



用户指南

# AWS 数据库迁移服务



# AWS 数据库迁移服务: 用户指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

# Table of Contents

什么是 AWS Database Migration Service ? .....	1
AWS DMS 执行的迁移任务 .....	2
如何 AWS DMS 运作 .....	4
的高级视图 AWS DMS .....	4
组件 .....	5
源 .....	10
数据迁移的源 .....	10
DMS Fleet Advisor 的源 .....	13
DMS 架构转换的源 .....	14
DMS 同构数据迁移的源 .....	14
目标 .....	15
数据迁移的目标 .....	15
DMS Fleet Advisor 的目标 .....	17
DMS 架构转换的目标 .....	18
DMS 同构数据迁移的目标 .....	18
Amazon 资源名称 .....	18
使用其他 AWS 服务 .....	21
Support AWS CloudFormation .....	21
开始使用 .....	23
设置 .....	23
注册获取 AWS 账户 .....	23
创建具有管理访问权限的用户 .....	24
先决条件 .....	25
创建 VPC .....	26
创建 Amazon RDS 参数组 .....	27
创建源 Amazon RDS 数据库 .....	28
创建目标 Amazon RDS 数据库 .....	30
创建 Amazon EC2 客户端 .....	30
填充您的源数据库 .....	32
迁移架构 .....	33
复制 .....	35
步骤 1：创建复制实例 .....	35
步骤 2：指定源和目标端点 .....	37
步骤 3：创建任务并迁移数据 .....	39

步骤 4：测试复制 .....	41
步骤 5：清理 AWS DMS 资源 .....	43
其他资源 .....	44
发现要迁移的数据库 .....	45
支持 AWS 区域 .....	46
开始使用 .....	47
设置 .....	48
创建所需资源 .....	48
创建数据库用户 .....	57
数据收集器 .....	63
权限 .....	64
创建数据收集器 .....	64
安装数据收集器 .....	67
发现操作系统和数据库服务器 .....	70
管理受监控对象 .....	72
使用 SSL .....	74
收集数据 .....	75
故障排除 .....	79
清单 .....	82
使用数据库清单 .....	82
使用架构清单 .....	83
目标建议 .....	85
目标实例 .....	85
DMS Fleet Advisor 如何确定目标规格？ .....	86
生成目标建议 .....	86
建议详细信息 .....	88
导出目标建议 .....	89
迁移限制 .....	90
排查问题 .....	102
限制 .....	103
转换数据库架构 .....	105
支持的 AWS 区域 .....	106
功能 .....	106
限制 .....	107
开始使用 .....	108
先决条件 .....	109

步骤 1：创建实例配置文件 .....	113
步骤 2：配置数据提供程序 .....	114
步骤 3：创建迁移项目 .....	114
步骤 4：创建评估报告 .....	115
步骤 5：转换源代码 .....	116
步骤 6：应用转换后的代码 .....	116
步骤 7：清理和故障排除 .....	116
设置网络 .....	118
单个 VPC 配置 .....	118
多 VPC 配置 .....	118
使用 AWS Direct Connect 或 VPN .....	119
使用互联网连接 .....	119
使用没有互联网网关的环境 .....	119
创建源数据提供程序 .....	120
将 SQL Server 作为源 .....	120
将 Oracle 作为源 .....	122
将 Oracle 数据仓库用作源 .....	123
将 PostgreSQL 作为源 .....	125
将 MySQL 作为源 .....	125
创建目标数据提供程序 .....	126
使用 MySQL 作为目标 .....	126
将 PostgreSQL 作为目标 .....	128
将 Amazon Redshift 作为目标 .....	128
管理迁移项目 .....	129
指定迁移项目设置 .....	130
数据库迁移评测报告 .....	130
创建评测报告 .....	131
查看评测报告 .....	132
保存评测报告 .....	132
架构转换 .....	134
设置转换规则 .....	135
转换数据库架构 .....	137
指定架构转换设置 .....	140
刷新数据库架构 .....	144
保存和应用架构 .....	145
使用扩展包 .....	146

同构数据迁移 .....	147
支持的 AWS 区域 .....	148
功能 .....	148
限制 .....	149
概述 .....	149
设置 .....	150
创建 IAM 资源 .....	150
设置网络 .....	155
创建源数据提供程序 .....	158
将 MySQL 或 MariaDB 作为源 .....	158
将 PostgreSQL 作为源 .....	161
使用 MongoDB 或亚马逊 DocumentDB 作为来源 .....	164
创建目标数据提供程序 .....	166
使用 MySQL 或 MariaDB 作为目标 .....	167
将 PostgreSQL 作为目标 .....	168
将 Amazon DocumentDB 作为目标 .....	170
迁移数据 .....	170
创建数据迁移 .....	171
选择规则 .....	172
管理数据迁移 .....	175
监控数据迁移 .....	176
迁移状态 .....	178
从 MySQL 迁移数据 .....	178
从 PostgreSQL 迁移数据 .....	180
从 MongoDB 迁移数据 .....	182
故障排除 .....	183
创建数据迁移 .....	183
启动数据库迁移 .....	183
连接问题 .....	183
在 PostgreSQL 中以表的形式迁移视图 .....	184
处理迁移项目 .....	185
创建子网组 .....	185
创建实例配置文件 .....	186
创建实例提供程序 .....	187
创建迁移项目 .....	189
管理迁移项目 .....	190

最佳做法 .....	192
AWS Database Migration Service 的迁移规划 .....	192
架构转换 .....	193
查看 AWS DMS 文档 .....	194
运行概念验证 .....	194
改进性能 .....	194
使用您自己的本地名称服务器 .....	198
将 Amazon Route 53 Resolver 与 AWS DMS 配合使用 .....	199
迁移大型二进制对象 (LOB) .....	200
使用受限 LOB 模式 .....	200
提高了 LOB 性能 .....	201
使用行筛选提高迁移大型表时的性能 .....	204
持续复制 .....	204
减少源数据库上的负载 .....	205
减少目标数据库的瓶颈 .....	205
使用数据验证 .....	205
监控指标 .....	206
事件 .....	206
使用任务日志 .....	207
使用 Time Travel 对复制进行故障排除 .....	207
为 Oracle 目标更改用户和架构 .....	207
更改 Oracle 目标的表和索引表空间 .....	208
升级复制实例 .....	209
了解迁移成本 .....	209
使用无 AWS DMS 服务器 .....	210
DMS Serverless 组件 .....	211
支持的引擎版本 .....	213
创建无服务器复制 .....	214
修改 AWS DMS 无服务器复制 .....	216
Compute Config .....	219
了解无服务器中的 AWS DMS 自动扩展 .....	220
监控 AWS DMS 无服务器复制 .....	220
扩展满载吞吐量 .....	224
Serverless 限制 .....	225
使用复制实例 .....	227
选择复制实例类型 .....	231

决定使用哪个实例类 .....	235
无限模式下的可突增实例 .....	236
调整复制实例的大小 .....	237
需要考虑的因素 .....	237
常见问题 .....	238
最佳实践 .....	238
复制引擎版本 .....	239
使用控制台升级引擎版本 .....	239
使用升级引擎版本 AWS CLI .....	240
公有和私有复制实例 .....	241
IP 地址和网络类型 .....	241
为复制实例设置网络 .....	243
数据库迁移的网络配置 .....	243
创建复制子组 .....	250
使用 DNS 解析域端点 .....	251
设置加密密钥 .....	252
创建复制实例 .....	252
修改复制实例 .....	256
重启复制实例 .....	259
删除复制实例 .....	262
DMS 维护时段 .....	263
维护对现有迁移任务的影响 .....	263
使用维护时段设置 .....	264
端点 .....	266
创建源和目标终端节点 .....	266
数据迁移的源 .....	270
将 Oracle 作为源 .....	271
将 SQL Server 作为源 .....	329
将 Azure SQL 数据库作为源 .....	352
使用 Azure SQL 托管实例作为来源 .....	353
将 Azure Database for PostgreSQL 作为来源 .....	353
将 Azure Database for MySQL 作为来源 .....	354
使用 OCI MySQL Heatwave 作为来源 .....	355
将 Google Cloud for MySQL 作为来源 .....	355
将 Google Cloud for PostgreSQL 作为来源 .....	356
将 PostgreSQL 作为源 .....	357



将 MySQL 作为源 .....	389
将 SAP ASE 作为源 .....	401
将 MongoDB 作为源 .....	408
将 Amazon DocumentDB 作为源 .....	424
将 Amazon S3 作为源 .....	440
将 IBM Db2 LUW 作为源 .....	452
将 IBM Db2 for z/OS 用作源 .....	459
数据迁移的目标 .....	498
将 Oracle 作为目标 .....	499
将 SQL Server 作为目标 .....	509
将 PostgreSQL 作为目标 .....	513
使用 MySQL 作为目标 .....	524
将 Amazon Redshift 作为目标 .....	531
将 SAP ASE 作为目标 .....	554
将 Amazon S3 作为目标 .....	557
将 Amazon DynamoDB 作为目标 .....	599
使用 Amazon Kinesis Data Streams 作为目标 .....	620
使用 Apache Kafka 作为目标 .....	637
将 OpenSearch 作为目标 .....	661
将 Amazon DocumentDB 作为目标 .....	666
将 Amazon Neptune 作为目标 .....	678
将 Redis 作为目标 .....	693
使用 Babelfish 作为目标 .....	699
将 Amazon Timestream 作为目标 .....	706
将 Db2 作为目标 .....	716
用于数据迁移的 VPC 端点 .....	717
迁移到 AWS DMS 版本 3.4.7 及更高版本时，谁会受到影响？ .....	717
迁移到 AWS DMS 版本 3.4.7 及更高版本时，谁不会受到影响？ .....	717
准备迁移到 AWS DMS 3.4.7 及更高版本 .....	718
支持的 DDL 语句 .....	719
任务 .....	720
创建任务 .....	723
Task settings ( 任务设置 ) .....	729
设置 LOB 支持 .....	775
创建多个任务 .....	777
连续复制任务 .....	777

