datenvalidierung von PostgreSQL zu Oracle behoben, bei dem die Validierung für erweiterte Datentypen fehlschlägt.                                                                                                                                                                                                                                                                                                                                            |

| Behobenes Problem                                                        | Beschreibung                                                                                                                                                                                                |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Problem mit der Datenvalidierung bei inkonsistenter Zeichenkodierung     | Es wurde ein Problem bei der Datenvalidierung von SQL Server zu PostgreSQL behoben, bei dem die Validierung fehlschlägt, wenn die Zeichenkodierung zwischen Quelle und Ziel inkonsistent ist.               |
| Problem mit der Datenüberprüfung ORA-01455                               | Es wurde ein Problem behoben, bei dem während der Validierung ein ORA-01455-Fehler auftritt, wenn ein PostgreSQL-integer einem Oracle-number(10) zugeordnet wird.                                           |
| Unterstützung von SQL Server IDENTITY                                    | Es wurde ein Problem bei der Datenreplikation von SQL Server zu SQL Server behoben, bei dem die Migration von Identitätsspalten fehlschlägt, wenn die Zielspalte die Eigenschaft IDENTITY hat.              |
| Zeichensatzproblem bei ALTER-Anweisungen                                 | Es wurde ein Problem bei der MySQL-zu-MySQL-Replikation behoben, bei dem der Zeichensatz bei der Migration einer ALTER-Anweisung während CDC auf UTF16 AWS DMS geändert wurde.                              |
| Unterstützung für den Datentyp PostgreSQL zu Redshift Spatial            | Es wurde Unterstützung für den spatial-Datentyp bei der Migration von PostgreSQL zu Amazon Redshift hinzugefügt.                                                                                            |
| GZIP-Komprimierung von .parquet-Dateien                                  | Es wurde ein Problem behoben, bei dem AWS DMS .parquet-Dateien mit GZIP-Komprimierung mit S3 als Ziel nicht generiert werden konnten.                                                                       |
| Migration von MongoDB/DocDB-Quellen                                      | Es wurde ein Problem behoben, bei dem einige Partitionen AWS DMS nicht aus einer MongoDB-Quelle migriert wurden.                                                                                            |
| Problem mit Statistiken in Tabelle                                       | Es wurde ein Problem behoben, bei dem Tabellenstatistiken nicht angezeigt werden, wenn mindestens eine der Aufgaben auf der Replikations-Instance mehr als 1 001 Tabellen enthält.                          |
| Die Tabelle wird für IBM Db2 LUW-Versionen 10.1.0 und niedriger gesperrt | Es wurde ein Problem mit der Db2-LUW-Quelle behoben, bei dem die Tabellenmigration mit dem Fehler TYPESTRINGUNITS is not valid unterbrochen wird, wenn die Quelldatenbankversion 10.1.0 oder niedriger ist. |



| Behobenes Problem                                            | Beschreibung                                                                                                                                                                                               |
|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MongoDB-Partitionierungsproblem                              | Es wurde ein Problem für MongoDB/DocDB behoben, bei dem ein oder mehrere Segmente der Quellpartition fehlen.                                                                                               |
| MongoDB-Partitionierungsproblem                              | Es wurde ein Problem behoben, bei dem die Segmentierung auf der Grundlage einer Spalte mit dem Typ <code>NumberLong</code> () aufgrund eines Fehlers bei der Typkonvertierung fehlschlägt.                 |
| MongoDB-Partitionierungsproblem                              | Verbesserte Autosegmentierungsleistung für große Datensätze mit MongoDB als Quelle.                                                                                                                        |
| MongoDB-Treiberversion                                       | Der MongoDB-Treiber wurde auf 1.20.0 herabgestuft, um die MongoDB-Versionen 3.6 und niedriger weiterhin zu unterstützen.                                                                                   |
| Amazon-S3-Apache-Parquet-Zeitstempel-Datentyp                | Ein Problem mit Amazon S3 Parquet Target wurde behoben. AWS DMS setzt den Formatparameter jetzt so <code>isAdjustedToUTC true</code> , dass er dem Verhalten in früheren Versionen von entspricht AWS DMS. |
| Amazon Redshift als Zielkopierbefehl                         | Es wurde ein Problem für Amazon Redshift als Ziel behoben, bei dem der Kopierbefehl für große Tabellen beim Kopieren von Daten von Amazon S3 zu Amazon Redshift fehlschlägt.                               |
| PostgreSQL-Geometriedatentypen                               | Es wurde ein Problem bei Migrationen von PostgreSQL zu PostgreSQL behoben, bei dem die Migration bei großen Geometriedatentypen fehlschlägt.                                                               |
| Oracle zu PostgreSQL XML                                     | Es wurde ein Problem behoben, bei dem bei der Migration bei der Replikation von Oracle zu PostgreSQL zusätzlicher Speicherplatz auf XML hinzugefügt wird.                                                  |
| Aktualisierung des Ziel-Checkpoints in unterstützten Engines | AWS DMS aktualisiert jetzt den Ziel-Checkpoint in der <code>awsdms_txn_state</code> Tabelle in der Zieldatenbank.                                                                                          |
| MongoDB/DocDB-Datensätze werden an falsche Sammlung gesendet | Es wurde ein Problem für MongoDB/DocDB behoben, bei dem Daten an die falsche Zielsammlung gesendet werden.                                                                                                 |

| Behobenes Problem                                                             | Beschreibung                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Neue Oracle-Quellentabelle mit EscapeCharacter Endpunkteinstellung            | Es wurde ein Problem mit der Oracle-Quelle behoben, bei dem neue Tabellen für die Replikation nur dann abgerufen wurden, wenn die Aufgabe gestoppt und wieder aufgenommen wurde, während die EscapeCharacter Endpunkteinstellung festgelegt war.                                                                                        |
| CDC-Wiederherstellungsprüfungspunkt                                           | Es wurde eine zwischen dem Zieldatenspeicher und der AWS DMS -Konsole beobachtete Inkonsistenz beim CDC-Wiederherstellungsprüfungspunkt behoben.                                                                                                                                                                                        |
| Nur-CDC-Validierungsaufgaben                                                  | Es wurde ein Problem mit Nur-CDC-Validierungsaufgaben behoben, bei denen die Aufgabe nur bei der CDC-Validierung fehlschlägt, selbst wenn alle Tabellen in der Aufgabe fehlschlagen.                                                                                                                                                    |
| Validierungsverhalten bei Problemen mit Quell- oder Zielverbindungen          | Es wurde ein Problem mit der Datenvalidierung behoben, bei dem AWS DMS die Tabellen auf der Quelle oder dem Ziel angehalten wurden, wenn die Verbindung unterbrochen wurde.                                                                                                                                                             |
| Falsche positive Ergebnisse bei der Datenvalidierung von Oracle zu PostgreSQL | Es wurde ein Problem mit der Datenvalidierung von Oracle nach PostgreSQL behoben, bei dem falsch positive Ergebnisse von AWS DMS gemeldet wurden. Der Grund dafür ist, dass die Unterschiede bei der Repräsentation von NULL-Zeichen in der Quelle auf dem Ziel nicht mit textbasierten Datentypen außer VARCHAR berücksichtigt werden. |
| Datenkürzung von Oracle zu PostgreSQL                                         | Es wurde ein Problem mit Oracle als Quelle und PostgreSQL als Ziel behoben, bei dem AWS DMS Daten für NVARCHAR-Spalten mit der Oracle-Einstellung NLS_NCHAR_CHARACTERSET auf AL16UTF16 kürzt.                                                                                                                                           |
| Datenvalidierungsfehler                                                       | Es wurde ein Problem mit der Datenüberprüfung behoben, bei dem ein unable to create where filter clause-Fehler ausgelöst wird, wenn sowohl die Quellenfilterung als auch eine Transformationsregel zum Hinzufügen von Spalten verwendet werden.                                                                                         |

| Behobenes Problem                                                                              | Beschreibung                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Umgang mit Redshift-Zielfehlern                                                                | Es wurde ein Problem mit Redshift als Ziel behoben, bei dem die Fehlerbehandlung nicht wie konfiguriert funktioniert, wenn die <code>ParallelApplyThreads</code> -Aufgabeneinstellung für die CDC-Aufgabe auf einen Wert größer als Null gesetzt ist. |
| Kommunikationsfehler bei Oracle als Quelle                                                     | Es wurde ein Problem mit Oracle als Quelle behoben, bei dem die Aufgabe im Status <code>RUNNING</code> verbleibt, aber nach einem Kommunikationsfehler keine Daten migrieren kann.                                                                    |
| Die CDC-Tabelle wird aufgrund von Spaltenfiltern gesperrt                                      | Es wurde ein Problem mit Volllast-plus-CDC-Aufgaben behoben, bei dem eine Tabelle während der CDC-Phase gesperrt wird, wenn Spaltenfilter angewendet werden.                                                                                          |
| S3-Zieldatenvalidierungsfehler für Sonderzeichen                                               | Es wurde ein Problem mit der S3-Zieldatenvalidierung behoben, bei dem die Aufgabe fehlschlägt, wenn der Tabellename ein anderes Sonderzeichen als einen Unterstrich enthält.                                                                          |
| MongoDB-Quelle: Volllast und CDC-Fehler                                                        | Es wurde ein Problem mit MongoDB als Quelle behoben, bei dem eine Volllast-plus-CDC-Aufgabe bei der Verarbeitung von Cache-Ereignissen während der Migration einer großen Sammlung fehlschlägt.                                                       |
| Upgrade-Problem mit <code>BatchApplyEnabled</code> -Einstellung auf „true“                     | Es wurde ein Problem behoben, bei dem eine Aufgabe, bei der die <code>BatchApplyEnabled</code> Task-Einstellung auf <code>true</code> gesetzt war, nach der Migration von AWS DMS Version 3.4.6 auf 3.5.1 in einigen Fällen fehlschlug.               |
| SQL AlwaysOn Server-Quelle mit Sortierung unter Berücksichtigung von Groß- und Kleinschreibung | Es wurde ein Problem mit SQL Server AlwaysOn als Quelle behoben, bei dem eine Aufgabe mit Sortierung unter Berücksichtigung der Groß- und Kleinschreibung fehlschlug.                                                                                 |
| MySQL-Quellaufgabe hängt                                                                       | Es wurde ein Problem mit MySQL als Quelle behoben, bei dem eine Aufgabe hängen bleibt, anstatt fehlschlagen, wenn die Quelle nicht richtig konfiguriert ist.                                                                                          |

| Behobenes Problem                                              | Beschreibung                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vollastaufgabenfehler bei S3-Quelle                            | Es wurde ein Problem mit S3 als Quelle behoben, bei dem eine Aufgabe nach dem Upgrade von AWS DMS Version 3.4.6 oder 3.4.7 auf Version 3.5.1 beim Wiederaufnehmen fehlschlug.                                                                                                                 |
| PostgreSQL-Quelle, bei der CaptureDDLs auf „false“ gesetzt ist | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem DDLs nicht richtig behandelt werden, wenn die CaptureDDLs - Endpunkteinstellung auf „false“ gesetzt ist.                                                                                                                      |
| Absturz der Oracle-Quellaufgabe bei Fortsetzung                | Es wurde ein Problem mit Oracle als Quelle behoben, bei dem eine Aufgabe bei der Fortsetzung aufgrund falscher Daten im Spaltennamen abstürzt.                                                                                                                                                |
| Fehler bei der LOB-Suche in der MySQL-Quelle                   | Es wurde ein Problem mit MySQL als Quelle behoben, bei dem eine LOB-Suche fehlschlägt, wenn die ParallelApplyThreads - Aufgabeneinstellung auf einen Wert größer als Null gesetzt ist.                                                                                                        |
| Unlogischer LSN-Fehler in der SQL-Server-Quelle                | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem eine Aufgabe nach dem Upgrade von AWS DMS Version 3.4.7 auf Version 3.5.1 mit einem <code>illogical LSN sequencing state error</code> Fehler fehlschlug.                                                                      |
| PostgreSQL-Quelle mit pglogical                                | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem eine Aufgabe, die das pglogical-Plugin verwendet, fehlschlägt, wenn die Aufgabe angehalten, eine Tabelle aus den Auswahlregeln entfernt, die Aufgabe fortgesetzt und Änderungen an der entfernten Tabelle vorgenommen werden. |
| Aurora MySQL – falscher Wiederherstellungs-Checkpoint.         | Es wurde ein Problem bei Aurora MySQL als Quelle behoben, bei dem aufgrund eines Aurora-Failovers oder eines Stop-and-Starts der Aurora-Quelle ein falscher Wiederherstellungs-Checkpoint gespeichert wird.                                                                                   |
| Absturz von SQL Server als Quellaufgabe.                       | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem eine Aufgabe abstürzt, wenn <code>SafeguardPolicy</code> auf <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code> eingestellt ist.                                                                                               |

| Behobenes Problem                                                                         | Beschreibung                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Falsches Datentyp-Casting mit MySQL als Ziel                                              | Es wurde ein Problem bei MySQL als Ziel behoben, bei dem die CDC-Replikation aufgrund einer falschen Datentypumwandlung in der Stapelanwendungsphase fehlschlägt.                               |
| Aufgabe schlägt fehl, wenn CaptureDDLs für PostgreSQL als Quelle auf „false“ gesetzt ist. | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem eine Aufgabe fehlschlägt, weil eine DDL als DML behandelt wird, wenn die Endpunkteinstellung CaptureDDLs auf false gesetzt ist. |
| Absturz einer leeren MongoDB-Sammlung                                                     | Es wurde ein Problem mit MongoDB als Quelle behoben, bei dem die Aufgabe aufgrund einer leeren Sammlung abstürzt..                                                                              |
| Absturz bei Vollastaufgabe mit Redshift                                                   | Es wurde ein Problem mit Redshift als Ziel behoben, bei dem eine Aufgabe während der Vollastphase abstürzt, wenn die Kontrolltabelle für den Wiederherstellungs-Checkpoint aktiviert ist.       |
| S3 zu S3 – keine Datenbewegung.                                                           | Es wurde ein Problem bei der Replikation von S3 nach S3 behoben, bei dem AWS DMS die Daten nicht repliziert wurden, wenn das nicht angegeben bucketFolder wurde.                                |
| CDC-Latenz mit GlueCatalogGeneration-Einstellung auf true                                 | Es wurde ein Problem mit S3 als Ziel behoben, bei dem eine übermäßige Latenz auftritt, wenn GlueCatalogGeneration auf true gesetzt ist.                                                         |
| Datenverkürzung bei Oracle als Ziel                                                       | Es wurde ein Problem mit Oracle als Ziel behoben, bei dem Daten in AWS DMS VARCHAR2-Spalten gekürzt wurden.                                                                                     |
| Verhalten des Unterstrichs als Platzhalter in PostgreSQL                                  | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem das Verhalten des Platzhalters '_' in den Auswahlregeln nicht wie dokumentiert funktioniert.                                    |
| Problem mit leerem WAL-Header bei PostgreSQL als Quelle.                                  | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem die Aufgabe aufgrund eines leeren WAL-Headers fehlschlägt, der vom Replikations-Slot empfangen wird.                            |

| Behobenes Problem                                                | Beschreibung                                                                                                                                                                                              |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MySQL oder MariaDB als Quelle mit komprimierten Binärprotokollen | Es wurde ein Problem für MySQL und MariaDB als Quellen behoben, bei dem keine korrekte Fehlermeldung ausgegeben wurde, wenn eine BINLOG-Komprimierung AWS DMS erkannt wurde.                              |
| Sonderzeichen bei der S3-Datenvalidierung                        | Die S3-Datenvalidierung zur Verarbeitung von Sonderzeichen in Primär- und Nicht-Primärschlüsselspalten wurde verbessert.                                                                                  |
| Irreführende Aufgabenprotokolleinträge bei Redshift als Ziel.    | Es wurde ein Problem mit Redshift als Ziel behoben, bei dem irreführende Einträge im Aufgabenprotokoll enthalten sind, die Fehler bei der Stapelanwendung von Anweisungen bei UPDATES und DELETES melden. |
| Absturz der Migration saufgabe von SQL Server zu S3.             | Es wurde ein Problem bei Migrationen von SQL Server zu S3 behoben, bei dem die Aufgabe beim Anwenden zwischeng gespeicherter Änderungen abstürzt.                                                         |
| Fehler mit fehlenden Daten bei Stapelanwendungen.                | Es wurde ein Problem mit dem Feature „Stapelanwendung“ behoben, bei dem ein Fehler beim Anwenden eines Stapels zu fehlenden Daten führen kann.                                                            |

## AWS Hinweise zur Beta-Version von Database Migration Service 3.5.0

### Important

AWS DMS 3.5.0 ist eine Betaversion der Replication Instance Engine. AWS DMS unterstützt diese Version genauso wie alle vorherigen Versionen. Wir empfehlen Ihnen jedoch, AWS DMS 3.5.0 Beta zu testen, bevor Sie sie für Produktionszwecke verwenden.

Die folgende Tabelle zeigt die neuen Funktionen und Verbesserungen, die in AWS Database Migration Service (AWS DMS) Version 3.5.0 Beta eingeführt wurden.

| Neue Funktion oder Änderung                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time Travel für Oracle und Microsoft SQL Server | Sie können Time Travel jetzt in allen AWS Regionen mit DMS-unterstützten Oracle-, Microsoft SQL Server- und PostgreSQL-Quellendpunkten und DMS-unterstützten PostgreSQL- und MySQL-Zielendpunkten verwenden.                                                                                                                                                                                                                                                                                  |
| S3-Validierung                                  | AWS DMS unterstützt jetzt die Validierung replizierter Daten auf Amazon S3 S3-Zielendpunkten. Informationen zur Validierung von Amazon-S3-Zieldaten finden Sie unter <a href="#">Zieldatenvalidierung in Amazon S3</a> .                                                                                                                                                                                                                                                                      |
| Integration des Glue-Katalogs                   | AWS Glue ist ein Service, der einfache Möglichkeiten zur Kategorisierung von Daten bietet und aus einem Metadaten-Repository besteht, das als AWS Glue Data Catalog bezeichnet wird. Sie können jetzt einen AWS Glue Data Catalog in Ihren Amazon S3 S3-Zielendpunkt integrieren und Amazon S3 S3-Daten über andere AWS Dienste wie Amazon Athena abfragen. Weitere Informationen finden Sie unter <a href="#">Verwenden von AWS Glue Data Catalog mit einem Amazon-S3-Ziel für AWS DMS</a> . |
| Parallele Anwendung für DocumentDB als Ziel     | Wenn DocumentDB als Ziel mit neuen ParallelApply* Aufgabeneinstellungen verwendet wird, werden AWS DMS jetzt maximal 5000 Datensätze pro Sekunde während der CDC-Replikation unterstützt. Weitere Informationen finden Sie unter <a href="#">Verwenden von Amazon DocumentDB als Ziel für den AWS Database Migration Service</a> .                                                                                                                                                            |
| Kundenorientierte Protokollierung               | Mit AWS DMS Version 3.5.0 können Sie Task-Logs jetzt effektiver untersuchen und verwalten. Informationen zum Anzeigen und Verwalten von AWS DMS-Aufgabenprotokollen finden Sie unter <a href="#">Anzeigen und Verwalten von AWS-DMS-Aufgabenprotokollen</a> .                                                                                                                                                                                                                                 |
| SASL_PLAIN-Mechanismus für Kafka-Zielendpunkte  | Sie können jetzt die SASL_PLAIN-Authentifizierung verwenden, um Kafka-MSK-Zielendpunkte zu unterstützen.                                                                                                                                                                                                                                                                                                                                                                                      |

| Neue Funktion oder Änderung                               | Beschreibung                                                                                                                                                                                                                |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Replikation von XA-Transaktionen in MySQL                 | Sie können jetzt XA-Transaktionen auf Ihrer MySQL-DMS-Quelle verwenden. Vor DMS 3.5.0 wurden DML-Änderungen, die als Teil von XA-Transaktionen vorgenommen wurden, nicht korrekt repliziert.                                |
| Erweiterte Oracle-Datentypen                              | AWS DMS unterstützt jetzt die Replikation erweiterter Datentypen in Oracle Version 12.2 und höher.                                                                                                                          |
| Db2-LUW-Umgebung PureScale                                | AWS DMS unterstützt jetzt die Replikation aus einer Db2-LUW-Umgebung. PureScale Diese Funktionalität wird nur mit der Option Verarbeitung von Änderungen ab Quellenänderungsposition starten unterstützt.                   |
| SQL Server als Quelle mit READ_COMMITTED_SNAPSHOT -Option | Wenn Sie eine Microsoft SQL Server-Quelldatenbank mit der READ_COMMITTED_SNAPSHOT Option auf verwenden TRUE, können Sie DML-Änderungen korrekt replizieren, indem Sie das Verbindungsattribut forceDataRowLookup festlegen. |

AWS DMS 3.5.0 umfasst die folgenden behobenen Probleme:

Probleme, die in AWS DMS Version 3.5.0, veröffentlicht am 17. März 2023, behoben wurden

| Thema                                                                                            | Auflösung                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Oracle – Vergleich der Sonderschreibung einer Zeichenfolge, die aus einer Zahl konvertiert wurde | Es wurde ein Problem mit Oracle als Quelle behoben, bei dem die Filterregeln für eine numerische Spalte nicht wie erwartet funktionieren, wenn für dieselbe Spalte eine Datentyptransformation in eine Zeichenfolge existiert. |
| Verbesserungen bei SQL Server AG (On-Premises)                                                   | Die Effizienz der Verbindungsverarbeitung mit der SQL Server-Quelle in der AlwaysOnKonfiguration wurde verbessert, indem unnötige Verbindungen zu Replikaten vermieden wurden, die nicht von DMS verwendet werden.             |



| Thema                                                                                                   | Auflösung                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interne Konvertierung von SQL Server HIERARCHYID                                                        | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem der HIERARCHYID-Datentyp als VARCHAR(250) statt als HIERARCHYID auf das SQL Server-Ziel repliziert wird.               |
| Korrektur der S3-Zielverschiebungsaufgabe                                                               | Es wurde ein Problem behoben, bei dem das Verschieben einer Aufgabe mit einem S3-Ziel sehr lange dauert, eingefroren erscheint oder nie abgeschlossen wird.                            |
| SASL-Plain-Mechanismus bei Kafka                                                                        | Es wurde Unterstützung für die SASL-Plain-Authentifizierungsmethode für einen Kafka-MSK-Zielendpunkt eingeführt.                                                                       |
| Das parallele Laden/Anwenden schlägt aufgrund des Parameters <code>_type</code> bei Opensearch 2.x fehl | Es wurde ein Problem bei Opensearch 2.x als Ziel behoben, bei dem paralleles Laden oder paralleles Anwenden fehlschlägt, weil der Parameter <code>_type</code> nicht unterstützt wird. |
| Unterstützung von Tabellenzuordnungsfiltern mit gemischten Operatoren                                   | Es wurde eine Einschränkung beseitigt, bei der nur ein Filter auf eine Spalte angewendet werden kann.                                                                                  |
| S3-, Kinesis- und Kafka-Endpunkte – alter-basierte Migration von LOB-Spalten in der CDC-Phase           | Es wurde ein Problem für Kinesis-, Kafka- und S3-Ziele behoben, bei dem Daten in LOB-Spalten, die während CDC hinzugefügt wurden, nicht repliziert werden.                             |
| MongoDB-Treiber-Upgrade                                                                                 | Der MongoDB-Treiber wurde auf Version 1.23.2 aktualisiert.                                                                                                                             |
| Kafka-Treiber-Update                                                                                    | Der Kafka-Treiber wurde von 1.5.3 auf 1.9.2 aktualisiert.                                                                                                                              |

| Thema                                                                                                                             | Auflösung                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Die S3-Endpunkt-Einstellung funktioniert nicht korrekt                                                                            | Es wurde ein Problem mit S3 als Ziel behoben, bei dem die <code>AddTrailingPaddingCharacter</code> -Endpunkt-Einstellung nicht funktioniert, wenn die Daten das als Trennzeichen für das S3-Ziel angegebene Zeichen enthalten.                                       |
| Die Kinesis-Zielaufgabe stürzte ab                                                                                                | Es wurde ein Problem mit Kinesis als Ziel behoben, bei dem eine Aufgabe abstürzt, wenn der PK-Wert leer und detailliertes Debuggen aktiviert ist.                                                                                                                    |
| Wenn die Spaltennamen von S3-Zielen um eine Position verschoben werden                                                            | Es wurde ein Problem für ein S3-Ziel behoben, bei dem Spaltennamen um eine Position verschoben werden, wenn <code>AddColumnName</code> auf <code>true</code> und <code>TimestampColumnName</code> auf <code>""</code> eingestellt ist.                               |
| Verbesserung der Warnung zur LOB-Kürzung bei der Protokollierung                                                                  | Die Warnprotokollierung für die LOB-Kürzung bei SQL Server als Quelle wurde verbessert und enthält jetzt die <code>SELECT</code> -Anweisung, die zum Abrufen des LOB verwendet wird.                                                                                 |
| Es wurde ein schwerwiegender Fehler hinzugefügt, um zu verhindern, dass DMS-Aufgaben abstürzen, wenn das TDE-Passwort falsch ist. | Es wurde eine aussagekräftige Fehlermeldung eingeführt und das Problem mit dem Absturz von Aufgaben in Situationen behoben, in denen die DMS-Aufgabe aufgrund eines falschen TDE-Passworts für Oracle als Quelle fehlschlägt und keine Fehlermeldung angezeigt wird. |
| Ermöglichung der Migration von PostgreSQL CTAS (Create table as selected) DDL während CDC.                                        | Die Einschränkungen, dass DMS PostgreSQL CTAS (Create table as selected)-DDLs während CDC nicht replizieren kann, wurden beseitigt.                                                                                                                                  |

| Thema                                                                                                             | Auflösung                                                                                                                                                                                                                                                                                                           |  |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Das Problem mit einem Absturz der pg_logical-Aufgabe, wenn Tabellenspalten im CDC gelöscht werden, wurde behoben. | Es wurde ein Problem mit PostgreSQL als Quelle und S3 als Ziel behoben, bei dem Spalten auf dem Ziel falsch ausgerichtet sind, wenn die Unterstützung für LOBs deaktiviert ist und LOBs vorhanden sind.                                                                                                             |  |
| Behebung des Speicherlecks beim Umgang mit MySQL-Verbindungen                                                     | Es wurde ein Problem mit MySQL als Quelle behoben, bei dem der Speicherverbrauch für Aufgaben kontinuierlich zunimmt.                                                                                                                                                                                               |  |
| Einstellung des Oracle-Quellendpunkts – ConvertTimestampWithZoneToUTC                                             | Setzen Sie dieses Attribut auf <code>true</code> , um den Zeitstempelwert der Spalten „TIMESTAMP WITH TIME ZONE“ und „TIMESTAMP WITH LOCAL TIME ZONE“ in UTC zu konvertieren. Standardmäßig ist der Wert dieses Attributs „false“ und die Daten werden unter Verwendung der Zeitzone der Quelldatenbank repliziert. |  |
| Oracle-Quelle – DataTruncationErrorPolicy zu SUSPEND_TABLE funktioniert nicht                                     | Es wurde ein Problem mit Oracle als Quelle und S3 als Ziel behoben, bei dem Tabellen nicht gesperrt werden, während die DataTruncationErrorPolicy - Aufgabeneinstellung auf SUSPEND_TABLE gesetzt ist.                                                                                                              |  |
| SQL Server schlägt beim Erstellen der Abfrageklausel bei einem langen Schema/einer langen Tabelle fehl            | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem die Aufgabe fehlschlägt oder nicht mehr reagiert, wenn die Auswahlregel eine durch Kommata getrennte Tabellenliste enthält.                                                                                                                         |  |


| Thema                                                                                                               | Auflösung                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Secret-Manager-Authentifizierung mit MongoDB-Endpoint                                                               | Es wurde ein Problem mit MongoDB- und DocumentDB-Endpunkten behoben, bei dem die Secret-Manager-basierte Authentifizierung nicht funktioniert.                                                                              |
| DMS kürzt die Daten während CDC für eine Multibyte-Varchar-Spalte, wenn NLS_NCHAR_CHARACTERSET auf UTF8 gesetzt ist | Es wurde ein Problem mit Oracle als Quelle und Oracle als Ziel behoben, bei dem Daten für Multibyte-VARCHAR-Spalten gekürzt wurden, bei denen NLS_NCHAR_CHARACTERSET auf UTF8 gesetzt war.                                  |
| <code>filterTransactionsOfUser</code> ECA für Oracle LogMiner                                                       | Es wurde ein zusätzliches Verbindungsattribut (ECA) <code>filterTransactionsOfUser</code> hinzugefügt, damit DMS Transaktionen von einem bestimmten Benutzer ignorieren kann, wenn von Oracle aus repliziert wird. LogMiner |
| Behebbarer Fehler bei der Einstellung von SQL Server, wenn LSN im Backup fehlt                                      | Es wurde ein Problem mit SQL Server behoben, bei dem eine Aufgabe bei fehlendem LSN nicht fehlschlägt.                                                                                                                      |