从 CDC 开始点开始的复制 .....	778
执行双向复制 .....	783
修改任务 .....	786
移动任务 .....	787
在任务期间重新加载表 .....	787
AWS Management Console .....	788
表映射 .....	789
通过控制台指定表选择和转换规则 .....	790
使用 JSON 指定表选择和转换规则 .....	793
选择规则和操作 .....	794
表映射中的通配符 .....	800
转换规则和操作 .....	801
使用转换规则表达式定义列内容 .....	820
表和集合设置规则和操作 .....	833
使用源筛选器 .....	860
应用筛选器 .....	861
按时间和日期筛选 .....	867
启用和使用迁移前评估 .....	868
先决条件 .....	869
指定、启动和查看评估运行 .....	871
个人评估 .....	875
启动和查看数据类型评估 .....	903
对评估运行进行故障排除 .....	906
指定补充数据 .....	908
监控任务 .....	909
任务状态 .....	910
任务期间的表状态 .....	913
使用 Amazon CloudWatch 监控复制任务 .....	914
AWS Database Migration Service 指标 .....	915
复制实例指标 .....	918
复制任务指标 .....	920
查看和管理 AWS DMS 日志 .....	923
使用 AWS DMS 记录 AWS CloudTrail API 调用 .....	924
CloudTrail 中的 AWS DMS 信息 .....	924
了解 AWS DMS 日志文件条目 .....	925
上下文日志记录 .....	928

对象类型 .....	929
日志记录示例 .....	930
限制 .....	931
使用 EventBridge 事件 .....	933
将 Amazon EventBridge 事件规则用于 AWS DMS .....	934
AWS DMS 事件类别和事件消息 .....	935
ReplicationInstance 事件消息 .....	935
ReplicationTask 事件消息 .....	937
复制事件消息 .....	939
处理 Amazon SNS 事件 .....	941
将事件订阅移至 Amazon EventBridge .....	941
使用 Amazon SNS 事件和通知 .....	942
AWS DMS 事件类别和 SNS 通知的事件消息 .....	943
使用 SNS 订阅 AWS DMS 事件通知 .....	946
使用 AWS Management Console .....	946
验证您的 SNS 主题的访问策略 .....	949
数据验证 .....	951
复制任务统计数据 .....	952
使用 Amazon CloudWatch 的复制任务统计数据 .....	954
在任务期间重新验证表 .....	955
AWS Management Console .....	955
使用 JSON 编辑器修改验证规则 .....	956
仅验证任务 .....	956
完全加载仅验证 .....	957
CDC 仅验证 .....	957
仅验证的使用案例 .....	957
故障排除 .....	958
Redshift 验证性能 .....	959
限制 .....	960
S3 验证 .....	961
先决条件 .....	962
权限 .....	962
限制 .....	964
仅验证任务 .....	965
为资源添加标签 .....	966
API .....	967

安全性 .....	969
数据保护 .....	971
数据加密 .....	971
互连网络流量隐私保护 .....	972
DMS Fleet Advisor 中的数据保护 .....	972
Identity and Access Management .....	973
受众 .....	973
使用身份进行身份验证 .....	974
使用策略管理访问 .....	976
如何 AWS Database Migration Service 与 IAM 配合使用 .....	978
基于身份的策略示例 .....	984
基于资源的策略示例 .....	991
使用密钥访问资源 .....	996
使用服务相关角色 .....	1004
故障排除 .....	1010
所需的 IAM 权限 .....	1013
适用于 CLI 和 API 的 IAM 角色 .....	1017
防止跨服务混淆座席 .....	1023
AWS 托管策略 .....	1026
合规性验证 .....	1033
故障恢复能力 .....	1035
基础设施安全性 .....	1036
精细访问控制 .....	1039
使用资源名称控制访问 .....	1039
使用标签控制访问 .....	1042
设置加密密钥 .....	1049
网络安全 .....	1051
使用 SSL .....	1053
将 SSL 与 AWS DMS 配合使用的限制 .....	1054
管理证书 .....	1055
为 MySQL 兼容、PostgreSQL 或 SQL Server 终端节点启用 SSL .....	1055
更改数据库密码 .....	1058
限制 .....	1059
AWS Database Migration Service 资源配额 .....	1059
了解 API 请求节流 .....	1060
故障排除和诊断支持 .....	1062

迁移任务运行缓慢 .....	1063
任务状态栏没有进展 .....	1063
任务已完成但未迁移任何内容 .....	1064
缺少外键和辅助索引 .....	1064
AWS DMS不创建 CloudWatch 日志 .....	1064
连接 Amazon RDS 时出现问题 .....	1065
错误消息：线程连接字符串不正确：线程值 0 不正确 .....	1065
出现网络问题 .....	1065
CDC 在完全加载后粘滞 .....	1066
重新启动任务时出现主键冲突错误 .....	1066
架构初始加载失败 .....	1066
任务失败，出现未知错误 .....	1066
任务重新启动操作从头开始加载表 .....	1067
每个任务的表数量导致问题 .....	1067
在 LOB 列上创建主键时，任务失败 .....	1067
不带主键的目标表上出现重复的记录 .....	1067
预留 IP 范围内的源端点失败 .....	1067
Amazon Athena 查询中的时间戳是乱码 .....	1068
排除 Oracle 中的问题 .....	1068
从视图中提取数据 .....	1068
从 Oracle 12c 迁移 LOB .....	1069
在甲骨文 LogMiner和二进制阅读器之间切换 .....	1069
错误：Oracle CDC 已停止 122301 Oracle CDC 已超出最大重试计数器。 .....	1070
自动将补充日志记录添加到 Oracle 源端点 .....	1070
未捕获 LOB 更改 .....	1070
错误：ORA-12899: 值对于列 <i>column-name</i> 过大 .....	1071
未正确解释 NUMBER 数据类型 .....	1071
完全加载期间缺少记录 .....	1071
表错误 .....	1071
错误：无法检索 Oracle 存档的重做日志目标 ID .....	1072
评估 Oracle 重做或存档日志的读取性能 .....	1072
排除 MySQL 中的问题 .....	1074
由于已禁用二进制日志记录，无法为 Amazon RDS 数据库实例终端节点完成 CDC 任务 ....	1074
与目标 MySQL 实例的连接在任务期间中断 .....	1075
将自动提交添加到与 MySQL 兼容的终端节点 .....	1075
在与 MySQL 兼容的目标终端节点上禁用外键 .....	1076

字符被替换为问号 .....	1076
“错误事件”日志条目 .....	1076
使用 MySQL 5.5 的更改数据捕获 .....	1077
增加 Amazon RDS 数据库实例的二进制日志保留期 .....	1077
日志消息：源数据库中的一些更改在应用于目标数据库时不起作用。 .....	1077
错误：标识符太长 .....	1077
错误：不受支持的字符集导致字段数据转换失败 .....	1077
错误：代码页 1252 到 UTF8 [120112] 字段数据转换失败 .....	1078
索引、外键或级联更新内容或删除内容未迁移 .....	1078
排除 PostgreSQL 中的问题 .....	1080
JSON 数据类型被截断 .....	1081
用户定义的数据类型的列未正确迁移 .....	1081
错误：未选择要在其中执行创建操作的架构 .....	1081
未使用 CDC 复制针对表的删除内容和更新内容 .....	1082
未传播截断语句 .....	1082
阻止 PostgreSQL 捕获 DDL .....	1082
选择在其中创建用于捕获 DDL 的数据库对象的架构 .....	1082
迁移到 PostgreSQL 后缺少 Oracle 表 .....	1082
ReplicationSlotDiskUsage 在长时间交易（例如 ETL 工作负载）期间，增加并且 restart_lsn 停止向前移动 .....	1083
使用视图作为源的任务未复制行 .....	1083
排除 Microsoft SQL Server 中的问题 .....	1083
捕获 SQL Server 数据库的更改时出错 .....	1084
缺少身份列 .....	1084
错误：SQL Server 不支持发布 .....	1084
更改未显示在目标中 .....	1084
跨分区映射的不一致表 .....	1085
排除 Amazon Redshift 中的问题 .....	1085
加载到不同 AWS 区域内的 Amazon Redshift 集群中 .....	1086
错误：关系“attrep_apply_exceptions”已存在 .....	1086
其名称以“awsdms_changes”开头的表出错 .....	1086
查看集群中名称类似于 dms.awsdms_changes000000000XXXX 的表 .....	1086
使用 Amazon Redshift 所需的权限 .....	1086
排除 Amazon Aurora MySQL 中的问题 .....	1086
错误：CHARACTER SET UTF8 字段由“,”终止，包含在“”中，行由“\n”终止 .....	1087
排除 SAP ASE 的问题 .....	1087

错误：当源具有包含 NULL 值的复合唯一索引时，LOB 列的值为 NULL .....	1087
排除 IBM Db2 中的问题 .....	1087
错误：不支持从时间戳恢复任务 .....	1088
排除延迟问题 .....	1088
CDC 延迟的类型 .....	1088
CDC 延迟的常见原因 .....	1089
排除延迟问题 .....	1092
使用诊断支持脚本 .....	1104
Oracle 支持脚本 .....	1105
SQL Server 支持脚本 .....	1108
MySQL 兼容数据库的支持脚本 .....	1133
PostgreSQL 支持脚本 .....	1135
使用诊断支持 AMI .....	1137
启动新的 AWS DMS 诊断 Amazon EC2 实例 .....	1138
创建 IAM 角色 .....	1138
运行诊断测试 .....	1139
后续步骤 .....	1143
按区域列出的 AMI ID .....	1143
参考 .....	1145
AWS DMS 数据类型 .....	1145
发布说明 .....	1148
AWS DMS 3.5.3 发行说明 .....	1149
AWS DMS 3.5.2 发行说明 .....	1151
AWS DMS 3.5.1 发行说明 .....	1153
AWS DMS 3.5.0 测试版发行说明 .....	1162
AWS DMS 3.4.7 发行说明 .....	1166
AWS DMS 3.4.6 发行说明 .....	1173
AWS DMS 3.4.5 发行说明 .....	1178
AWS DMS 3.4.4 发行说明 .....	1180
AWS DMS 3.4.3 发行说明 .....	1182
AWS DMS 3.4.2 发行说明 .....	1184
AWS DMS 3.4.1 发行说明 .....	1185
AWS DMS 3.4.0 版本说明 .....	1186
AWS DMS 3.3.4 发行说明 .....	1187
AWS DMS 3.3.3 发行说明 .....	1188
文档历史记录 .....	1189