## AWS Versionshinweise zu Database Migration Service 3.4.7

Die folgende Tabelle zeigt die neuen Funktionen und Verbesserungen, die in AWS Database Migration Service (AWS DMS) Version 3.4.7 eingeführt wurden.

| Neue Funktion oder Änderung          | Beschreibung                                                                                                                                                                                                       |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unterstützung von Babelfish als Ziel | AWS DMS unterstützt jetzt Babelfish als Target. Mit dieser AWS DMS Funktion können Sie jetzt Live-Daten von jeder AWS DMS unterstützten Quelle auf einen Babelfish migrieren, und das mit minimalen Ausfallzeiten. |

| Neue Funktion oder Änderung                                            | Beschreibung                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                        | <p>Weitere Informationen finden Sie unter <a href="#">Verwenden von Babelfish als Ziel für AWS Database Migration Service</a>.</p>                                                                                                                                                                                                                       |
| Unterstützung von IBM-Db2-z/OS-Datenbanken als Quelle nur für Volllast | <p>AWS DMS unterstützt jetzt IBM Db2 z/OS-Datenbanken als Quelle. Mit dieser AWS DMS Option können Sie jetzt Live-Migrationen von Db2-Mainframes zu jedem unterstützten Ziel durchführen. AWS DMS</p> <p>Weitere Informationen finden Sie unter <a href="#">Verwendung von Datenbanken in IBM Db2 für z/OS als Quelle für AWS DMS</a>.</p>               |
| Unterstützung von SQL-Server-Lesereplikaten als Quelle                 | <p>AWS DMS unterstützt jetzt SQL Server Read Replica als Quelle. Mit dieser AWS DMS Funktion können Sie jetzt Live-Migrationen von der SQL Server-Read Replica zu jedem unterstützten Ziel durchführen. AWS DMS</p> <p>Weitere Informationen finden Sie unter <a href="#">Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS</a>.</p> |
| Support Sie EventBridge DMS-Veranstaltungen                            | <p>AWS DMS unterstützt die Verwaltung von Veranstaltungsabonnements mithilfe von EventBridge DMS-Ereignissen.</p> <p>Weitere Informationen finden Sie unter <a href="#">Arbeiten mit Amazon-Event-Bridge-Ereignissen und Benachrichtigungen in AWS Database Migration Service</a>.</p>                                                                   |

| Neue Funktion oder Änderung                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unterstützung von VPC-Quell- und Zielendpunkten | <p>AWS DMS unterstützt jetzt Amazon Virtual Private Cloud (VPC) -Endpunkte als Quellen und Ziele. AWS DMS kann jetzt eine Verbindung zu jedem AWS Dienst mit VPC-Endpunkten herstellen, wenn explizit definierte Routen zu den Diensten in ihrer AWS DMS VPC definiert sind.</p> <div data-bbox="544 541 1507 997" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Upgrades auf AWS DMS Versionen 3.4.7 und höher erfordern, dass Sie zunächst die Verwendung von VPC-Endpunkten oder öffentlichen Routen konfigurieren AWS DMS . Diese Anforderung gilt für Quell- und Zielendpunkte für Amazon S3, Amazon Kinesis Data Streams, Amazon DynamoDB AWS Secrets Manager, Amazon Redshift und Amazon Service. OpenSearch</p></div> <p>Weitere Informationen finden Sie unter <a href="#">Konfigurieren von VPC-Endpunkten als Quell- und Zielendpunkte in AWS DMS</a>.</p> |
| Neue PostgreSQL-Version                         | PostgreSQL Version 14.x wird jetzt als Quelle und Ziel unterstützt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Unterstützung von Aurora Serverless v2 als Ziel | <p>AWS DMS unterstützt jetzt Aurora Serverless v2 als Ziel. Mit AWS DMS können Sie jetzt Live-Migrationen zu Aurora Serverless v2 durchführen.</p> <p>Informationen zu unterstützten AWS DMS Zielen finden Sie unter <a href="#">Ziele für die Datenmigration</a></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Neue Funktion oder Änderung        | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Neue Versionen von IBM Db2 für LUW | <p>AWS DMS unterstützt jetzt IBM Db2 für LUW-Versionen 11.5.6 und 11.5.7 als Quelle. Mit dieser AWS DMS Funktion können Sie jetzt Live-Migrationen von den neuesten Versionen von IBM DB2 for LUW durchführen.</p> <p>Hinweise zu AWS DMS Quellen finden Sie unter <a href="#">Quellen für die Datenmigration</a></p> <p>Informationen zu unterstützten AWS DMS Zielen finden Sie unter <a href="#">Ziele für die Datenmigration</a>.</p> |

AWS DMS 3.4.7 beinhaltet das folgende neue oder geänderte Verhalten und behobene Probleme:

- Sie können jetzt ein Datumsformat aus der Tabellendefinition verwenden, um eine Datenzeichenfolge zu einem Datumsobjekt zu analysieren, wenn Sie Amazon S3 als Quelle verwenden.
- Neue Zähler für Tabellenstatistiken sind jetzt verfügbar: `AppliedInserts`, `AppliedDdls`, `AppliedDeletes` und `AppliedUpdates`.
- Sie können jetzt den Standard-Zuordnungstyp wählen, wenn Sie ihn OpenSearch als Ziel verwenden.
- Mit der neuen `TrimSpaceInChar`-Endpunkteinstellung für Oracle-, PostgreSQL- und SQLServer-Quellen können Sie angeben, ob Daten der CHAR- und NCHAR-Datentypen gekürzt werden sollen.
- Die neue `ExpectedBucketOwner`-Endpunkteinstellung für Amazon S3 verhindert Sniping, wenn S3 als Quelle oder Ziel verwendet wird.
- Für RDS SQL Server, Azure SQL Server und selbstverwaltetes SQL Server bietet DMS jetzt die automatische Einrichtung von MS-CDC für alle für eine Migrationsaufgabe ausgewählten Tabellen mit oder ohne PRIMARY KEY oder mit einem eindeutigen Index, wobei die Aktivierungspriorität für MS-REPLICATION auf selbstverwalteten SQL-Server-Tabellen mit PRIMARY KEY berücksichtigt wird.
- Es wurde Unterstützung für die Replikation von DDL-Vorgängen auf Oracle-Partitionen und Unterpartitionen bei homogenen Oracle-Migrationen hinzugefügt.

- Es wurde ein Problem behoben, bei dem eine Datenvalidierungsaufgabe mit einem zusammengesetzten Primärschlüssel abstürzt, während Oracle als Quelle und Ziel verwendet wird.
- Es wurde ein Problem behoben, bei dem ein variierender Zeichentyp korrekt in einen booleschen Wert umgewandelt wird, während die Zielspalte zuvor als boolescher Wert erstellt wurde, wenn Redshift als Ziel verwendet wird.
- Es wurde ein Problem behoben, das aufgrund eines bekannten ODBC-Problems bei der Verwendung von PostgreSQL als Ziel zu `varchar`-Datenkürzungen für als `varchar(255)` migrierte Datentypen führt.
- Es wurde ein Problem behoben, bei dem der parallele Hinweis für den DELETE-Vorgang nicht berücksichtigt wird, wenn bei Oracle als Ziel `BatchApplyEnabled` auf `true` und `BatchApplyPreserveTransaction` auf `false` gesetzt ist.
- Die neue `AddTrailingPaddingCharacter`-Endpunkteinstellung für Amazon S3 fügt Zeichenkettendaten hinzu, wenn S3 als Ziel verwendet wird.
- Die neue `max_statement_timeout_seconds`-Aufgabeneinstellung verlängert das Standard-Timeout für Endpunktabfragen. Diese Einstellung wird derzeit von Abfragen von MySQL-Endpunktmetadaten verwendet.
- Bei der Verwendung von PostgreSQL als Ziel wurde ein Problem behoben, bei dem eine CDC-Aufgabe die Einstellungen für die Fehlerbehandlungsaufgabe nicht richtig nutzt.
- Es wurde ein Problem behoben, bei dem DMS den Redis-Modus für eine Redis-Enterprise-Instance nicht korrekt identifizieren kann.
- Die Unterstützung von zusätzlichen `includeOpForFullLoad`-Verbindungsattributen (ECA) für das S3-Ziel-Parquet-Format Parquet wurde erweitert.
- Einführung einer neuen PostgreSQL-Endpunkteinstellung `migrateBooleanAsBoolean`. Wenn diese Einstellung für eine Migration von PostgreSQL zu Redshift auf `true` gesetzt ist, wird ein boolescher Wert als `varchar(1)` migriert. Wenn sie auf `false` gesetzt ist, wird ein boolescher Wert als `varchar(15)` migriert, was das Standardverhalten ist.
- Bei Verwendung von SQL Server als Quelle wurde ein Migrationsproblem mit dem Datentyp `datetime` behoben. Mit dieser Korrektur wird das Problem des Einfügens von `Null` behoben, wenn die Genauigkeit in Millisekunden angegeben ist.
- Für PostgreSQL als Quelle mit `PGLOGICAL` wurde ein Migrationsproblem behoben, das bei der Verwendung von `pglogical` und beim Entfernen eines Felds aus der Quelltable während der CDC-Phase auftritt, in der der Wert nach dem entfernten Feld nicht in die Zieltabelle migriert wird.
- Es wurde ein Problem mit der SQL-Server-Loopback-Migration behoben, bei der während der bidirektionalen Replikation wiederholte Datensätze abgerufen werden.



- Es wurde eine neue ECA `mapBooleanAsBoolean` für PostgreSQL als Quelle hinzugefügt. Mit diesem zusätzlichen Verbindungsattribut können Sie die standardmäßige Datentypzuordnung eines Booleschen PostgreSQL-Datentyps zu einem booleschen Datentyp überschreiben. RedShift
- Es wurde ein Migrationsproblem bei der Verwendung von SQL Server als Quelle behoben, das dazu führt, dass `ALTER DECIMAL/NUMERIC SCALE` nicht auf Ziele repliziert wird.
- Behebung eines Verbindungsproblems mit SQL Server 2005.
- Seit dem 17. Oktober 2022 unterstützt DMS 3.4.7 jetzt Amazon-EC2-Instance-Klassen der 6. Generation für Replikations-Instances.
- Seit dem 25. November 2022 können Sie mit DMS 3.4.7 Datenbankschemata und Codeobjekte mithilfe der DMS Schema Conversion konvertieren und Datenbanken in Ihrer Netzwerkumgebung ermitteln, die sich gut für eine Migration mit DMS Fleet Advisor eignen.
- Seit dem 25. November 2022 ist DMS Studio eingestellt.
- Seit dem 31. Januar 2023 unterstützt DMS Schema Conversion Aurora MySQL und Aurora PostgreSQL als Zieldatenanbieter.
- Seit dem 6. März 2023 können Sie mit DMS Fleet Advisor zielgerichtete Empfehlungen für Ihre Quelldatenbanken generieren.
- AWS DMS unterstützt seit dem 6. März 2023 die AWS verwaltete Richtlinie, die die Veröffentlichung von metrischen Datenpunkten auf Amazon ermöglicht CloudWatch.