---

AWS 术语表 .....	1193
.....	mcxciv



# 什么是 AWS Database Migration Service ?

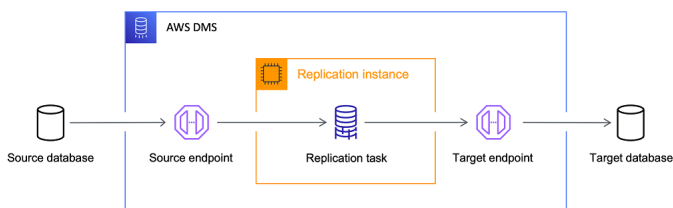
AWS Database Migration Service (AWS DMS) 是一项云服务，可轻松迁移关系数据库、数据仓库、NoSQL 数据库及其他类型的数据存储。您可以使用 AWS DMS 将数据迁移到 AWS Cloud 中，或者结合使用云和本地设置进行迁移。

使用 AWS DMS，您可以搜索源数据存储、转换源架构和迁移数据。

- 要搜索您的源数据基础设施，您可以使用 DMS Fleet Advisor。此服务从您的本地数据库和分析服务器收集数据，并构建可迁移到 AWS Cloud 的服务器、数据库和架构的清单。
- 要迁移到其他数据库引擎，您可以使用 DMS 架构转换。此服务会自动评测您的源架构，并将其转换到新的目标引擎。或者，您可以将 AWS Schema Conversion Tool (AWS SCT) 下载到本地 PC 以转换源架构。
- 转换源架构并将转换后的代码应用于目标数据库之后，您可以使用 AWS DMS 迁移数据。您可以执行一次性迁移，或者复制持续更改以保持源和目标同步。由于 AWS DMS 是 AWS Cloud 的一部分，您将获得 AWS 服务提供的成本效益、上市速度、安全性与灵活性。

在基本级别上，AWS DMS 是 AWS Cloud 中运行复制软件的服务器。您创建源和目标连接以告知 AWS DMS 在哪里提取数据，以及将数据加载到哪里。然后，您计划在此服务器上运行的任务以迁移数据。AWS DMS 会创建表和关联的主键（如果它们在目标上不存在）。如果您愿意，可以自行创建目标表。或者，您也可以使用 AWS Schema Conversion Tool (AWS SCT) 创建部分或全部目标表、索引、视图、触发器等。

下列示意图说明了 AWS DMS 复制过程。



## 参考

- 支持 AWS DMS 的 AWS 区域 – 有关哪些 AWS 区域支持 AWS DMS 的信息，请参阅 [使用 AWS DMS 复制实例](#)。
- 数据库迁移成本 – 有关数据库迁移成本的信息，请参阅 [AWS Database Migration Service 定价页面](#)。

- AWS DMS 功能和优点 – 有关 AWS DMS 的功能和优点的信息，请参阅 [AWS Database Migration Service 功能](#)。
- 可用数据库选项 – 要详细了解 Amazon Web Services 上提供的各种数据库选项，请参阅 [为组织选择合适的数据库](#)。

## AWS DMS 执行的迁移任务

AWS DMS 负责迁移项目中涉及的许多困难或繁琐的任务。

- 在传统解决方案中，您需要执行容量分析、采购硬件和软件、安装和管理系统以及测试和调试安装。AWS DMS 会自动管理迁移所需的所有硬件和软件的部署、管理和监控。您的迁移可在启动 AWS DMS 配置过程的几分钟内启动并运行。
- 利用 AWS DMS，您可以根据需要向上扩展（或向下扩展）迁移以匹配实际工作负载。例如，如果您确定需要额外的存储空间，则可轻松增加分配的存储空间并重新开始您的迁移，通常在几分钟内完成。
- AWS DMS 采用即付即用模式。您只需为所使用的 AWS DMS 资源付费，与传统的许可模式相对，无需支付前期购买成本和日常维护费用。
- AWS DMS 自动管理迁移服务器的所有支持基础设施，包括硬件和软件、软件修补和错误报告。
- AWS DMS 提供自动故障转移。如果您的主复制服务器因任何原因发生故障，则备份复制服务器可接管运行，而对服务造成的中断很少或没有。
- AWS DMS Fleet Advisor 会自动清点您的数据基础设施。它会创建报告，帮助您确定迁移候选项并规划迁移。
- AWS DMS 架构转换会针对您的源数据提供商自动评测迁移的复杂性。它还将数据库架构和代码对象转换为与目标数据库兼容的格式，然后应用转换后的代码。
- AWS DMS 可以帮助您切换到一个现代化、可能比您现在运行的引擎更具成本效益的数据库引擎。例如，AWS DMS 可以帮助您利用由 Amazon Relational Database Service (Amazon RDS) 或 Amazon Aurora 提供的托管式数据库服务。或者，它可以帮助您迁移到由 Amazon Redshift 提供的托管数据仓库服务、NoSQL 平台（例如 Amazon DynamoDB）或低成本存储平台（例如 Amazon Simple Storage Service (Amazon S3)）。相反，如果要迁离旧的基础设施但继续使用相同的数据库引擎，AWS DMS 也支持该过程。
- AWS DMS 支持将几乎所有当今最流行的 DBMS 引擎作为源端点。有关更多信息，请参阅 [数据迁移的源](#)。
- AWS DMS 全面涵盖了可用的目标引擎。有关更多信息，请参阅 [数据迁移的目标](#)。

- 您可以从任何支持的数据源迁移到任何支持的数据目标。AWS DMS 完全支持受支持的引擎之间的异构数据迁移。
- AWS DMS 确保数据迁移是安全的。静态数据通过 AWS Key Management Service (AWS KMS) 加密进行加密。迁移过程中，您可以在传输中的数据从源传输至目标时使用安全套接字层 (SSL) 对其进行加密。

# AWS 数据库迁移服务的工作原理

AWS Database Migration Service (AWS DMS) 是一项 Web 服务，可用于将数据从源数据存储迁移到目标数据存储。这两个数据存储称为终端节点。您可在使用相同数据库引擎的源和目标终端节点之间迁移，例如从 Oracle 数据库到 Oracle 数据库。您也可在使用不同数据库引擎的源和目标终端节点之间迁移，例如从 Oracle 数据库到 PostgreSQL 数据库。唯一的使用要求 AWS DMS 是您的一个终端节点必须位于 AWS 服务上。您不能使用从本地数据库迁移 AWS DMS 到另一个本地数据库。

有关数据库迁移成本的信息，请参阅 [AWS Database Migration Service 定价页面](#)。

使用以下主题来更好地理解 AWS DMS。

## 主题

- [的高级视图 AWS DMS](#)
- [的组成部分 AWS DMS](#)
- [的来源 AWS DMS](#)
- [的目标 AWS DMS](#)
- [为其构造亚马逊资源名称 \(ARN\) AWS DMS](#)
- [与其他 AWS 服务 AWS DMS 一起使用](#)

## 的高级视图 AWS DMS

要执行数据库迁移，请 AWS DMS 连接到源数据存储，读取源数据并格式化数据以供目标数据存储使用。然后，它会将数据加载到目标数据存储中。此处理大部分在内存中进行，不过大型事务可能需要部分缓冲到磁盘。缓存事务和日志文件也会写入磁盘。

简而言之，在使用 AWS DMS 时，请执行以下操作：

- 发现您的网络环境中适合迁移的数据库。
- 自动将源数据库架构和大部分数据库代码对象转换为与目标数据库兼容的格式。
- 创建复制服务器。
- 创建源和目标终端节点，它们具有有关您的数据存储的连接信息。
- 创建一个或多个迁移任务以在源和目标数据存储之间迁移数据。

任务可能包括三个主要阶段：

- 迁移现有数据 ( 完全加载 )
- 应用缓存的更改
- 持续复制 ( 更改数据捕获 )

在满负荷迁移期间，将源中的现有数据移动到目标，会将数据从源数据存储中的表 AWS DMS 加载到目标数据存储上的表。在完全加载进行期间，对所加载表进行的更改将缓存到复制服务器上；这些是缓存的更改。值得注意的是，在给定表的满负荷开始之前，它 AWS DMS 不会捕获该表的更改。换句话说，对于每个单独的表，开始捕获更改的时间点是不同的。

当给定表的满载完成后，AWS DMS 立即开始对该表应用缓存的更改。加载表并应用缓存的更改后，将 AWS DMS 开始收集更改作为正在进行的复制阶段的交易。如果事务中的表尚未完全加载，则更改将存储在复制实例的本地。AWS DMS 将所有缓存的更改应用于所有表后，表在事务上保持一致。此时，AWS DMS 进入正在进行的复制阶段，将更改作为事务进行应用。

持续复制阶段开始之后，积压的事务通常会导致源数据库与目标数据库之间的一些滞后。在处理完这些积压事务之后，迁移最终进入稳态。此时，您可以关闭应用程序，允许任何剩余的事务应用到目标，然后启动您的应用程序，现在指向目标数据库。

AWS DMS 创建执行数据迁移所需的目标架构对象。您可以使用 AWS DMS 采用极简主义的方法，仅创建高效迁移数据所需的对象。使用这种方法 AWS DMS 可以创建表、主键，在某些情况下还会创建唯一索引，但它不会创建任何其他不需要从源中高效迁移数据的对象。

或者，您也可以在其中使用 DMS 架构转换将源数据库架构和大多数数据库代码对象自动转换为与目标数据库兼容的格式。AWS DMS 此转换包括表、视图、存储过程、函数、数据类型、同义词等。任何 DMS 架构转换无法自动转换的对象都会被明确标记。要完成迁移，您可以手动转换这些对象。

## 的组成部分 AWS DMS

本节介绍的内部组件 AWS DMS 以及它们如何协同工作以完成数据迁移。了解 AWS DMS 的基础组件可以帮助您更高效地迁移数据，并可在进行故障排除或调查问题时提供更好的洞察力。

AWS DMS 迁移由五个部分组成：发现要迁移的数据库、自动架构转换、复制实例、源和目标端点以及复制任务。您可以通过在中创建必要的复制实例、终端节点和任务来创建 AWS DMS 迁移 AWS 区域。

### 数据库发现

DMS Fleet Advisor 从多个数据库环境收集数据，以深入了解您的数据基础设施。DMS Fleet Advisor 从一个或多个中心位置，从您的本地数据库和分析服务器收集数据，无需将其安装在

每台计算机上。目前，DMS Fleet Advisor 支持 Microsoft SQL Server、MySQL、Oracle 和 PostgreSQL 数据库服务器。

根据从您的网络中发现的数据，DMS Fleet Advisor 会构建一份清单，供您查看以确定要监控哪些数据库服务器和对象。收集有关这些服务器、数据库和架构的详细信息后，您可以分析目标数据库迁移的可行性。

## 架构和代码迁移

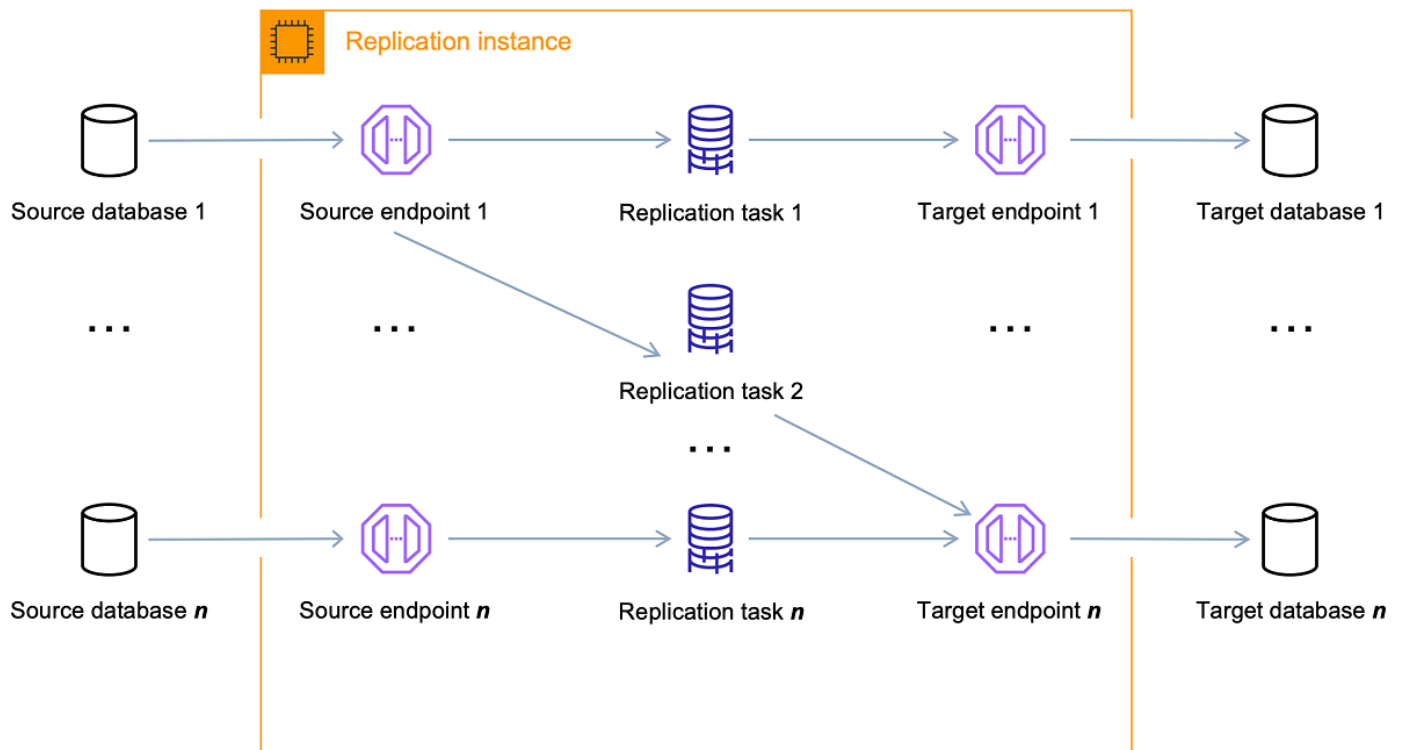
中的 DMS 架构转换 AWS DMS 使不同类型的数据库之间的数据库迁移更具可预测性。您可以使用 DMS 架构转换来评估源数据提供程序的迁移复杂性，然后使用它来转换数据库架构和代码对象。然后，您可以将转换后的代码应用于目标数据库。

简而言之，DMS 架构转换由以下三个组件运行：实例配置文件、数据提供程序和迁移项目。实例配置文件指定网络和安全设置。数据提供程序存储数据库连接凭证。迁移项目包含数据提供程序、实例配置文件和迁移规则。AWS DMS 使用数据提供程序和实例配置文件来设计转换数据库架构和代码对象的流程。

## 复制实例

简而言之，AWS DMS 复制实例就是托管一个或多个复制任务的托管亚马逊弹性计算云 (Amazon EC2) 实例。

下图显示了运行多个关联的复制任务的示例复制实例。



单个复制实例可以托管一个或多个复制任务，具体取决于您的迁移特征和复制服务器的容量。AWS DMS 提供了各种复制实例，因此您可以根据自己的用例选择最佳配置。有关各类复制实例的更多信息，请参阅[为您的迁移选择正确的 AWS DMS 复制实例](#)。

AWS DMS 在 Amazon EC2 实例上创建复制实例。一些较小的实例类足够用于测试服务或小型迁移。如果您的迁移涉及大量表，或者您打算同时运行多个复制任务，则应考虑使用较大的实例之一。建议使用这种方法，因为 AWS DMS 可能会占用大量内存和 CPU。

根据所选的 Amazon EC2 实例类，您的复制实例附带 50 GB 或 100 GB 的数据存储。对于大多数客户，此数量通常已足够使用。但是，如果您的迁移涉及大型事务或大量数据更改，则可能需要增加基本存储分配。更改数据捕获 (CDC) 可能会导致数据被写入磁盘，具体取决于目标写入更改的速度。由于日志文件也会写入磁盘，因此提高日志记录的严重性级别也会导致更高的存储消耗。

AWS DMS 可以使用多可用区部署提供高可用性和故障转移支持。在多可用区部署中，在不同的可用区域中 AWS DMS 自动配置和维护复制实例的备用副本。主复制实例将同步复制到备用副本。如果主复制实例发生故障或没有响应，备用副本将以最少中断恢复任何正在运行的任务。由于主复制实例会不断将其状态复制到备用副本中，因此多可用区部署会产生一些性能开销。

有关 AWS DMS 复制实例的更多详细信息，请参阅[使用 AWS DMS 复制实例](#)。

您可以使用 AWS DMS Serverless 自动 AWS DMS 配置复制，而不必创建和管理复制实例。有关更多信息，请参阅[使用无 AWS DMS 服务器](#)。

## 终端节点

AWS DMS 使用端点访问您的源或目标数据存储。根据您的数据存储，特定连接信息会有所不同，但通常，您在创建终端节点时会提供以下信息：

- 端点类型 – 源或目标。
- 引擎类型 – 数据库引擎的类型，如 Oracle 或 PostgreSQL。
- 服务器名称- AWS DMS 可以访问的服务器名称或 IP 地址。
- 端口 – 用于数据库服务器连接的端口号。
- 加密 – 安全套接字层 (SSL) 模式 (如果使用 SSL 加密连接)。
- 凭证 – 具有所需访问权限的账户的用户名和密码。

使用 AWS DMS 控制台创建终端节点时，控制台要求您测试终端节点连接。在 AWS DMS 任务中使用终端节点之前，测试必须成功。与连接信息相似，不同引擎类型的特定测试标准不同。一般而言，AWS DMS 会验证给定服务器名称和端口处是否存在数据库以及提供的凭证是否可用于连接数据库，并具有执行迁移所需的必要权限。如果连接测试成功，则会 AWS DMS 下载并存储架构信息，以便稍后在任务配置期间使用。例如，架构信息可能包括表定义、主键定义和唯一键定义。

多个复制任务可以使用单个终端节点。例如，您可能有两个在逻辑上不同的应用程序托管在要单独迁移的同一源数据库中。此种情况下，您将创建两个复制任务，每组应用程序表对应一个复制任务。您可以在两个任务中使用相同的 AWS DMS 终端节点。

您可以使用端点设置自定义端点的行为。端点设置可以控制多种行为，如日志记录详细信息、文件大小以及其他参数。每种数据存储引擎类型都有不同的端点设置可用。您可以在各个数据存储的源或目标部分中，查找该数据存储的特定端点设置。有关支持的源和目标数据存储的列表，请参阅[的来源 AWS DMS](#)和[的目标 AWS DMS](#)。