#### In der Wartungsversion von DMS 3.4.7 vom 5. Mai 2023 behobene Probleme

| Thema                                           | Auflösung                                                                                                                                                                                 |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fehler bei der PostgreSQL-Quellaufrage          | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem Aufgaben fehlschlagen, wenn die maximal zulässige Anzahl von DDL-Vorgängen in einem einzigen Ereignis überschritten wird. |
| PostgreSQL-Quelldatenvalidierung falsch positiv | Es wurde ein Problem mit PostgreSQL als Quelle und Oracle als Ziel behoben, bei dem eine falsche Umwandlung des Timestamp-Felds zu falsch positiven Datenvalidierungsfehlern führt.       |
| Umgang mit MySQL-Quellfehlern                   | Es wurde ein Problem mit MySQL als Quelle behoben, bei dem die DMS-Aufgabe nicht                                                                                                          |

| Thema                                                               | Auflösung                                                                                                                                                                                                      |
|---------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                     | fehlschlägt, wenn das nächste BIN-Protokoll nicht verfügbar ist.                                                                                                                                               |
| ROTATE_EVENT-Protokollierung bei MySQL als Quelle                   | Die Protokollierung für MySQL als Quelle im Zusammenhang mit ROTATE_EVENT wurde verbessert – einschließlich des Lesens des BIN-Protokollnamens.                                                                |
| Timeout-Problem bei Datenvalidierung                                | Es wurde ein Problem mit dem Datenvalidierungs-Feature behoben, bei dem die <code>executeTimeout</code> -Endpunkt-Einstellung bei Abfragen im Zusammenhang mit der Datenvalidierung nicht berücksichtigt wird. |
| Problem bei paralleler Volllast des PostgreSQL-Ziels                | Es wurde ein Problem mit PostgreSQL als Ziel behoben, bei dem die segmentierte (parallele) Volllast aufgrund eines Fehlers „Verbindung unterbrochen“ fehlschlägt.                                              |
| Problem beim Verschieben von DMS-Aufgaben                           | Es wurde ein Problem mit S3 als Ziel behoben, bei dem ein DMS-Vorgang zum Verschieben von Aufgaben sehr lange dauert oder nie abgeschlossen wird.                                                              |
| Problem mit doppelten Datensätzen bei PostgreSQL als Quelle         | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem eine DMS-Aufgabe nach dem Anhalten und Fortsetzen einer Aufgabe Fehler im Zusammenhang mit Duplikaten auf dem Ziel auslöst.                    |
| Falsch positive Ergebnisse bei der Überprüfung von Oracle-Zieldaten | Es wurde ein Problem mit Oracle als Ziel behoben, bei dem bei der Datenvalidierung aufgrund einer falsch replizierten Zeitzone für Zeitstempelfelder falsch positive Fehler gemeldet werden.                   |

## In der Wartungsversion von DMS 3.4.7 vom 22. Februar 2023 behobene Probleme

| Thema                                                         | Auflösung                                                                                                                                                                                                                                                                                                   |  |
|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Replikate von SQL Server AG als Quelle                        | Unterstützung für die SQL Server-Quelle in einer AlwaysOnKonfiguration hinzugefügt, in der sich der Listener-TCP-Port vom TCP-Replikat-Port unterschied.                                                                                                                                                    |  |
| Datenverlust bei Amazon Redshift als Ziel                     | Es wurde ein Problem mit Redshift als Ziel behoben, bei dem in seltenen Fällen ein unerwarteter Redshift-Neustart zu fehlenden Daten auf dem Ziel führen kann.                                                                                                                                              |  |
| Unterstützung für SQL Server Source Safeguard                 | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem die DMS-Aufgabe mit einem Fehler fehlschlagen kann, der darauf hinweist, dass Transaktionsprotokollsicherungen nicht gelesen werden können, wenn die Endpunkteinstellung "SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION" angegeben ist. |  |
| Fehler bei der Datenvalidierungsaufgabe mit Oracle als Quelle | Es wurde ein Problem mit Oracle als Quelle behoben, bei dem die DMS-Aufgabe bei der Datenvalidierung aufgrund falsch identifizierter Primärschlüsselwerte fehlschlagen kann.                                                                                                                                |  |
| Problem mit Kinesis-Bilddaten                                 | Es wurde ein Problem mit Streaming-Zielen (Kinesis, Kafka) behoben, bei dem die "EnableBeforeImage" -Aufgabeneinstellung nur für Zeichendatentypen funktioniert.                                                                                                                                            |  |
| Time-Travel-Protokolldateien                                  | Es wurde ein Problem mit dem Time-Travel-Feature behoben, bei dem DMS Time-Travel-Protokolldateien mit Byte-Zeit Null erstellt, wenn die Quelle inaktiv ist.                                                                                                                                                |  |

## In der Wartungsversion von DMS 3.4.7 vom 16. Dezember 2022 behobene Probleme

| Thema                                                                       | Auflösung                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BatchApplyEnabled                                                           | Ein Problem mit übermäßiger Protokollierung wurde behoben, wenn der Wert auf BatchApplyEnabled True gesetzt war.                                                                                                                                           |
| Neue MongoDB-Endpunkteinstellung—FullLoadNoCursorTimeout                    | Die MongoDB-Endpunkteinstellung FullLoadNoCursorTimeout gibt NoCursorTimeout für den Vollladecursor an. NoCursorTimeout ist eine MongoDB-Verbindungseinstellung, die verhindert, dass der Server den Cursor schließt, wenn er inaktiv ist.                 |
| MongoDB – Filterfunktion für Einzelspaltensegmentierung                     | Die neue Filterfunktion verbessert die Leistung bei der Migration von MongoDB-Datenbanken mithilfe einer einzigen Spalte für die Segmentierung.                                                                                                            |
| MongoDB zu Redshift                                                         | Wenn bei der Migration von MongoDB zu Redshift die MongoDB-Erfassung den binären Datentyp hat, wurde ein Problem behoben, bei dem DMS die Zieltabelle nicht auf Redshift erstellt.                                                                         |
| Neues MongoDB SocketTimeout MS-Verbindungsattribut                          | Das neue zusätzliche Verbindungsattribut MongoDB SocketTimeout MS konfiguriert das Verbindungstimeout für MongoDB-Clients in Einheiten von Millisekunden. Wenn der Wert kleiner oder gleich Null ist, wird der Standardwert des MongoDB-Clients verwendet. |
| Behebung eines Problems, das zum Absturz einer Amazon-Kinesis-Aufgabe führt | Bei der Migration zu Amazon Kinesis Data Streams als Ziel wurde ein Problem bei der Verarbeitung von Nullwerten behoben, wenn kein Primärschlüssel in der Tabelle vorhanden ist.                                                                           |
| Unterstützung von Oracle-NUL-UK-Datenvalidierung                            | Die Einschränkung, dass die Datenvalidierung von NUL-UK-Werten nicht unterstützt wird, wurde beseitigt.                                                                                                                                                    |

| Thema                                                                                      | Auflösung                                                                                                                                                        |
|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Oracle zu Amazon S3                                                                        | Bei der Migration von Oracle zu Amazon S3 wurde ein Problem behoben, bei dem einige Datensätze fälschlich herweise als NULL migriert wurden.                     |
| Oracle Standby                                                                             | Bei der Verwendung von Oracle Standby als Quelle wurde die Möglichkeit hinzugefügt, dass DMS offene Transaktionen verarbeiten kann.                              |
| Migration von Oracle zu Oracle mit dem räumlichen Datentyp SDO_GEOMETRY                    | Bei der Migration von Oracle zu Oracle wurde ein Problem behoben, bei dem die Aufgabe fehlschlägt, wenn die Tabelle eine SDO_GEOMETRY-Spalte in der DDL enthält. |
| Oracle als Quelle                                                                          | Bei Verwendung von Oracle als Quelle wurde ein Problem behoben, bei dem DMS gelegentlich eine Oracle-Redo-Protokoll-Sequenznummer übersprungen hat.              |
| Oracle als Quelle – fehlende Archiv-/Online-Redo-Protokolle                                | Bei Verwendung von Oracle als Quelle wurde ein Problem behoben, bei dem die DMS-Aufgabe fehlschlägt, wenn Archivprotokolle fehlen.                               |
| Problembhebung – DMS überspringt gelegentlich das Oracle-Standby-Redo-Protokoll            | Bei Verwendung von Oracle als Quelle wurde ein Problem behoben, bei dem DMS gelegentlich eine Oracle-Redo-Protokoll-Sequenznummer übersprungen hat.              |
| Problembhebung – Räumliche Datentypen werden bei CDC nicht von Oracle zu Oracle repliziert | Bei der Replikation von Oracle zu Oracle wurde ein Problem behoben, bei dem räumliche Datentypen bei CDC nicht repliziert werden.                                |

| Thema                                                                                                                | Auflösung                                                                                                                                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Oracle als Ziel                                                                                                      | Bei der Verwendung von Oracle als Ziel wurde ein Problem behoben, bei dem die Zielanwendung mit einem ORA-01747-Fehler fehlschlägt.                                                                                                                                                      |
| Amazon S3 – Datenverlust beim erneuten Laden von Tabellen behoben                                                    | Bei der Verwendung von Amazon S3 als Ziel wurde ein Problem behoben, bei dem beim Neuladen einer Tabelle keine CDC-Dateien generiert werden.                                                                                                                                             |
| Problembhebung – Kontextinitialisierung von SQL Server Always On, falls der primäre Server als Quelle verwendet wird | Bei Verwendung von SQL Server Always On als Quelle wurde ein Problem behoben, das dazu führte, dass Availability Groups (AG) nicht initialisiert wurden, wenn die Quelle primär ist und auf true gesetzt ist. AlwaysOnSharedSyncedBackupsEnabled                                         |
| Aktualisierung der SQL-Server-Endpointinstellung                                                                     | Wenn es sich bei einem Quellendpunkt um eine SQL Server Always-On-Verfügbarkeitsgruppe handelt und es sich um ein sekundäres Replikat handelt, wurde ein Problem behoben, bei dem die Replikationsaufgabe fehlschlägt, wenn sie auf AlwaysOnSharedSyncedBackupsEnabled True gesetzt ist. |
| PostgreSQL als Quelle                                                                                                | Es wurde ein Problem behoben, bei dem CDC Lösch-/Aktualisierungsvorgänge auf der PostgreSQL-Quelle, die in 3.4.7 zur Unterstützung von Boolean eingeführt wurde, nicht migrieren konnte. mapBooleanAs                                                                                    |

## AWS Versionshinweise zu Database Migration Service 3.4.6

Die folgende Tabelle zeigt die neuen Funktionen und Verbesserungen, die in AWS Database Migration Service (AWS DMS) Version 3.4.6 eingeführt wurden.

| Neue Funktion oder Änderung                                        | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AWS DMS Zeitreise                                                  | <p>AWS DMS führt <a href="#">Time Travel</a> ein, eine Funktion, die Kunden Flexibilität bei ihren Protokollierungsmöglichkeiten bietet und ihre Erfahrung bei der Problembeseitigung verbessert. Mit Time Travel können Sie Protokolle mit Amazon S3 speichern und verschlüsseln und die AWS DMS Protokolle innerhalb eines bestimmten Zeitraums anzeigen, herunterladen und verschleiern.</p>                                                       |
| Unterstützung von Microsoft Azure SQL Managed Instance als Quelle  | <p>AWS DMS unterstützt jetzt Microsoft Azure SQL Managed Instance als Quelle. Mithilfe AWS DMS können Sie jetzt Live-Migrationen von Microsoft Azure SQL Managed Instance zu jedem AWS DMS unterstützten Ziel durchführen.</p> <p>Hinweise zu AWS DMS Quellen finden Sie unter <a href="#">Quellen für die Datenmigration</a></p> <p>Informationen zu unterstützten AWS DMS Zielen finden Sie unter <a href="#">Ziele für die Datenmigration</a>.</p> |
| Unterstützung von Google Cloud for MySQL als Quelle                | <p>AWS DMS unterstützt jetzt Google Cloud SQL for MySQL als Quelle. Mithilfe AWS DMS können Sie jetzt Live-Migrationen von Google Cloud SQL for MySQL zu jedem AWS DMS unterstützten Ziel durchführen.</p> <p>Informationen zu AWS DMS Quellen finden Sie unter <a href="#">Quellen für die Datenmigration</a></p> <p>Informationen zu unterstützten AWS DMS Zielen finden Sie unter <a href="#">Ziele für die Datenmigration</a>.</p>                |
| Unterstützung des parallelen Ladens für partitionierte Daten zu S3 | <p>AWS DMS unterstützt jetzt das parallele Laden von partitionierten Daten nach Amazon S3, wodurch die Ladezeiten für die Migration partitionierter Daten aus unterstützten Datenbank-Engine-Quelldaten zu Amazon S3 verbessert werden. Dieses Feature erstellt Amazon-S3-Unterverzeichnisse für jede Partition der Tabelle in der</p>                                                                                                                |

| Neue Funktion oder Änderung                                              | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                          | Datenbankquelle, so dass AWS DMS parallel Prozesse ausführen kann, um jeden Unterordner aufzufüllen.                                                                                                                                                                                                                                                                                                                                                          |
| Unterstützung mehrerer Apache-Kafka-Zielthemen in einer einzigen Aufgabe | AWS DMS unterstützt jetzt Apache Kafka-Ziele mit mehreren Themen mit einer einzigen Aufgabe. Mit AWS DMS können Sie jetzt mehrere Schemata aus einer einzigen Datenbank mit derselben Aufgabe auf verschiedene Apache-Kafka-Zielthemen replizieren. Dadurch entfällt die Notwendigkeit, mehrere separate Aufgaben in Situationen zu erstellen, in denen viele Tabellen aus derselben Quelldatenbank zu verschiedenen Kafka-Zielthemen migriert werden müssen. |

In AWS DMS 3.4.6 wurden unter anderem die folgenden Probleme behoben:

- Es wurde ein Problem behoben, bei dem Spalten aus UPDATE-Anweisungen mit falschen Spalten gefüllt werden, wenn die Primärschlüsselspalte nicht die erste Spalte ist, wenn Amazon S3 als Ziel im CSV-Format verwendet wurde.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben abstürzen konnten, wenn das pglogical-Plugin mit NULL Werten in BYTEA Spalten im eingeschränkten LOB-Modus verwendet wurde, wenn PostgreSQL als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben abstürzen konnten, wenn eine große Anzahl von Quelltabellen gelöscht wurde, wenn PostgreSQL als Quelle verwendet wurde.
- Die datumsbasierte Ordnerpartitionierung in Amazon S3 wurde verbessert, indem eine neue Amazon-S3-Einstellung eingeführt wurde, die `DatePartitionTimezone` die Partitionierung an Nicht-UTC-Daten ermöglicht.
- Die Zuordnung zwischen Daten des Typs `TIMESTAMP WITH TIME ZONE` von Quellen zu `TIMESTAMPTZ` wird bei Verwendung von Redshift als Ziel unterstützt
- Die Leistung von CDC für Aufgaben ohne Platzhalterausswahlregeln bei der Verwendung von MongoDB oder Amazon DocumentDB als Quelle wurde verbessert.
- Es wurde ein Problem behoben, bei dem Schemanamen mit einem Unterstrich, einem Platzhalter und einer Länge von weniger als 8 nicht von AWS DMS -Aufgaben erfasst werden, wenn Db2 LUW als Quelle verwendet wird.