有关 AWS DMS 终端节点的更多详细信息，请参阅[使用 AWS DMS 端点](#)。

## 复制任务

您可以使用 AWS DMS 复制任务将一组数据从源终端节点移动到目标终端节点。创建复制任务是在开始迁移之前需要执行的最后一步。

当您创建复制任务时，需要指定以下任务设置：

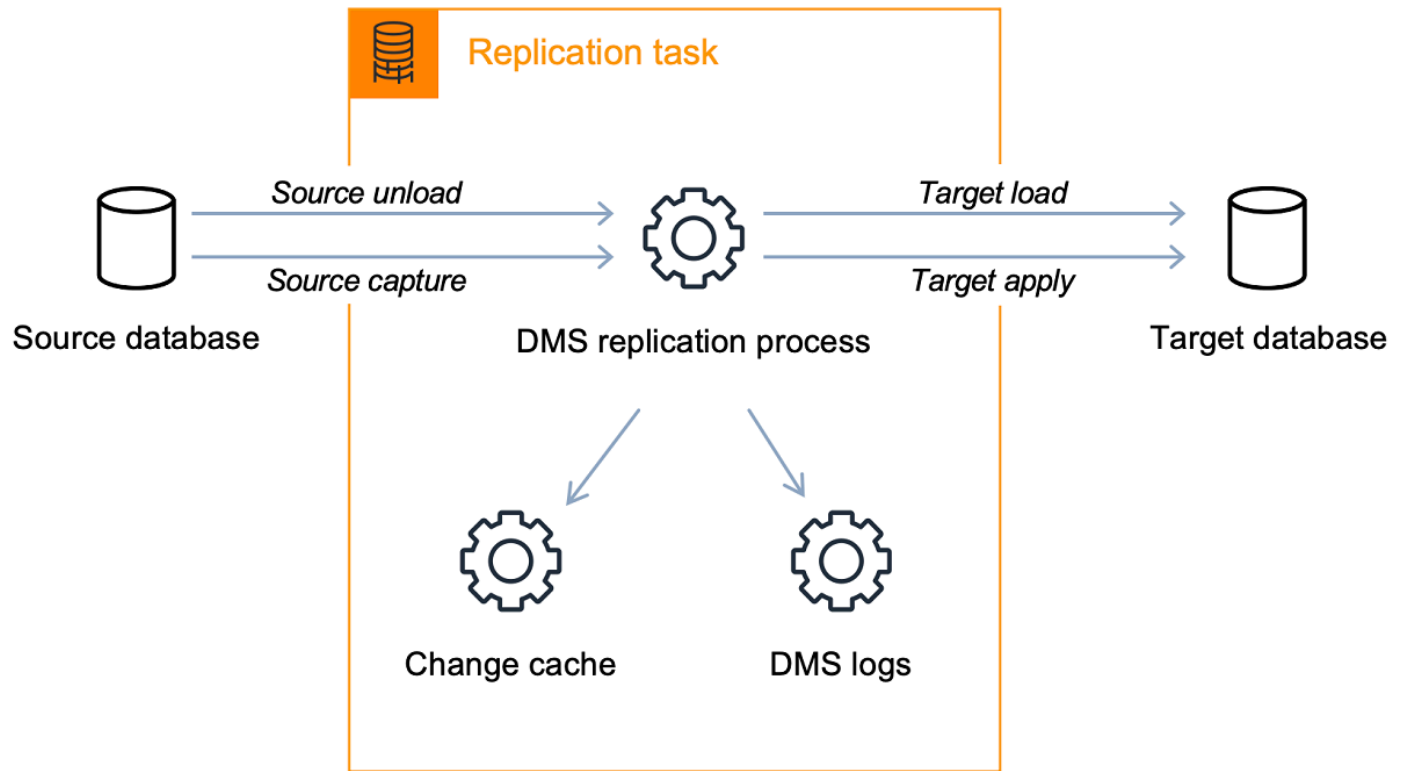
- 复制实例 – 用于托管和运行任务的实例
- 源终端节点
- 目标终端节点
- 迁移类型选项，如下所列。有关迁移类型选项的完整说明，请参阅 [创建任务](#)。
  - 完全加载（迁移现有数据）– 如果您可以承受足够长时间的中断来复制现有数据，则适合选择此选项。该选项仅将数据从您的源数据库迁移到目标数据库，在需要时创建表。
  - 完全加载 + CDC（迁移现有数据和复制持续更改）– 此选项执行完整数据加载，同时捕获源上的更改。在完全加载完成后，将捕获的更改应用到目标。最终更改应用达到稳态。此时您可以关闭应用程序，让剩余的更改传播到目标，然后重新启动应用程序并指向目标。
  - 仅 CDC（仅复制数据更改）– 在某些情况下，使用 AWS DMS 之外的方法复制现有数据可能会更高效。例如，在同构迁移中，使用本机导出和导入工具可能会在加载批量数据时更有效。在这种情况下，您可以使用复制更改 AWS DMS，从开始批量加载开始，使源数据库和目标数据库保持同步。
- 目标表准备模式选项，如下所列。有关目标表模式的完整说明，请参阅 [创建任务](#)。
  - 什么都不做 — AWS DMS 假设目标表是在目标上预先创建的。
  - 在目标上删除表- AWS DMS 删除并重新创建目标表。
  - 截断 – 如果您在目标上创建表，AWS 将在开始迁移之前截断这些表。如果不存在任何表并且您选择了此选项，则 AWS DMS 会创建所有缺失的表。
- LOB 模式选项，如下所列。有关 LOB 模式的完整说明，请参阅 [在任务中为源数据库设置 LOB 支持 AWS DMS](#)。



- 不包括 LOB 列 – 从迁移操作中排除 LOB 列。
- 完整 LOB 模式 — 无论大小如何，都迁移完整的 LOB。AWS DMS 按最大 LOB 大小参数控制的区块分段迁移 LOB。此模式比受限 LOB 模式的速度要慢。
- 受限 LOB 模式 – 按最大 LOB 大小参数指定的值截断 LOB。此模式比使用完整 LOB 模式的速度要快。
- 表映射 – 指示要迁移的表及其迁移方式。有关更多信息，请参阅 [使用表映射指定任务设置](#)。
- 数据转换，如下所列。有关数据转换的更多信息，请参阅 [使用 JSON 指定表选择和转换规则](#)。
  - 更改架构、表和列名称。
  - 更改表空间名称（针对 Oracle 目标终端节点）。
  - 在目标上定义主键和唯一索引。
- 数据验证
- 亚马逊 CloudWatch 日志

您可以使用任务将数据从源终端节点迁移到目标终端节点，任务处理在复制实例上完成。您可以指定要迁移的表和架构以及任何特殊处理，例如，日志记录要求、控制表数据和错误处理。

从概念上讲，AWS DMS 复制任务执行两个不同的功能，如下图所示。



完全加载过程非常简单明了，易于理解。从源中以批量提取方式提取数据，并将其直接加载到目标中。您可以在 AWS DMS 控制台的“高级设置”下指定要并行提取和加载的表的数量。

有关 AWS DMS 任务的更多信息，请参阅[使用 AWS DMS 任务](#)。

### 持续复制，或更改数据捕获 (CDC)

在将数据迁移到目标时，您还可以使用 AWS DMS 任务来捕获源数据存储的持续更改。从源端点复制正在进行的更改时 AWS DMS 使用的更改捕获过程使用数据库引擎的本机 API 来收集对数据库日志的更改。

在 CDC 过程中，复制任务旨在使用内存中缓冲区保存传输中的数据，从而从源到目标流式处理更改。如果内存中缓冲区因任何原因而耗尽，负责任务会将待处理更改溢出到磁盘上的更改缓存。例如，如果 AWS DMS 从源捕获更改的速度快于在目标系统上应用更改的速度，则可能会发生这种情况。在这种情况下，您会看到任务的目标延迟 超出任务的源延迟。

您可以通过在 AWS DMS 控制台上导航到您的任务并打开“任务监控”选项卡来检查这一点。CDC LatencyTarget 和 CDC LatencySource 图表显示在页面底部。如果您有显示目标延迟的任务，则可能需要对目标终端节点进行一些优化，以提高应用程序速率。

复制任务还会为任务日志使用存储空间，如上所述。随复制实例预配置的磁盘空间通常足以满足日志记录和溢出更改需要。如果您需要更多磁盘空间，例如，在使用详细调试调查迁移问题时，您可以修改复制实例以分配更多空间。

## 的来源 AWS DMS

您可以在不同的 AWS DMS 功能中使用不同的源数据存储。以下各节包含每项 AWS DMS 功能支持的源数据存储列表。

### 主题

- [数据迁移的源端点](#)
- [DMS Fleet Advisor 的源数据库](#)
- [DMS 架构转换的源数据提供程序](#)
- [DMS 同构数据迁移的源数据提供程序](#)

## 数据迁移的源端点

在使用 AWS DMS 进行数据迁移时，可以将以下数据存储作为源终端节点。

## 本地和 EC2 实例数据库

- Oracle 10.2 及更高版本 ( 对于版本 10.x ) 、 11g 直至 12.2、18c 以及 19c ( Enterprise、Standard、Standard One 和 Standard Two 版 )
- Microsoft SQL Server 版本 2005、2008、2008R2、2012、2014、2016、2017、2019 和 2022。
  - 企业版、标准版、工作组版、开发版和 Web 版支持满载复制。
  - 企业版、标准版 ( 2016 版及更高版本 ) 和开发者版除了支持满载外，还支持 CDC ( 持续 ) 复制。
  - 不支持 Express 版本。
- MySQL 版本 5.5、5.6、5.7 和 8.0。

### Note

3.4.0 及更高 AWS DMS 版本支持 MySQL 8.0 作为源，除非事务负载被压缩。3.4.6 及更高 AWS DMS 版本支持将适用于 MySQL 8.0 的 Google Cloud 作为源。

- MariaDB ( 支持作为 MySQL 兼容数据来源 ) 版本 10.0 ( 仅版本 10.0.24 及更高版本 ) 、 10.2、10.3、10.4、10.5 和 10.6。

### Note

所有支持 MySQL 的 AWS DMS 版本都支持将 MariaDB 作为源代码。

- PostgreSQL 版本 9.4 及更高版本 ( 适用于 9.x 版本 ) 、 10.x、11.x、12.x、13.x 14.x、15.x 和 16.x。

### Note

AWS DMS 在 3.5.1 及更高版本中仅支持 PostgreSQL 15.x 版本。AWS DMS 在 3.5.3 及更高版本中仅支持 PostgreSQL 16.x 版本。