- Es wurde ein Problem behoben, bei dem AWS DMS Instanzen bei einem großen Datenvolumen nicht mehr genügend Speicher hatten, wenn OpenSearch Service als Ziel verwendet wurde.
- Die Leistung der Datenvalidierung wurde dadurch verbessert, dass nur Aufgaben der Vollastvalidierung unterstützt werden.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben nach einem erzwungenen Failover nicht fortgesetzt werden konnten, wenn Sybase als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei dem fälschlicherweise eine Warnung `AWS DMS Invalid BC timestamp was encountered in column` gesendet wurde.

Zu den in der DMS-Wartungsversion 3.4.6 behobenen Problemen gehören die folgenden:

- Es wurde ein Problem behoben, bei dem eine Aufgabe abstürzt, wenn der Massenanwendungsmodus aktiviert ist und Oracle als Quelle und Ziel verwendet wird.
- Es wurde ein Problem behoben, so dass eine Vollladeaufgabe jetzt die `ExecuteTimeout`-Endpunkteinstellung mit PostgreSQL als Quelle ordnungsgemäß verwendet.
- Es wurde ein Problem mit der Migration von Spalten des Datentyps Array behoben, wenn die Aufgabe auf den eingeschränkten LOB-Modus gesetzt ist und PostgreSQL als Quelle verwendet wird.
- Es wurde ein Problem mit der Migration von Zeitstempeln mit einer Zeitzone vor dem 01.01.1970 behoben, wenn PostgreSQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem DMS eine leere Zeichenfolge während der Replikation als Null behandelt, wenn SQL Server als Quelle und Ziel verwendet wird.
- Es wurde ein Problem behoben, so dass jetzt die Endpunkteinstellungen für das Lesen und Schreiben von Session-Timeouts berücksichtigt werden, wenn MySQL-Quelle/Ziel verwendet wird.
- Es wurde ein Problem behoben, bei dem eine DMS-CDC-Aufgabe Vollast-Dateien herunterlädt, wenn Amazon S3 als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem das Protokoll abstürzt, wenn `CdcInsertsAndUpdates` und `PreserveTransactions` beide auf `true` eingestellt sind, wenn Amazon S3 als Ziel verwendet wird.
- Es wurde ein Problem behoben, bei dem eine Aufgabe abstürzte, wenn die Funktion `ParallelApply` \* aktiviert war, aber einige Tabellen keinen Standard-Primärschlüssel hatten, wenn Amazon Kinesis Data Streams als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei dem `StreamArn` bei der Verwendung von Amazon Kinesis Data Streams als Quelle kein Fehler für eine falsche Angabe ausgegeben wurde.

- Es wurde ein Problem behoben, bei dem ein Primärschlüsselwert als leere Zeichenfolge dazu führte, dass eine Aufgabe abstürzte, wenn sie OpenSearch als Ziel verwendet wurde.
- Es wurde ein Problem behoben, bei dem zu viel Festplattenspeicher für die Datenüberprüfung verwendet wird.

### In der Wartungsversion von DMS 3.4.6 vom 13. Dezember 2022 behobene Probleme

| Thema                                                                   | Auflösung                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SAP-ASE-ODBC-Treiber                                                    | Es wurde ein Problem mit SAP ASE als Quelle behoben, so dass der ODBC-Treiber Zeichensätze unterstützen kann.                                                                                                                                                        |
| SQL-Server-Datetime-Primärschlüsselfehler für die LOB-Suche             | Es wurde ein Problem mit SQL Server als Quelle behoben, bei dem die LOB-Suche nicht richtig funktioniert, wenn der Primärschlüssel den Datentyp Datum/Uhrzeit mit einer Genauigkeit in Millisekunden hat.                                                            |
| SQL Server zu Redshift – „datetime offset“ wird „timestampz“ zugeordnet | Für Migrationen von SQL Server zu Redshift wurde die Zuordnung verbessert, so dass das SQL-Server-Format „datetimeoffset“ dem Redshift-Format „timestampz“ zugeordnet wird.                                                                                          |
| Datenvalidierung — SkipLobColumns ist wahr                              | Es wurde ein Problem behoben, bei dem die DMS-Aufgabe abstürzt, wenn der Wert True SkipLobColumns ist, in der Quelle ein LOB vorhanden ist, sich der Primärschlüssel in der letzten Spalte befindet und bei der Überprüfung ein Datenunterschied festgestellt wurde. |
| Datenvalidierung mit MySQL als Quelle                                   | Es wurde ein Problem mit MySQL als Quelle mit aktivierter Datenvalidierung behoben, bei dem ein DMS-Aufgabenabsturz auftritt, wenn eine Tabelle mit einem zusammengesetzten eindeutigen Schlüssel mit Nullwerten verwendet wird.                                     |
| MySQL als Quelle                                                        | Es wurde ein Problem mit MySQL als Quelle behoben, bei dem eine Tabelle mit einem Overflow-Fehler                                                                                                                                                                    |

| Thema                                                                | Auflösung                                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                      | gesperrt wird, wenn die Spalten geändert werden, um die Genauigkeit zu erhöhen.                                                                                                                                                                                                                              |
| Upgrade des MySQL-ODBC-Treibers zu 8.0.23                            | Es wurde ein Problem mit MySQL als Quelle behoben, bei dem die Kollation „utf8mb4_0900_bin“ nicht mit dem von DMS verwendeten MySQL-Treiber kompatibel ist.                                                                                                                                                  |
| MySQL – Unterstützung von DDL-Änderungen für partitionierte Tabellen | Es wurde eine neue MySQL-Endpunkteinstellung eingeführt <code>skipTableSuspensionForPartitionDdl</code> , die es dem Benutzer ermöglicht, die Tabellensperre für Partitions-DDL-Änderungen während CDC zu überspringen, sodass DMS jetzt DDL-Änderungen für partitionierte MySQL-Tabellen unterstützen kann. |
| Migration von MongoDB zu Redshift                                    | Es wurde ein Problem bei Migrationen von MongoDB zu Redshift behoben, bei dem DMS keine Zieltabelle in Redshift erstellt, wenn die MongoDB-Erfassung den Datentyp „Binär“ hat.                                                                                                                               |
| Redshift als Ziel – Time Travel Segfault bei Bulk-Anwendung          | Es wurde ein Problem für Redshift als Ziel behoben, bei dem die DMS-Aufgabe abstürzte, wenn sie auf <code>true BatchApplyEnabled</code> gesetzt war.                                                                                                                                                         |
| Redshift als Ziel                                                    | Es wurde ein Problem mit Redshift als Ziel behoben, bei dem <code>parallel-load</code> auf <code>type=partitions-auto</code> gesetzt ist, parallele Segmente Massen-CSV-Dateien in dasselbe Tabellenverzeichnis schreiben und sich gegenseitig stören.                                                       |
| Redshift als Ziel                                                    | Es wurde ein Problem mit Redshift als Ziel behoben, bei dem während CDC die Zielspalte den Typ Boolean hat, während die Quelle den Typ Character Varying hat.                                                                                                                                                |
| Redshift als Ziel                                                    | Das Aufgabenprotokoll wurde verbessert, um eine DDL-Änderung zu identifizieren, die nicht auf Redshift als Ziel repliziert werden kann.                                                                                                                                                                      |

| Thema                                                          | Auflösung                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Datenvalidierung mit PostgreSQL                                | Es wurde ein Problem bei der Validierung mit PostgreSQL behoben, bei dem die Validierung fehlschlägt, wenn boolesche Datentypen vorhanden sind.                                                                                                                              |
| PostgreSQL als Quelle                                          | Es wurde ein Problem für PostgreSQL als Quelle behoben, sodass das ExecuteTimeout Feld in den zusätzlichen Verbindungsattributen für die Volllast verwendet wird.                                                                                                            |
| PostgreSQL als Quelle                                          | Es wurde ein Problem mit PostgreSQL als Quelle behoben, so dass eine Aufgabe fehlschlägt, wenn sie länger als 60 min LSNs liest, die höher als der angeforderte Aufgaben-LSN sind, um anzuzeigen, dass es sich um ein Problem mit dem verwendeten Replikations-Slot handelt. |
| PostgreSQL als Quelle – timestamptz vor 1970-01-01             | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem timestamptz vor dem 01.01.1970 während CDC nicht korrekt migriert werden.                                                                                                                                    |
| PostgreSQL als Quelle                                          | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem DMS während CDC Datenwerte des Typs „Character Varying“ kürzte.                                                                                                                                              |
| PostgreSQL als Quelle – Fortsetzung einer angehaltenen Aufgabe | Es wurde ein Problem mit PostgreSQL als Quelle behoben, bei dem beim Fortsetzen einer zuvor angehaltenen Aufgabenwiedergabe eine oder mehrere Transaktionen bei CDC verpasst werden.                                                                                         |
| Amazon S3 als Ziel                                             | Es wurde ein Problem für S3 als Ziel behoben, bei dem der Header der resultierenden CSV-Datei um eine Spalte daneben lag, wenn der Wert wahr AddColumn Name ist und „TimestampColumnName lautet.                                                                             |

| Thema                                                                              | Auflösung                                                                                                                                                                                  |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon S3 als Quelle – Speichernutzungsverhalten in der Volllastphase für Aufgaben | Es wurde ein Problem mit S3 als Quelle behoben, bei dem eine DMS-Aufgabe bei Volllast den belegten Speicher erst freigibt, nachdem die gesamte Tabelle in die Zieldatenbank geladen wurde. |
| Amazon S3 als Ziel – Vorgang des erneuten Ladens von Tabellen                      | Es wurde ein Problem mit S3 als Ziel behoben, bei dem beim Neuladen einer Tabelle keine CDC-Dateien generiert werden.                                                                      |

## AWS Versionshinweise zu Database Migration Service 3.4.5

Die folgende Tabelle zeigt die neuen Funktionen und Verbesserungen, die in AWS Database Migration Service (AWS DMS) Version 3.4.5 eingeführt wurden.

| Neue Funktion oder Änderung                                                 | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unterstützung für Redis als Ziel                                            | AWS DMS unterstützt jetzt Redis als Ziel. Mit dieser AWS DMS Option können Sie jetzt Live-Daten von jeder AWS DMS unterstützten Quelle in einen Redis-Datenspeicher migrieren, und das mit minimalen Ausfallzeiten. Informationen zu AWS DMS Zielen finden Sie unter <a href="#">Ziele für die Datenmigration</a> .                                                                         |
| Unterstützung für MongoDB 4.2 und 4.4 als Quellen                           | AWS DMS unterstützt jetzt MongoDB 4.2 und 4.4 als Quellen. Mit dieser AWS DMS Option können Sie jetzt Daten von MongoDB 4.2- und 4.4-Clustern auf jedes AWS DMS unterstützte Ziel migrieren, einschließlich Amazon DocumentDB (mit MongoDB-Kompatibilität), mit minimalen Ausfallzeiten. Informationen zu AWS DMS Quellen finden Sie unter <a href="#">Quellen für die Datenmigration</a> . |
| Unterstützung für mehrere Datenbanken bei Verwendung von MongoDB als Quelle | AWS DMS unterstützt jetzt die Migration mehrerer Datenbanken in einer Aufgabe mit MongoDB als Quelle. Mithilfe von AWS DMS können Sie jetzt mehrere Datenbanken eines MongoDB-Clusters gruppieren und sie mithilfe einer Datenbankmigrationsaufgabe migrieren. Sie können mit minimaler Ausfallzeit zu jedem AWS DMS                                                                        |

| Neue Funktion oder Änderung                                                                                             | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                         | unterstützten Ziel migrieren, einschließlich Amazon DocumentDB (mit MongoDB-Kompatibilität).                                                                                                                                                                                                                                                                                                                                       |
| Unterstützung für automatische Segmentierung mit MongoDB oder Amazon DocumentDB (mit MongoDB-Kompatibilität) als Quelle | AWS DMS unterstützt jetzt die automatische Segmentierung mit MongoDB oder Amazon DocumentDB als Quelle. Mithilfe können Sie Datenbankmigrationsaufgaben so konfigurieren AWS DMS, dass die Sammlung eines MongoDB- oder DocumentDB-Clusters automatisch segmentiert wird. Anschließend können Sie die Segmente mit minimaler Ausfallzeit parallel zu jedem AWS DMS unterstützten Ziel migrieren, einschließlich Amazon DocumentDB. |
| Leistungsverbesserung bei Amazon Redshift mit Volllast                                                                  | AWS DMS unterstützt jetzt die Verwendung parallel Threads, wenn Amazon Redshift als Ziel bei Volllast verwendet wird. Durch die Nutzung der Einstellungen für Multithread-Aufgaben zum Vollladen können Sie die Leistung Ihrer ersten Migration von jeder AWS DMS unterstützten Quelle zu Amazon Redshift verbessern. Informationen zu AWS DMS Zielen finden Sie unter. <a href="#">Ziele für die Datenmigration</a>               |

In AWS DMS 3.4.5 wurden unter anderem die folgenden Probleme behoben:

- Es wurde ein Problem behoben, bei dem Daten nach der Fortsetzung möglicherweise fehlen oder dupliziert sind, wenn PostgreSQL als Quelle mit hoher Transaktionsparallelität verwendet wird.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben mit dem Fehler Relations-ID nicht gefunden... fehlschlagen, wenn PostgreSQL als Quelle verwendet wird und das pglogical-Plugin aktiviert ist.
- Es wurde ein Problem behoben, bei dem VARCHAR-Spalten nicht korrekt repliziert werden, wenn PostgreSQL als Quelle und Oracle als Ziel verwendet wird.
- Es wurde ein Problem behoben, bei dem Löschvorgänge nicht korrekt erfasst werden, wenn der Primärschlüssel nicht die erste Spalte in der Tabellendefinition ist, wenn PostgreSQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem bei Datenbankmigrationsaufgaben LOB-Updates in einer speziellen Metadateneinstellung fehlen, wenn MySQL als Quelle verwendet wird.