- MongoDB 版本 3.x、4.0、4.2、4.4、5.0 和 6.0

### Note

AWS DMS 3.5.0 及更高版本不支持 3.6 之前的 MongoDB 版本。

- SAP Adaptive Server Enterprise ( ASE ) 版本 12.5、15、15.5、15.7、16 及更高版本

- 适用于 Linux、UNIX 和 Windows (Db2 LUW) 的 IBM Db2 版本：
  - 版本 9.7，所有修复包
  - 版本 10.1，所有修复包
  - 版本 10.5，除修复包 5 之外的所有修复包。
  - 版本 11.1，所有修复包
  - 版本 11.5，仅修复包零的模組 ( 0-8 )
- IBM Db2 for z/OS 版本 12

#### 第三方托管数据库服务：

- Microsoft Azure SQL 数据库
- Microsoft Azure PostgreSQL Flexible Server 版本 11.2、12.15、13.11 和 15.3。
- Microsoft Azure MySQL Flexible Server 5.7 和 8。
- Google Cloud for MySQL 版本 5.6、5.7 和 8.0。
- Google Cloud for PostgreSQL 版本 9.6、10、11、12、13、14 和 15。
- OCI MySQL Heatwave 版本 8.0.34。

#### Amazon RDS 实例数据库，和 Amazon Simple Storage Service ( Amazon S3 )

- Oracle 11g ( 版本 11.2.0.4 及更高版本 ) 直至 12.2、18c 以及 19c ( Enterprise、Standard、Standard One 和 Standard Two 版 )
- Microsoft SQL Server 版本 2012、2014、2016、2017、2019 和 2022 ( Enterprise、Standard、Workgroup 和 Developer 版 )

#### Note


AWS DMS 不支持 SQL 服务器 Express。仅完全加载复制支持 Web 版本。

- MySQL 版本 5.5、5.6、5.7 和 8.0。

#### Note


3.4.0 及更高 AWS DMS 版本支持 MySQL 8.0 作为源，除非事务负载被压缩。

- MariaDB ( 支持作为与 MySQL 兼容的数据来源 ) 版本 10.0.24 到 10.0.28、10.2、10.3、10.4、10.5 和 10.6。

 Note

所有支持 MySQL 的 AWS DMS 版本都支持将 MariaDB 作为源代码。

- PostgreSQL 版本 10.x、11.x、12.x、13.x、14.x、15.x 和 16.x。

 Note

AWS DMS 在 3.5.1 及更高版本中仅支持 PostgreSQL 15.x 版本。AWS DMS 在 3.5.3 及更高版本中仅支持 PostgreSQL 16.x 版本。

- 与 MySQL 兼容的 Amazon Aurora ( 支持作为与 MySQL 兼容的数据来源 ) 。
- 与 PostgreSQL 兼容的 Amazon Aurora ( 支持作为与 PostgreSQL 兼容的数据来源 ) 。
- Amazon S3
- 亚马逊 DocumentDB ( 兼容 MongoDB ) 版本 3.6、4.0 和 5.0。
- 适用于 IBM Db2 LUW 的 Amazon RDS。

有关使用特定来源的信息，请参阅[使用 AWS DMS 端点](#)。

有关支持的目标端点的信息，请参阅[数据迁移的目标端点](#)。

## DMS Fleet Advisor 的源数据库

DMS Fleet Advisor 支持以下源数据库。

- Microsoft SQL Server 版本 2012 直至版本 2019
- MySQL 版本 5.6 直至版本 8
- Oracle 版本 11g 第 2 版直至 12c、19c 和 21c
- PostgreSQL 版本 9.6 直至版本 13

有关使用特定源的信息，请参阅[为 AWS DMS Fleet Advisor 创建数据库用户](#)。

有关 DMS Fleet Advisor 用来生成目标建议的数据库列表，请参阅[DMS Fleet Advisor 的目标](#)。

## DMS 架构转换的源数据提供程序

DMS 架构转换支持将以下数据提供程序作为迁移项目的源。

- Microsoft SQL Server 版本 2008 R2、2012、2014、2016、2017 和 2019
- Oracle 版本 10.2 及更高版本、11g 直至 12.2、18c 和 19c，以及 Oracle 数据仓库
- PostgreSQL 版本 9.2 及更高版本
- MySQL 版本 5.5 及更高版本

您的源数据提供程序可以是自管理引擎，可运行在本地，也可运行在 Amazon Elastic Compute Cloud ( Amazon EC2 ) 实例上。

有关使用特定源的信息，请参阅[在 DMS 架构转换中创建源数据提供程序](#)。

有关支持的目标数据库的信息，请参阅[DMS 架构转换的目标数据提供程序](#)。

与 DMS 架构转换相比，AWS Schema Conversion Tool (AWS SCT) 支持的源数据库和目标数据库更多。有关 AWS SCT 支持的数据库的信息，请参阅[什么是 AWS Schema Conversion Tool](#)。

## DMS 同构数据迁移的源数据提供程序

对于同构数据迁移，您可以使用以下数据提供程序作为源。

- MySQL 版本 5.7 及更高版本
- MariaDB 版本 10.2 及更高版本
- PostgreSQL 版本 10.4 至 14.x。
- MongoDB 版本 4.x、5.x、6.0
- 亚马逊 DocumentDB 版本 3.6、4.0、5.0

您的源数据提供程序可以是自管理引擎，可运行在本地，也可运行在 Amazon EC2 实例上。此外，您还可以使用 Amazon RDS 数据库实例作为源数据提供程序。

有关使用特定源的信息，请参阅[为同构数据迁移创建源数据提供器 AWS DMS](#)。

有关支持的目标数据库的信息，请参阅[DMS 同构数据迁移的目标数据提供程序](#)。

## 的目标 AWS DMS

您可以在不同的 AWS DMS 功能中使用不同的目标数据存储。以下各节包含每项 AWS DMS 功能支持的目标数据存储列表。

### 主题

- [数据迁移的目标端点](#)
- [DMS Fleet Advisor 的目标数据库](#)
- [DMS 架构转换的目标数据提供程序](#)
- [DMS 同构数据迁移的目标数据提供程序](#)

## 数据迁移的目标端点

在使用 AWS DMS 进行数据迁移时，可以将以下数据存储作为目标终端节点。

### 本地实例数据库和 Amazon EC2 实例数据库

- Oracle 版本 10g、11g、12c、18c 以及 19c ( Enterprise、Standard、Standard One 和 Standard Two 版 )
- Microsoft SQL Server 版本 2005、2008、2008R2、2012、2014、2016、2017、2019 和 2022 ( Enterprise、Standard、Workgroup 和 Developer 版 )

#### Note

AWS DMS 不支持 SQL Server Web 和 Express 版本。

- MySQL 版本 5.5、5.6、5.7 和 8.0。
- MariaDB ( 支持作为 MySQL 兼容数据目标 ) 版本 10.0.24 到 10.0.28、10.2、10.3、10.4、10.5 和 10.6。

#### Note

所有支持 MySQL 的 AWS DMS 版本都支持将 MariaDB 作为目标。

- PostgreSQL 版本 9.4 及更高版本 ( 适用于 9.x 版本 )、10.x、11.x、12.x、13.x、14.x、15.x 和 16.x。

**Note**

AWS DMS 在 3.5.1 及更高版本中仅支持 PostgreSQL 15.x。AWS DMS 在 3.5.3 及更高版本中仅支持 PostgreSQL 16.x 版本。

- SAP Adaptive Server Enterprise ( ASE ) 版本 15、15.5、15.7、16 及更高版本
- Redis 版本 6.x

亚马逊 RDS 实例数据库、亚马逊 Redshift、亚马逊 Redshift Serverless、亚马逊 DynamoDB、亚马逊 S3 OpenSearch、亚马逊 ElastiCache 服务、亚马逊 Redis 版、亚马逊 Kinesis Data Streams、亚马逊 DocumentDB、亚马逊 Neptune 和 Apache Kafka

- Oracle 版本 11g ( 版本 11.2.0.3.v1 及更高版本 )、12c、18c 以及 19c ( Enterprise、Standard、Standard One 和 Standard Two 版 )
- Microsoft SQL Server 版本 2012、2014、2016、2017、2019 和 2022 ( Enterprise、Standard、Workgroup 和 Developer 版 )

**Note**

AWS DMS 不支持 SQL Server Web 和 Express 版本。

- MySQL 版本 5.5、5.6、5.7 和 8.0。
- MariaDB ( 支持作为 MySQL 兼容数据目标 ) 版本 10.0.24 到 10.0.28、10.2、10.3、10.4、10.5 和 10.6。

**Note**

所有支持 MySQL 的 AWS DMS 版本都支持将 MariaDB 作为目标。

- PostgreSQL 版本 10.x、11.x、12.x、13.x、14.x、15.x 和 16.x。

**Note**

AWS DMS 在 3.5.1 及更高版本中仅支持 PostgreSQL 15.x。AWS DMS 在 3.5.3 及更高版本中仅支持 PostgreSQL 16.x。

- IBM Db2 LUW 版本 11.1 和 11.5



- Amazon Aurora MySQL 兼容版
- Amazon Aurora PostgreSQL 兼容版
- Amazon Aurora Serverless v2
- Amazon Redshift
- Amazon Redshift Serverless
- Amazon S3
- Amazon DynamoDB
- 亚马逊 OpenSearch 服务
- ElastiCache 适用于 Redis 的 Amazon
- Amazon Kinesis Data Streams
- Amazon DocumentDB ( 与 MongoDB 兼容 )
- Amazon Neptune
- Apache Kafka – [Amazon Managed Streaming for Apache Kafka \( Amazon MSK \)](#) 和 [自管理 Apache Kafka](#)
- 适用于 Aurora PostgreSQL ( 版本 15.3/14.8 及更高版本 ) 的 Babelfish ( 版本 3.2.0 及更高版本 )

有关使用特定目标的信息，请参阅[使用 AWS DMS 端点](#)。

有关支持的源端点的信息，请参阅[数据迁移的源端点](#)。

## DMS Fleet Advisor 的目标数据库

DMS Fleet Advisor 使用以下目标数据库的最新版本生成目标建议。

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

有关 DMS Fleet Advisor 中目标建议的信息，请参阅[使用 AWS DMS Fleet Advisor 目标建议功能](#)。

有关支持的源数据库的信息，请参阅[DMS Fleet Advisor 的源数据库](#)。

## DMS 架构转换的目标数据提供程序

DMS 架构转换支持将以下数据提供程序作为迁移项目的目标。

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

有关使用特定目标的信息，请参阅[在 DMS 架构转换中创建目标数据提供程序](#)。

有关支持的源数据库的信息，请参阅[DMS 架构转换的源数据提供程序](#)。

## DMS 同构数据迁移的目标数据提供程序

对于同构数据迁移，您可以使用以下数据提供程序作为目标。

- Amazon Aurora MySQL 版本 5.7 及更高版本
- Amazon Aurora PostgreSQL 版本 10.4 至 14.x
- Amazon Aurora Serverless v2
- Amazon RDS for MySQL 版本 5.7 及更高版本
- Amazon RDS for MariaDB 版本 10.2 及更高版本
- Amazon RDS for PostgreSQL 版本 10.4 至 14.x
- 亚马逊 DocumentDB 版本 4.0、5.0 和 DocumentDB 弹性集群

有关使用特定目标的信息，请参阅[为同构数据迁移创建目标数据提供器 AWS DMS](#)。

有关支持的源数据库的信息，请参阅[DMS 同构数据迁移的源数据提供程序](#)。

## 为其构造亚马逊资源名称 (ARN) AWS DMS

如果您使用 AWS CLI 或 AWS DMS API 自动迁移数据库，则可以使用亚马逊资源名称 (ARN)。在 Amazon Web Services 中创建的每个资源都通过 ARN 进行标识，这是唯一标识符。如果您使用 AWS CLI 或 AWS DMS API 来设置数据库迁移，则需要提供要使用的资源的 ARN。

AWS DMS 资源的 ARN 使用以下语法：

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

在此语法中，以下内容适用：

- *region* 是 AWS DMS 资源创建 AWS 区域 地的 ID，例如 `us-west-2`。

下表显示了构建 ARN 时应使用的 AWS 区域 名称和值。

区域	名称
亚太区域 ( 东京 )	ap-northeast-1
亚太区域 ( 首尔 )	ap-northeast-2
亚太地区 ( 孟买 ) 区域	ap-south-1
亚太区域 ( 新加坡 )	ap-southeast-1
亚太区域 ( 悉尼 )	ap-southeast-2
加拿大 ( 中部 ) 区域	ca-central-1
中国 ( 北京 ) 区域	cn-north-1
中国 ( 宁夏 ) 区域	cn-northwest-1
欧洲地区 ( 斯德哥尔摩 ) 区域	eu-north-1
欧洲地区 ( 米兰 )	eu-south-1
欧洲 ( 法兰克福 ) 区域	eu-central-1
欧洲地区 ( 爱尔兰 ) 区域	eu-west-1
欧洲 ( 伦敦 ) 区域	eu-west-2
欧洲地区 ( 巴黎 ) 区域	eu-west-3
南美洲 ( 圣保罗 ) 区域	sa-east-1

区域	名称
美国东部 ( 弗吉尼亚北部 ) 区域	us-east-1
美国东部 ( 俄亥俄 ) 区域	us-east-2
美国西部 ( 北加利福尼亚 ) 区域	us-west-1
美国西部 ( 俄勒冈州 ) 区域	us-west-2

- *account number* 是省略了短划线的账号。要查找您的账号，请通过 <http://aws.amazon.com> 登录您的 AWS 账户，选择“我的账户/主机”，然后选择“我的账户”。
- *resourcetype* 是 AWS DMS 资源的类型。

下表显示了为特定 AWS DMS 资源构建 ARN 时要使用的资源类型。

AWS DMS 资源类型	ARN 格式
复制实例	arn:aws:dms: <i>region</i> : <i>account</i> :rep: <i>resourcename</i>
终端节点	arn:aws:dms: <i>region</i> : <i>account</i> :endpoint: <i>resourcename</i>
复制任务	arn:aws:dms: <i>region</i> : <i>account</i> :task: <i>resourcename</i>
子网组	arn:aws:dms: <i>region</i> : <i>account</i> :subgrp: <i>resourcename</i>

- *resourcename* 是分配给资源的 AWS DMS 资源名称。这是一个随机生成的字符串。

下表显示了 AWS DMS 资源 ARN 的示例。此处，我们假设 AWS 账户为 123456789012，该账户位于美国东部 ( 弗吉尼亚州北部 ) 并具有资源名称。

资源类型	示例 ARN
复制实例	arn:aws:dms:us-east-1:123456789012:rep:QLXQZ64MH7CXF4QCQMGRVYVXAI
终端节点	arn:aws:dms:us-east-1:123456789012:endpoint:D3HMZ2IGUCGFF3NTAXUXGF6S5A

资源类型	示例 ARN
复制任务	arn:aws:dms:us-east-1:123456789012:task:2PVRE MWNPGYJCVU2IBPTOYTIV4
子网组	arn:aws:dms:us-east-1:123456789012:subgrp:test- tag-grp

## 与其他 AWS 服务 AWS DMS 一起使用

您可以与其他几项 AWS 服务 AWS DMS 一起使用：

- 您可以将 Amazon EC2 实例或 Amazon RDS 数据库实例用作数据迁移的目标。
- 您可以使用 AWS Schema Conversion Tool (AWS SCT) 将源架构和 SQL 代码转换为等效的目标架构和 SQL 代码。
- 您可以使用 Amazon S3 作为数据的存储站点，或者将其用于迁移大量数据的中间步骤。
- 您可以使用 AWS CloudFormation 来设置 AWS 资源以进行基础设施管理或部署。例如，您可以预置复制实例、任务、证书和终端节点等 AWS DMS 资源。您可以创建一个描述所需所有 AWS 资源的模板，并为您预置 AWS CloudFormation 置和配置这些资源。

## AWS DMS 支持 AWS CloudFormation

您可以使用配置 AWS DMS 资源 AWS CloudFormation。AWS CloudFormation 是一项服务，可帮助您对 AWS 资源进行建模和设置，以便进行基础设施管理或部署。例如，您可以预置复制实例、任务、证书和终端节点等 AWS DMS 资源。您可以创建一个描述所需的所有 AWS 资源的模板，并为您 AWS CloudFormation 预置和配置这些资源。

作为开发人员或系统管理员，您可以创建和管理这些资源的集合，然后将其用于重复性迁移任务或将资源部署到您的组织中。有关的更多信息 AWS CloudFormation，请参阅《AWS CloudFormation 用户指南》中的[AWS CloudFormation 概念](#)。

AWS DMS 支持使用以下资源创建以下 AWS DMS 资源 AWS CloudFormation：

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

- [AWS::DMS::ReplicationInstance](#)
- [AWS::DMS::ReplicationSubnet](#) 群组
- [AWS::DMS::ReplicationTask](#)

# AWS Database Migration Service 入门

在以下教程中，您可以了解如何使用 AWS Database Migration Service (AWS DMS) 执行数据库迁移。

要执行数据库迁移，请采取以下步骤：

1. 使用[正在设置 AWS Database Migration Service](#) 中的以下步骤设置您的 AWS 账户。
2. 创建您的示例数据库和 Amazon EC2 客户端，以填充源数据库并测试复制。此外，创建基于 Amazon Virtual Private Cloud (Amazon VPC) 服务的虚拟私有云 (VPC)，以容纳您的教程资源。要创建这些资源，请按照[的先决条件 AWS Database Migration Service](#)中的步骤操作。
3. 使用[示例数据库创建脚本](#)填充源数据库。
4. 使用 DMS Schema Conversion 或 AWS Schema Conversion Tool (AWS SCT)，将架构从源数据库转换为目标数据库。要使用 DMS Schema Conversion，请按照[开始使用 DMS 架构转换](#)中的步骤操作。要使用 AWS SCT 转换架构，请按照[迁移架构](#)中的步骤操作。
5. 创建复制实例以执行迁移的所有过程。要执行此任务和以下任务，请执行[复制](#)中的步骤。
6. 指定源和目标数据库端点。有关创建端点的信息，请参阅[创建源和目标终端节点](#)。
7. 创建一个任务来定义您要使用的表和复制过程，然后开始复制。有关创建数据库迁移任务的更多信息，请参阅[创建任务](#)。
8. 通过对目标数据库运行查询来验证复制是否在正常运行。

## 正在设置 AWS Database Migration Service

### 注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

报名参加 AWS 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建 AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行 [需要根用户访问权限的任务](#)。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 <https://aws.amazon.com/> 并选择我的账户来查看当前的账户活动并管理您的账户。

## 创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就可以不会使用 root 用户执行日常任务。

### 保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。 [AWS Management Console](#) 在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的 [以根用户身份登录](#)。

2. 为您的根用户启用多重身份验证 (MFA)。

有关说明，请参阅 [IAM 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备 \(控制台\)](#)。

### 创建具有管理访问权限的用户

1. 启用 IAM Identity Center

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 [启用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，为用户授予管理访问权限。

有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅 [《用户指南》 IAM Identity Center 目录中的使用默认设置配置 AWS IAM Identity Center 用户访问权限](#)。

### 以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。



有关使用 IAM Identity Center 用户[登录的帮助](#)，请参阅[AWS 登录 用户指南中的登录 AWS 访问门户](#)。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[创建权限集](#)。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[添加组](#)。