- Es wurde ein Problem behoben, bei dem `TIMESTAMP`-Spalten im vollständigen `LOB`-Modus als `DATETIME` behandelt werden, wenn MySQL Version 8 als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben beim Parsen von `NULL DATETIME`-Datensätzen fehlschlagen, wenn MySQL 5.6.4 und höher als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben hängen bleiben, nachdem bei Verwendung von Amazon Redshift als Ziel mit parallelem Anwenden der Fehler Thread wird beendet auftrat.
- Es wurde ein Problem behoben, bei dem Daten möglicherweise verloren gehen können, wenn Datenbankmigrationsaufgaben während des `Batch-Apply-CDCs` die Verbindung zu einem Amazon-Redshift-Zielendpunkt trennen.
- Die Leistung bei Volllast durch Aufruf von `ACCEPTINVCHARS` bei Verwendung von Amazon Redshift als Ziel wurde verbessert.
- Es wurde ein Problem behoben, bei dem doppelte Datensätze repliziert wurden, wenn mit Amazon Redshift als Ziel vom `one-by-one` Modus zum `parallel Anwenden` zurückgekehrt wurde.
- Es wurde ein Problem behoben, bei dem bei Datenbankmigrationsaufgaben die Eigentümerschaft von Amazon-S3-Objekten nicht auf den Bucket-Eigentümer mit `cannedAclForObjects=bucket_owner_full_control` umgestellt wird, wenn Amazon S3 als Ziel verwendet wird.
- Verbessert AWS DMS durch die Unterstützung mehrerer Archivziele mit `ECA`, `additionalArchivedLogDestId` wenn Oracle als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben beim Aktualisieren einer `LOB`-Spalte im vollständigen `LOB`-Modus mit dem Fehler `OCI_INVALID_HANDLE` fehlschlagen.
- Es wurde ein Problem behoben, bei dem `NVARCHAR2`-Spalten während des `CDC` nicht korrekt migriert werden, wenn Oracle als Quelle verwendet wird.
- Verbessert AWS DMS durch Aktivierung `SafeguardPolicy` bei Verwendung von RDS for SQL Server als Quelle.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben einen Fehler bei `rdsadmin` melden, wenn eine andere SQL-Server-Quelle als RDS verwendet wird.
- Es wurde ein Problem behoben, bei dem die Datenvalidierung mit `UUID` als Primärschlüssel in einer Partitionseinstellung fehlschlägt, wenn SQL Server als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Volllast- und `CDC`-Aufgaben fehlschlagen können, wenn der erforderliche `LSN` nicht im Datenbankprotokoll gefunden werden kann, wenn Db2 LUW als Quelle verwendet wird.

- Verbessert AWS DMS durch die Unterstützung benutzerdefinierter CDC-Zeitstempel bei der Verwendung von MongoDB als Quelle.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben beim Anhalten hängen bleiben, wenn MongoDB als Quelle verwendet wird und der MongoDB-Treiber Fehler bei `endSessions` aufweist.
- Es wurde ein Problem behoben, bei AWS DMS dem nicht-primäre Felder nicht aktualisiert werden konnten, wenn DynamoDB als Ziel verwendet wurde
- Es wurde ein Problem behoben, bei dem bei der Datenvalidierung falsch positive Abweichungen zwischen den Spalten CLOB und NCLOB gemeldet werden.
- Es wurde ein Problem behoben, bei dem die Datenvalidierung von Nur-Whitespace-Spalten fehlschlägt, wenn Oracle als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben abstürzen, wenn eine partitionierte Tabelle gekürzt wird.
- Es wurde ein Problem behoben, bei dem Datenbankmigrationsaufgaben beim Erstellen der `awsdms_apply_exceptions`-Steuertabelle fehlschlagen.
- Erweiterte Unterstützung des `caching_sha2_password`-Authentifizierungs-Plugins bei Verwendung von MySQL Version 8.

## AWS Versionshinweise zu Database Migration Service 3.4.4

Die folgende Tabelle zeigt die neuen Features und Verbesserungen, die in AWS DMS Version 3.4.4 eingeführt wurden.

| Neue Funktion oder Änderung                                                                       | Beschreibung                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unterstützung der TLS-Verschlüsselung und der TLS- oder SASL-Authentifizierung bei Kafka als Ziel | AWS DMS unterstützt jetzt TLS-Verschlüsselung und TLS- oder SASL-Authentifizierung mit Amazon MSK und einem lokalen Kafka-Cluster als Ziel. Weitere Informationen zur Verwendung von Verschlüsselung und Authentifizierung für Kafka-Endpunkte finden Sie unter <a href="#">Herstellen einer Verbindung zu Kafka mit Transport Layer Security (TLS)</a> . |

In AWS DMS 3.4.4 wurden unter anderem die folgenden Probleme behoben:



- Die AWS DMS Protokollierung von Task-Fehlern bei der Verwendung von Oracle-Endpunkten wurde verbessert.
- AWS DMS Die verbesserte Aufgabenausführung setzt die Verarbeitung fort, wenn Oracle-Quellendpunkte nach einem Failover von Oracle Data Guard die Rollen wechseln.
- Die verbesserte Fehlerbehandlung behandelt ORA-12561 bei der Verwendung von Oracle-Endpunkten als behebbaren Fehler.
- Es wurde ein Problem behoben, bei dem die Spalten `EMPTY_BLOB()` und `EMPTY_CLOB()` während des CDC nicht korrekt migriert werden, wenn Oracle als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben Datensätze nicht aktualisieren konnten, nachdem die DDL zum Hinzufügen von Spalten geändert wurde, wenn SQL Server als Quelle verwendet wurde.
- Verbesserung der Migration mit PostgreSQL als Quelle durch Unterstützung des `TIMESTAMP WITH TIME ZONE`-Datentyps.
- Es wurde ein Problem behoben, bei dem die `afterConnectScript`-Einstellung bei Volllast nicht funktioniert, wenn PostgreSQL als Ziel verwendet wird.
- Es wurde eine neue `mapUnboundedNumericAsString`-Einstellung eingeführt, um den `NUMERIC`-Datentyp ohne Genauigkeit und Skalierung bei der Verwendung von PostgreSQL-Endpunkten besser handhaben zu können.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben mit „0 betroffenen Zeilen“ fehlschlagen, nachdem sie gestoppt und wieder aufgenommen wurden, wenn PostgreSQL als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei AWS DMS dem der `TIMESTAMP` Datentyp mit dem `BC` Suffix nicht migriert werden konnte, wenn PostgreSQL als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei AWS DMS dem der `TIMESTAMP` Wert „`±infinity`“ nicht migriert werden konnte, wenn PostgreSQL als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei dem leere Zeichenfolgen als `NULL` behandelt werden, wenn S3 als Quelle verwendet wird die `csvNullValue`-Einstellung auf andere Werte gesetzt ist.
- Das zusätzliche Verbindungsattribut `timestampColumnName` bei Volllast mit CDC wurde verbessert, so dass CDC sortiert werden kann, wenn S3 als Ziel verwendet wird.
- Die Behandlung von binären Datentypen im Hex-Format wie `BYTE`, `BINARY` und `BLOB` bei Verwendung von S3 als Quelle wurde verbessert.
- Es wurde ein Problem behoben, bei dem gelöschte Datensätze mit Sonderzeichen migriert werden, wenn S3 als Ziel verwendet wird.

- Es wurde ein Problem mit leeren Schlüsselwerten behoben, wenn Amazon DocumentDB (mit MongoDB-Kompatibilität) als Ziel verwendet wird.
- Es wurde ein Problem behoben, bei AWS DMS dem Decimal128 Spalten nicht repliziert werden NumberDecimal konnten, wenn MongoDB oder Amazon DocumentDB (mit MongoDB-Kompatibilität) als Quelle verwendet wurden.
- Es wurde ein Problem behoben, bei dem CDC-Aufgaben wiederholt werden können, wenn es zu einem Failover auf MongoDB oder Amazon DocumentDB (mit MongoDB-Kompatibilität) als Quelle kommt.
- Es wurde eine Option hinzugefügt, um das hexadezimale Präfix „0x“ für RAW Datentypwerte zu entfernen, wenn Kinesis, Kafka oder als Ziel verwendet wird. OpenSearch
- Es wurde ein Problem behoben, bei dem die Validierung von Zeichenspalten mit fester Länge fehlschlägt, wenn Db2 LUW als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem die Validierung fehlschlägt, wenn nur der Quelldatentyp oder der Zieldatentyp FLOAT oder DOUBLE ist.
- Es wurde ein Problem behoben, bei dem die Validierung von NULL-Zeichen fehlschlägt, wenn Oracle als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem die Validierung von XML-Spalten fehlschlägt, wenn Oracle als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben abstürzten, wenn Spalten in zusammengesetzten Schlüsseln, die MySQL als Quelle verwenden, NULL-Werte zulassen.
- Es wurde ein Problem behoben, bei AWS DMS dem sowohl UNIQUEIDENTIFIER Spalten von SQL Server-Quellendpunkten als auch UUID-Spalten von PostgreSQL-Zielendpunkten nicht validiert werden konnten.
- Es wurde ein Problem behoben, bei dem eine CDC-Aufgabe keine aktualisierte Quelltabellendefinition verwendet, nachdem diese geändert wurde.
- Das AWS DMS Failover wurde verbessert, sodass Aufgabenfehler, die durch einen ungültigen Benutzernamen oder ein ungültiges Passwort verursacht wurden, als behebbare Fehler behandelt werden.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben aufgrund fehlender LSNs fehlschlagen, wenn RDS für SQL Server als Quelle verwendet wurde.

## AWS Versionshinweise zu Database Migration Service 3.4.3

Die folgende Tabelle zeigt die neuen Features und Verbesserungen, die in AWS DMS Version 3.4.3 eingeführt wurden.

| Neue Funktion oder Änderung                            | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Neue Amazon-DocumentDB-Version                         | Amazon DocumentDB Version 4.0 wird jetzt als Quelle unterstützt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Neue MariaDB-Version                                   | MariaDB Version 10.4 wird jetzt als Quelle und Ziel unterstützt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Support bei der AWS Secrets Manager Integration        | Sie können die Datenbankverbindungsdetails (Benutzeranmeldinformationen) für unterstützte Endpunkte sicher in AWS Secrets Manager speichern. Sie können dann bei der Erstellung oder Änderung eines Endpunkts den entsprechenden geheimen Schlüssel anstelle von Klartext-Anmeldeinformationen einreichen. AWS DMS stellt dann mithilfe des Geheimnisses eine Verbindung zu den Endpunktdatenbanken her. Weitere Informationen zum Erstellen von Geheimnissen für AWS DMS Endgeräte finden Sie unter <a href="#">Verwenden von Secrets für den Zugriff auf AWS Database Migration Service -Endpunkte</a> . |
| Größere Optionen für C5- und R5-Replikations-Instances | Sie können jetzt die folgenden größeren Replikations-Instances erstellen: C5-Größen bis zu 96 vCPUs und 192 GiB Arbeitsspeicher und R5-Größen bis zu 96 vCPUs und 768 GiB Arbeitsspeicher.                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Leistungsverbesserung bei Amazon Redshift              | AWS DMS unterstützt jetzt paralleles Anwenden, wenn Redshift als Ziel verwendet wird, um die Leistung der laufenden Replikation zu verbessern. Weitere Informationen finden Sie unter <a href="#">Multithread-Aufgabeneinstellungen für Amazon Redshift</a> .                                                                                                                                                                                                                                                                                                                                              |

In AWS DMS 3.4.3 wurden unter anderem die folgenden Probleme behoben:

- Es wurde ein Problem behoben, bei dem der Commit-Zeitstempel für verzögerte Ereignisse zu „1970-01-01 00:00:00“ wird, wenn Db2 LUW als Quelle verwendet wird.


- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben mit einer NVARCHAR Spalte als Primärschlüssel fehlschlugen, wenn SQL Server als Quelle im vollständigen LOB-Modus verwendet wurde.
- Es wurde ein Problem behoben, bei dem Datensätze während der Phase mit zwischengespeicherten Änderungen fehlen, wenn SQL Server als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Datensätze übersprungen wurden, nachdem AWS DMS Aufgaben wieder aufgenommen wurden, wenn RDS for SQL Server als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei dem die AWS DMS ASSERTION-Protokollierungskomponente große Protokolle für SQL Server generiert.
- Es wurde ein Problem behoben, bei dem die Datenvalidierung während der CDC-Phase aufgrund eines Überlaufs der Spaltenanalyse fehlschlägt, wenn MySQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben aufgrund eines Segmentierungsfehlers während der Datenvalidierung abstürzten, wenn PostgreSQL als Ziel verwendet wurde.
- Es wurde ein Problem behoben, bei dem die Datenvalidierung für den DOUBLE-Datentyp während CDC fehlschlägt, wenn PostgreSQL als Quelle und Ziel verwendet wird.
- Es wurde ein Problem behoben, bei dem Datensätze, die mit dem Befehl copy eingefügt werden, nicht korrekt repliziert werden, wenn PostgreSQL als Quelle und Redshift als Ziel verwendet wird.
- Es wurde ein Datenverlustproblem während der Phase zwischengespeicherter Änderungen behoben, wenn PostgreSQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, das möglicherweise zu Datenverlust oder Datensatzduplikaten führen konnte, wenn PostgreSQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Schemata mit gemischter Groß- und Kleinschreibung nicht mit pglogical migriert werden, wenn PostgreSQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem die letzte Fehlermeldung den ORA-Fehler nicht enthält, wenn Oracle als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben keine UPDATE-Anweisungen erstellen konnten, wenn Oracle als Ziel verwendet wurde.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben keine Daten replizierten, wenn Oracle 12.2 als Quelle mit ASM- und Pluggable Database-Konfiguration verwendet wurde.
- Die Analyse von Datensätzen wurde verbessert, indem Anführungszeichen beibehalten werden, um RFC 4180 zu entsprechen, wenn S3 als Quelle verwendet wird.

- Die Handhabung von `timestampColumnName` wurde verbessert, so dass die Spalte aus Full Load zusammen mit der Spalte aus CDC sortiert werden kann.
- Durch die Einführung einer neuen Endpunkteinstellung wurde ein Problem behoben `messageMaxBytes`, bei dem AWS DMS Aufgaben fehlschlagen, wenn LOB-Elemente größer als 1 MB waren.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben aufgrund eines Segmentierungsfehlers abstürzten, wenn Redshift als Ziel verwendet wurde.
- Verbesserte Fehlerprotokollierung für die Redshift-Testverbindung.
- Es wurde ein Problem behoben, bei dem beim Vollladen AWS DMS nicht alle Dokumente von MongoDB nach DocumentDB übertragen wurden.
- Es wurde ein Problem behoben, bei dem AWS DMS Aufgaben schwerwiegende Fehler meldeten, wenn keine Tabellen in den Tabellenzuordnungsregeln enthalten waren.
- Es wurde ein Problem behoben, bei dem Schemata und Tabellen, die vor dem Neustart von AWS DMS -Aufgaben erstellt wurden, nicht auf das Ziel repliziert werden, wenn MySQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Platzhalter-Escape `[_]` den Platzhalter „\_“ in der Ausschlussregel nicht umgehen kann, wenn MySQL als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Spalten des Datentyps UNSIGNED BIGINT nicht korrekt repliziert werden, wenn MySQL als Quelle verwendet wird.

## AWS Versionshinweise zu Database Migration Service 3.4.2

Die folgende Tabelle zeigt die neuen Features und Verbesserungen, die in AWS DMS Version 3.4.2 eingeführt wurden.

| Neue Funktion oder Änderung                                                                                                                                      | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Support für die private Verbindung Ihrer Amazon Virtual Private Cloud (Amazon VPC) mit dem AWS Database Migration Service (DMS), ohne dass ein Internet-Gateway, | Sie können jetzt über einen AWS DMS von Ihnen erstellten VPC-Schnittstellenendpunkt eine Verbindung zu Ihrer Amazon VPC herstellen und von dort aus darauf zugreifen. Dieser Schnittstellenendpunkt ermöglicht es Ihnen, alle Netzwerkaktivitäten Ihrer AWS DMS Replikationsinstanz innerhalb der Amazon-Netzwerkinfrastruktur zu isolieren. Indem Sie in alle API-Aufrufe zur AWS DMS Verwendung des AWS CLI oder eines SDK einen Verweis auf |

| Neue Funktion oder Änderung                                                                         | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung erforderlich ist.</p> | <p>diesen Schnittstellenendpunkt aufnehmen, stellen Sie sicher, dass alle AWS DMS Aktivitäten für das öffentliche Internet unsichtbar bleiben. Weitere Informationen finden Sie unter <a href="#">Sicherheit der Infrastruktur in AWS Database Migration Service</a>.</p> <div data-bbox="544 478 1507 697" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Diese Funktion ist in allen unterstützten AWS DMS Engine-Versionen verfügbar.</p> </div> |
| <p>Datumsbasierte CDC-Ordnerpartitionierung mit Amazon S3 als Ziel</p>                              | <p>AWS DMS unterstützt jetzt die datumsbasierte Ordnerpartitionierung bei der Replikation von Daten mit S3 als Ziel. Weitere Informationen finden Sie unter <a href="#">Verwenden einer datumsbasierten Ordnerpartitionierung</a>.</p>                                                                                                                                                                                                                                                                                                                                                                 |

In AWS DMS 3.4.2 wurden unter anderem die folgenden Probleme behoben:

- Es wurde eine STATUPDATE-Option hinzugefügt, wenn eine Migration mit Redshift als Ziel durchgeführt wird.
- Die Validierungsaufgaben wurden durch die Einführung einer neuen Einstellung verbessert. `ValidQueryCdcDelaySecond` verzögert die erste Validierungsabfrage sowohl auf Quell- als auch auf Zielpunkten, um Ressourcenkonflikte bei hoher Migrationslatenz zu reduzieren.
- Es wurde ein Problem behoben, bei dem das Starten von Validierungsaufgaben lange AWS DMS dauerte.
- Es wurde ein Problem behoben, bei dem leere Datensätze generiert werden, wenn Replikationsaufgaben mit S3 als Ziel gestartet oder angehalten werden.
- Es wurde ein Problem behoben, bei dem Aufgaben nach Abschluss eines Volllast-Vorgangs hängen bleiben.
- Es wurde ein Problem behoben, bei dem Aufgaben hängen bleiben, wenn eine Quelltable Datenfehler aufweist, während S3 als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Aufgaben beim Start hängen bleiben, wenn das Benutzerkonto des Quellendpunkts deaktiviert ist.

- Es wurde ein Problem behoben, bei dem Aufgaben abstürzen, wenn PostgreSQL als Quelle mit `REPLICA IDENTITY FULL` verwendet wurde.
- Es wurde ein Problem behoben, bei dem Aufgaben Transaktionen verpassen, wenn PostgreSQL als Quelle mit dem `pglogical`-Plugin verwendet wird.
- Es wurde ein Problem behoben, bei dem komprimierte Quelldateien AWS DMS nicht gelöscht wurden, wenn Redshift als Ziel verwendet wurde.
- Es wurde ein Problem behoben, bei dem Validierungsaufgaben falsch negative Ergebnisse melden, wenn MySQL sowohl als Quelle als auch als Ziel mit Datentyp `BIGINT UNSIGNED` verwendet wird.
- Es wurde ein Problem behoben, bei dem bei Validierungsaufgaben falsch positive Ergebnisse melden, wenn SQL Server als Quelle mit einer Primärschlüsselspalte vom Typ `CHAR` verwendet wird.
- Es wurde ein Problem behoben, bei dem Zielobjekte AWS DMS nicht gelöscht wurden `start-replication`, wenn Replizierungsaufgaben mit S3 als Ziel gestartet wurden.
- Es wurden mehrere Probleme bei der Datenvalidierung behoben, wenn Db2 als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Validierungsaufgaben hängen bleiben, wenn SQL Server als Quelle mit der `VARCHAR`-Spalte als Primärschlüssel verwendet wird.
- Es wurde Unterstützung für den Datentyp `TIMESTAMP WITH TIMEZONE` hinzugefügt, wenn PostgreSQL als Quelle verwendet wird

## AWS Hinweise zur Beta-Version von Database Migration Service

### 3.4.1

Die folgende Tabelle zeigt die neuen Features und Verbesserungen, die in AWS DMS Version 3.4.1 Beta eingeführt wurden.

| Neue Funktion oder Änderung          | Beschreibung                                                |
|--------------------------------------|-------------------------------------------------------------|
| Neue MongoDB-Version                 | MongoDB Version 4.0 wird jetzt als Quelle unterstützt.      |
| TLS-1.2-Unterstützung für SQL Server | AWS DMS unterstützt jetzt TLS 1.2 für SQL Server-Endpunkte. |

In AWS DMS 3.4.1 Beta wurden unter anderem die folgenden Probleme behoben:

- Verbesserte Unterstützung für Oracle 19c TDE.
- Verbesserte Unterstützung des Zeichensatzes utf8mb4 und des Identitätsdatentyps bei Verwendung von Redshift als Ziel.
- Verbessertes Umgang mit Fehlern bei Replikationsaufgaben bei Verwendung von MySQL als Quelle und Abwesenheit des Binärprotokolls.
- Verbesserte Unterstützung für die Datenvalidierung für verschiedene Datentypen und Zeichensätze.
- Der Umgang mit Nullwerten wurde mit einer neuen Endpunkteinstellung `IncludeNullAndEmpty` verbessert, wenn Kinesis und Kafka als Ziel verwendet werden.
- Verbesserte Fehlerprotokollierung und -behandlung bei der Verwendung von Kafka als Ziel.
- Verbesserter Zeitversatz für die Sommerzeit bei Verwendung von SQL Server als Quelle.
- Es wurde ein Problem behoben, bei dem Replikationsaufgaben versuchen, vorhandene Tabellen für Oracle als Ziel zu erstellen.
- Es wurde ein Problem behoben, bei dem Replikationsaufgaben hängen bleiben, nachdem die Datenbankverbindung unterbrochen wurde, wenn Oracle als Quelle verwendet wird.
- Es wurde ein Problem behoben, bei dem Replikationsaufgaben den neuen primären Server nicht erkennen und erneut eine Verbindung herstellen können, wenn SQL Server als Quelle mit `AlwaysOn`-Einstellung verwendet wurde.
- Es wurde ein Problem behoben, bei dem bei Replikationsaufgaben unter bestimmten Bedingungen kein "D" für die "OP"-Spalte bei Verwendung von S3 als Ziel hinzufügen.


## AWS Hinweise zur Beta-Version von Database Migration Service

### 3.4.0

Die folgende Tabelle zeigt die neuen Features und Verbesserungen, die in AWS DMS Version 3.4.0 eingeführt wurden.

| Neue Funktion oder Änderung | Beschreibung                                                                                                 |
|-----------------------------|--------------------------------------------------------------------------------------------------------------|
| Neue MySQL-Version          | AWS DMS unterstützt jetzt MySQL Version 8.0 als Quelle, außer wenn die Transaktions-Payload komprimiert ist. |



| Neue Funktion oder Änderung                                                        | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TLS 1.2-Unterstützung für MySQL                                                    | AWS DMS unterstützt jetzt TLS 1.2 für MySQL-Endpunkte.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Neue MariaDB-Version                                                               | AWS DMS unterstützt jetzt MariaDB Version 10.3.13 als Quelle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Kein SysAdmin Zugriff auf selbstverwaltete Microsoft SQL Server-Quellen            | <p>AWS DMS unterstützt jetzt den Zugriff von SysAdmin Nichtbenutzern auf lokale und in EC2 gehostete SQL Server-Quellendpunkte.</p> <div data-bbox="544 611 1507 877" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> <b>Note</b></p> <p>Dieses Feature befindet sich derzeit im Beta-Status. Wenn Sie es ausprobieren möchten, wenden Sie sich an den AWS Support, um weitere Informationen zu erhalten.</p> </div> |
| Mit <code>CREATE TABLE AS</code> erstellte CDC-Aufgaben und Oracle-Quellentabellen | AWS DMS unterstützt jetzt sowohl Volllast- als auch CDC- und Nur-CDC-Aufgaben, die in Oracle-Quellentabellen ausgeführt werden, die mit der Anweisung erstellt wurden. <code>CREATE TABLE AS</code>                                                                                                                                                                                                                                                                                                                                                       |

In AWS DMS 3.4.0 wurden unter anderem die folgenden Probleme behoben:

- Verbesserte Aufgabenbewertungen vor der Migration. Weitere Informationen finden Sie unter [Aktivieren und Verwenden von Vormigrationsbewertungen für eine Aufgabe](#).
- Verbesserte Datenvalidierung für Float-, Real- und Double-Datentypen.
- Amazon Redshift als Ziel wurde verbessert, indem der folgende Fehler besser behandelt wurde: „Der angegebene Schlüssel existiert nicht.“
- Unterstützt Einstellungen für CDC-Ladeaufgaben mit mehreren `ThreadsParallelApplyThreads`, einschließlich `ParallelApplyBufferSize`, und `ParallelApplyQueuesPerThread`, für Amazon OpenSearch Service (OpenSearch Service) als Ziel.
- OpenSearch Service als Ziel wurde verbessert, indem die Verwendung zusammengesetzter Primärschlüssel unterstützt wurde.
- Es wurde ein Problem behoben, bei dem die Testverbindung fehlschlägt, wenn PostgreSQL als Quelle verwendet wird und das Passwort Sonderzeichen enthält.

- Es wurde ein Problem bei der Verwendung von SQL Server als Quelle behoben, das auftritt, wenn einige VARCHAR-Spalten gekürzt werden.
- Es wurde ein Problem behoben, bei dem offene Transaktionen AWS DMS nicht geschlossen wurden, wenn Amazon RDS SQL Server als Quelle verwendet wurde. Dies kann zu Datenverlust führen, wenn der Parameter für das Abfrageintervall falsch eingestellt ist. Weitere Informationen zum Einrichten eines empfohlenen Werts für das Abfrageintervall finden Sie unter [Verwenden einer Microsoft SQL Server-Datenbank als Quelle für AWS DMS](#).
- Es wurde ein Problem mit Oracle Standby als Quelle behoben, bei dem CDC-Aufgaben unerwartet beendet werden, wenn Binary Reader verwendet wird.
- Es wurde ein Problem mit IBM DB2 für LUW behoben, bei dem die Aufgabe mit der Meldung „Das numerische Literal 0 ist nicht gültig, weil sein Wert außerhalb des zulässigen Bereichs liegt“ fehlschlägt.
- Es wurde ein Problem bei der Migration von PostgreSQL zu PostgreSQL behoben, bei dem eine neue Spalte zur PostgreSQL-Quelle hinzugefügt wird und die Spalte mit einem anderen Datentyp als demjenigen erstellt wird, für den die Spalte ursprünglich an der Quelle erstellt wurde.
- Es wurde ein Problem mit MySQL als Quelle behoben, bei dem die Migrationsaufgabe unerwartet angehalten wird, wenn Binärprotokolle nicht abgerufen werden können.
- Es wurde ein Problem im Zusammenhang mit einem Oracle-Ziel behoben, wenn BatchApply verwendet wird.
- Es wurde ein Problem mit MySQL und MariaDB bei der Migration des TIME-Datentyps behoben.
- Es wurde ein Problem mit einer IBM DB2-LUW-Quelle behoben, bei dem die Migration von Tabellen mit LOBs fehlschlägt, wenn die Tabellen keinen Primärschlüssel oder eindeutigen Schlüssel haben.

## AWS Versionshinweise zu Database Migration Service 3.3.4

In AWS DMS 3.3.4 wurden unter anderem die folgenden Probleme behoben:

- Es wurde ein Problem behoben, bei dem Transaktionen gelöscht oder dupliziert werden, wenn PostgreSQL als Quelle verwendet wird.
- Die Unterstützung der Verwendung des Dollarzeichens (\$) in Schemanamen wurde verbessert.
- Es wurde ein Problem behoben, bei dem Replikations-Instances offene Transaktionen nicht schließen, wenn Amazon RDS SQL Server als Quelle verwendet wird.

- Es wurde ein Problem behoben, bei dem die Testverbindung fehlschlägt, wenn PostgreSQL als Quelle verwendet wird und das Passwort Sonderzeichen enthält.
- Amazon Redshift als Ziel wurde verbessert, indem der folgende Fehler besser behandelt wurde: „Der angegebene Schlüssel existiert nicht.“
- Verbesserte Unterstützung für die Datenvalidierung für verschiedene Datentypen und Zeichensätze.
- Es wurde ein Problem behoben, bei dem Replikationsaufgaben versuchen, vorhandene Tabellen für Oracle als Ziel zu erstellen.
- Es wurde ein Problem behoben, bei dem bei Replikationsaufgaben unter bestimmten Bedingungen kein "D" für die "OP"-Spalte bei Verwendung von Amazon S3 als Ziel hinzufügen.

## AWS Versionshinweise zu Database Migration Service 3.3.3

Die folgende Tabelle zeigt die neuen Features und Verbesserungen, die in AWS DMS Version 3.3.3 eingeführt wurden.

| Neue Funktion oder Änderung                                                         | Beschreibung                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Neue PostgreSQL-Version                                                             | PostgreSQL Version 12 wird jetzt als Quelle und Ziel unterstützt.                                                                                                                                |
| Support für zusammengesetzte Primärschlüssel mit Amazon OpenSearch Service als Ziel | Ab AWS DMS 3.3.3 wird die Verwendung eines zusammengesetzten Primärschlüssels von OpenSearch Service-Zielen unterstützt.                                                                         |
| Unterstützung für erweiterte Oracle-Datentypen                                      | Erweiterte Oracle-Datentypen für Oracle-Quelle und -Ziele werden jetzt unterstützt.                                                                                                              |
| Höhere Anzahl von AWS DMS Ressourcen pro Konto                                      | Das Limit für die Anzahl der AWS DMS Ressourcen, die Sie erstellen können, wurde erhöht. Weitere Informationen finden Sie unter <a href="#">Kontingente für AWS Database Migration Service</a> . |

In AWS DMS 3.3.3 wurden unter anderem die folgenden Probleme behoben:

- Es wurde ein Problem behoben, bei dem eine Aufgabe mit einer bestimmten Update-Anweisung mit Parallel Apply in Amazon Kinesis abstürzt.
- Es wurde ein Problem behoben, bei dem eine Aufgabe in der ALTER TABLE-Anweisung mit Amazon S3 als Ziel abstürzt.
- Es wurde ein Problem behoben, bei dem Werte in Polygonspalten abgeschnitten wurden, wenn Microsoft SQL Server als Quelle verwendet wurde.
- Es wurde ein Problem beim Unicode-Konverter von JA16SJISTILDE und JA16EUCTILDE behoben, das auftrat, wenn Oracle als Quelle verwendet wurde.
- Es wurde ein Problem behoben, bei dem die Spalten MEDIUMTEXT und LONGTEXT nicht vom MySQL- zum S3 CSV (Comma Separated Value)-Format migriert wurden.
- Es wurde ein Problem behoben, bei dem boolesche Spalten mit Apache-Parquet-Ausgabe in falsche Typen umgewandelt wurden.
- Es wurde ein Problem mit erweiterten varchar-Spalten in Oracle behoben.
- Es wurde ein Problem behoben, bei dem Datenüberprüfungsaufgaben aufgrund bestimmter Zeitstempelkombinationen fehlschlagen.
- Es wurde ein Problem mit der Sybase Data Definition Language (DDL)-Replikation behoben.
- Es wurde ein Problem behoben, bei dem eine RAC-Quelle (Oracle Real Application Clusters) mit Oracle Binary Reader abstürzte.
- Es wurde ein Problem mit der Validierung für Oracle-Ziele mit der Groß-/Kleinschreibung von Schemanamen behoben.
- Es wurde ein Problem mit der Validierung von IBM Db2-Versionen 9.7 und 10 behoben.
- Es wurde ein Problem behoben, bei dem eine Aufgabe bei Aktivierung von StopTaskCachedChangesApplied und StopTaskCachedChangesNotApplied nicht zweimal aktiviert wurde.

# Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen der Dokumentation zu AWS Database Migration seit Januar 2018 beschrieben.

Sie können einen RSS-Feed abonnieren, um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten. Weitere Informationen zu AWS DMS-Versionen finden Sie unter [AWS DMS-Versionshinweise](#).

| Änderung                                                                           | Beschreibung                                                                                                                                                                             | Datum             |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">AWS DMS-Unterstützung für RDS IBM DB2 als Ziel hinzugefügt</a>         | AWS DMS unterstützt jetzt die Verwendung von Amazon RDS IBM DB2 als Ziel.                                                                                                                | 4. Dezember 2023  |
| <a href="#">AWS DMS-Unterstützung für Timestream als Ziel hinzugefügt.</a>         | AWS DMS unterstützt jetzt Timestream als Ziel.                                                                                                                                           | 17. November 2023 |
| <a href="#">AWS DMS-Unterstützung für Redshift-Zielvalidierung hinzugefügt</a>     | AWS DMS unterstützt jetzt die Validierung von Daten in Redshift-Zielen.                                                                                                                  | 14. November 2023 |
| <a href="#">AWS DMS – Unterstützung für vier neue Endpunkttypen</a>                | AWS DMS unterstützt jetzt die Verwendung von Microsoft Azure Database für PostgreSQL, Microsoft Azure Database für MySQL, OCI MySQL Heatwave und Google Cloud für PostgreSQL als Quelle. | 26. Oktober 2023  |
| <a href="#">AWS DMS hat eine neue von AWS serviceverknüpfte Rolle hinzugefügt.</a> | AWS DMS unterstützt jetzt die serviceverknüpfte AWS-Rolle <code>AWSServiceRoleForDMSServerless</code> , mit der AWS DMS Ressourcen für Sie erstellen und verwalten kann,                 | 22. Mai 2023      |

z. B. das Veröffentlichen von metrischen Datenpunkten in Amazon CloudWatch.

[AWS DMS hat eine neue von AWS verwaltete Richtlinie hinzugefügt.](#)

AWS DMS unterstützt jetzt die von AWS verwaltete Richtlinie, die die Veröffentlichung von Serverless-Replikationsprotokollen in CloudWatch Logs ermöglicht.

22. Mai 2023

[AWS DMS hat eine neue von AWS verwaltete Richtlinie hinzugefügt.](#)

AWS DMS unterstützt jetzt die AWS-verwaltete Richtlinie, die die Veröffentlichung von metrischen Datenpunkten in Amazon CloudWatch ermöglicht. Dazu hat AWS DMS mit der Nachverfolgung von Änderungen für seine AWS-verwalteten Richtlinien begonnen.

6. März 2023

[Unterstützung von VPC-Quell- und Zielendpunkten](#)

AWS DMS unterstützt jetzt Amazon-Virtual-Private-Cloud (VPC)-Endpunkte als Quellen und Ziele. AWS DMS kann jetzt eine Verbindung zu jedem AWS-Service mit VPC-Endpunkten herstellen, wenn explizit definierte Routen zu den Services in ihrer AWS DMS-VPC definiert sind.

30. Juni 2022

|                                                                                               |                                                                                                                                                                                                 |               |
|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <a href="#"><u>Unterstützung von SQL-Server-Lesereplikaten als Quelle</u></a>                 | AWS DMS unterstützt jetzt SQL-Server-Lesereplikate als Quelle. Mit AWS DMS können Sie jetzt Live-Migrationen von SQL-Server-Lesereplikaten zu jedem von AWS DMS unterstützten Ziel durchführen. | 30. Juni 2022 |
| <a href="#"><u>Unterstützung von IBM-Db2-z/OS-Datenbanken als Quelle nur für Volllast</u></a> | AWS DMS unterstützt jetzt IBM-Db2-z/OS-Datenbanken als Quelle. Mit AWS DMS können Sie jetzt Live-Migrationen von Db2-Mainframes zu jedem von AWS DMS unterstützten Ziel durchführen.            | 30. Juni 2022 |
| <a href="#"><u>Unterstützung von EventBridge-DMS-Ereignissen</u></a>                          | AWS DMS unterstützt die Verwaltung von Ereignissen mithilfe von EventBridge für DMS-Ereignisse.                                                                                                 | 30. Juni 2022 |
| <a href="#"><u>Unterstützung von Babelfish als Ziel</u></a>                                   | AWS DMS unterstützt jetzt Babelfish als Ziel. Mit AWS DMS können Sie jetzt Live-Daten von jeder AWS DMS-unterstützten Quelle mit minimalen Ausfallzeiten zu Babelfish migrieren.                | 30. Juni 2022 |
| <a href="#"><u>Unterstützung von Aurora Serverless v2 als Ziel</u></a>                        | AWS DMS unterstützt jetzt Aurora Serverless v2 als Ziel. Mit AWS DMS können Sie jetzt Live-Migrationen zu Aurora Serverless v2 durchführen.                                                     | 30. Juni 2022 |

---

|                                                                                                       |                                                                                                                                                          |                   |
|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Erste Schritte – Tutorial</a>                                                             | Ein Update für das Tutorial „Erste Schritte“ für AWS DMS. Das Tutorial verwendet eine MySQL-Datenbank als Quelle und eine PostgreSQL-Datenbank als Ziel. | 20. Mai 2021      |
| <a href="#">Unterstützung für Amazon Neptune als Ziel</a>                                             | Unterstützung für Amazon Neptune als Ziel für die Datenmigration hinzugefügt.                                                                            | 1. Juni 2020      |
| <a href="#">Unterstützung für Apache Kafka als Ziel</a>                                               | Unterstützung für Apache Kafka als Ziel für die Datenmigration hinzugefügt.                                                                              | 20. März 2020     |
| <a href="#">Aktualisierte Sicherheitsinhalte</a>                                                      | Aktualisierte und standardisierte Sicherheitsinhalte als Antwort auf Kundenanfragen.                                                                     | 20. Dezember 2019 |
| <a href="#">Migrieren mit AWS Snowball Edge</a>                                                       | Unterstützung für die Verwendung von AWS Snowball Edge zum Migrieren großer Datenbanken hinzugefügt.                                                     | 24. Januar 2019   |
| <a href="#">Unterstützung für Amazon DocumentDB (mit MongoDB-Kompatibilität) als Ziel</a>             | Unterstützung für Amazon DocumentDB (mit MongoDB-Kompatibilität) als Ziel für Datenmigrationen hinzugefügt.                                              | 9. Januar 2019    |
| <a href="#">Unterstützung für Amazon OpenSearch Service und Amazon Kinesis Data Streams als Ziele</a> | Unterstützung für OpenSearch Services und Kinesis Data Streams als Ziele für Datenmigrationen hinzugefügt.                                               | 15. November 2018 |



[Unterstützung für nativen CDC-Start](#)

Native Startpunkte bei der Verwendung von Change Data Capture (CDC) werden nun unterstützt.

28. Juni 2018

[Unterstützung für Db2 LUW](#)

IBM Db2 LUW als Quelle der Datenmigration wird nun unterstützt.

26. April 2018

[Unterstützung für SQL Server als Ziel](#)

Es wurde Unterstützung für Amazon RDS for Microsoft SQL Server als Quelle hinzugefügt.

6. Februar 2018

# AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.