## 的先决条件 AWS Database Migration Service

在本节中，您可以学习的先决任务 AWS DMS，例如设置源数据库和目标数据库。作为这些任务的一部分，您还可以设置基于 Amazon VPC 服务的虚拟私有云 ( VPC ) 来容纳您的资源。此外，您还要设置一个 Amazon EC2 实例，用于填充源数据库并验证目标数据库上的复制。

### Note

填充源数据库最多需要 45 分钟。

在本教程中，您将创建一个 MariaDB 数据库作为源，并创建一个 PostgreSQL 数据库作为目标。此场景使用常用的低成本数据库引擎来演示复制。使用不同的数据库引擎演示了在异构平台之间迁移数据的 AWS DMS 功能。

本教程中使用美国西部 ( 俄勒冈州 ) 地区的资源。如果您想使用其他 AWS 区域，请改为在美国西部 ( 俄勒冈 ) 出现的任何地方指定您选择的区域。

### Note

为简单起见，您为本教程创建的数据库不使用加密或其他高级安全功能。您必须使用安全功能来保护生产数据库的安全。有关更多信息，请参阅[Amazon RDS 中的安全性](#)。

有关先决步骤，请参阅以下主题。

## 主题

- [创建 VPC](#)
- [创建 Amazon RDS 参数组](#)
- [创建源 Amazon RDS 数据库](#)
- [创建目标 Amazon RDS 数据库](#)
- [创建 Amazon EC2 客户端](#)
- [填充您的源数据库](#)

## 创建 VPC

在本节中，您将创建一个 VPC 来容纳您的 AWS 资源。在使用 AWS 资源时，使用 VPC 是最佳实践，这样您的数据库、Amazon EC2 实例、安全组等才有逻辑组织和安全。

为教程资源使用 VPC 还可以确保您在完成教程后删除使用的所有资源。您必须先删除 VPC 包含的所有资源，然后才能删除 VPC。

### 创建 VPC 以与配合使用 AWS DMS

1. 登录 AWS Management Console 并打开亚马逊 VPC 控制台，[网址为 https://console.aws.amazon.com/vpc/](https://console.aws.amazon.com/vpc/)。
2. 在导航窗格中，选择 VPC 控制面板，然后选择创建 VPC。
3. 在创建 VPC 页面上，输入以下选项：
  - 要创建的资源：VPC 等
  - 名称标签自动生成：选择自动生成，然后输入 **DMSVPC**。
  - IPv4 块：**10.0.1.0/24**
  - IPv6 CIDR 块：无 IPv6 CIDR 块
  - 租赁：默认
  - 可用区数量：2
  - 公有子网的数量：2
  - 私有子网的数量：2
  - NAT 网关 (\$)：无
  - VPC 端点：无

选择创建 VPC。

4. 在导航窗格中，选择您的 VPC。记下 DMSVPC 的 VPC ID。
5. 在导航窗格中，选择安全组。
6. 选择名为 default 的组，该组的 VPC ID 与您记下的 DMSVPC 的 ID 匹配。
7. 选择入站规则选项卡，然后选择编辑入站规则。
8. 选择 添加规则。添加一条类型为 MySQL/Aurora 的规则，然后为源选择 Anywhere-IPv4。
9. 再次选择添加规则。添加一条类型为 PostgreSQL 的规则，然后为源选择 Anywhere-IPv4。
10. 选择保存规则。

## 创建 Amazon RDS 参数组

要为的源数据库和目标数据库指定设置 AWS DMS，请使用 Amazon RDS 参数组。要允许数据库之间的初始复制和持续复制，请确保配置以下内容：

- 源数据库的二进制日志，因此 AWS DMS 可以确定需要复制哪些增量更新。
- 目标数据库的复制角色，因此在初始数据传输期间 AWS DMS 会忽略外键约束。使用此设置，AWS DMS 可以乱序迁移数据。

### 创建参数组以与一起使用 AWS DMS

1. 通过以下网址打开 Amazon RDS 控制台：<https://console.aws.amazon.com/rds/>。
2. 在导航窗格中，选择参数组。
3. 在参数组页面上，选择创建参数组。
4. 在创建参数组页上，输入以下设置：
  - 参数组系列：mariadb10.6
  - 组名：**dms-mariadb-parameters**
  - 描述：**Group for specifying binary log settings for replication**

选择创建。

5. 在参数组页面上，选择 dms-mariadb-parameters，然后在 dms-mariadb-parameters 页面上选择编辑。

6. 将以下参数设置为以下值之一：

- binlog\_checksum : NONE
- binlog\_format : ROW

选择保存更改。

7. 在参数组页面上，再次选择创建参数组。

8. 在创建参数组页面上，输入以下设置：

- 参数组系列 : postgres13
- 组名 : **dms-postgresql-parameters**
- 描述 : **Group for specifying role setting for replication**

选择创建。

9. 在参数组页面上，选择 dms-postgresql-parameters。

10. 在 dms-postgresql-parameters 页面上，选择编辑，然后将 session\_replication\_role 参数设置为 replica。请注意，session\_replication\_role 参数不在参数的第一页上。使用分页控件或搜索字段查找该参数。

11. 选择保存更改。

## 创建源 Amazon RDS 数据库

使用以下过程创建源 Amazon RDS 数据库。

### 创建源 Amazon RDS for MariaDB 数据库

1. 通过以下网址打开 Amazon RDS 控制台：<https://console.aws.amazon.com/rds/>。
2. 在控制面板页面上，选择数据库部分中的创建数据库。不要在页面顶部的尝试适用于 MySQL 和 PostgreSQL 的新 Amazon RDS Multi-AZ 部署选项部分中选择创建数据库。
3. 在创建数据库页面上，设置以下选项：
  - 选择数据库创建方法：选择标准创建。
  - 引擎选项：对于引擎类型，请选择 MariaDB。对于版本，请保留选择 MariaDB 10.6.14。
  - 模板：选择开发/测试。
  - 设置：

- 数据库实例标识符：输入 **dms-mariadb**。
- 在凭证设置部分，输入以下内容：
  - 主用户名：保留为 **admin**。
  - 取消选中 S AWS secrets Manager 中的管理主凭证。
  - 自动生成密码：保持未选中状态。
  - 主密码：输入 **changeit**。
  - 确认密码：再次输入 **changeit**。
- 实例配置：
  - 数据库实例类：保留选中标准类。
  - 对于数据库实例类别，请选择 db.m5.large。
- 存储：
  - 清除启用存储自动缩放复选框。
  - 其余设置保持原样。
- 可用性和持久性：保留选中不要创建备用实例。
- Connectivity (连接)：
  - 计算资源：保留不连接到 EC2 计算资源
  - 网络类型：保留选中 IPv4。
  - 虚拟私有云：DMSVPC-vpc
  - 公有访问权限：是。您必须启用公有访问权限才能使用 AWS Schema Conversion Tool。
  - 可用区：us-west-2a
  - 其余设置保持原样。
- 数据库身份验证：保留选中密码身份验证。
- 在监控下，清除开启性能详情复选框。展开其他配置部分，然后清除启用增强监控复选框。
- 展开其他配置：
  - 在数据库选项下，对于初始数据库名称，键入 **dms\_sample**。
  - 在数据库参数组下，选择 dms-mariadb-parameters。
  - 对于选项组，保留选中 default:mariadb-10-6。
  - 在 Backup ( 备份 ) 下，执行以下操作：
    - ~~保留选中启用自动备份。您的源数据库必须启用自动备份才能支持持续复制。~~
    - 对于备份保留期，选择 1 天。

- 在备份窗口中，保留选中无首选项。
- 清除将标签复制到快照复选框。
- 取消选中“在其他 AWS 区域启用复制”。
- 在加密下，清除启用加密复选框。
- 日志导出部分保留不变。
- 在维护下，清除允许自动次要版本升级复选框，并将维护窗口设置保留为无首选项。
- 保留启用删除保护未选中。

#### 4. 选择创建数据库。

## 创建目标 Amazon RDS 数据库

重复前面的步骤来创建您的目标 Amazon RDS 数据库，并进行以下更改。

### 创建目标 RDS for PostgreSQL 数据库

1. 重复上述过程的步骤 1 和 2。
2. 在创建数据库页面上，设置相同的选项，但以下选项除外：
  - a. 对于引擎选项，选择 PostgreSQL。
  - b. 对于版本，请选择 PostgreSQL 13.7-R1
  - c. 对于 DB instance identifier（数据库实例标识符），输入 **dms-postgresql**。
  - d. 对于主用户名，请保留选中 **postgres**。
  - e. 对于数据库参数组，选择 dms-postgresql-parameters。
  - f. 清除启用自动备份。
3. 选择创建数据库。

## 创建 Amazon EC2 客户端

在本节中，您将创建一个 Amazon EC2 客户端。您可以使用此客户端在源数据库中填充要复制的数据。您还可以使用此客户端，通过对目标数据库运行查询来验证复制。

与通过互联网访问您的数据库相比，使用 Amazon EC2 客户端访问数据库具有以下优势：

- 您可以限制只有同一 VPC 中的客户端才能访问您的数据库。

- 我们已经确认您在本教程中使用的工具可以在 Amazon Linux 2023 上运行且易于安装，推荐在本教程中使用这些工具。
- VPC 中组件之间的数据操作，性能通常比通过互联网执行的数据操作要好。

## 创建和配置 Amazon EC2 客户端以填充源数据库

1. 通过以下网址打开 Amazon EC2 控制台：<https://console.aws.amazon.com/ec2/>。
2. 在控制面板上，选择启动实例。
3. 在启动实例页面上，输入以下值：
  - a. 在名称和标签部分，对于名称，输入 **DMSClient**。
  - b. 在应用程序和操作系统映像（亚马逊机器映像）部分，按原样保留设置。
  - c. 在实例类型部分中，选择 t2.xlarge。
  - d. 在密钥对（登录）部分中，选择新建密钥对。
  - e. 在创建密钥对页面上，输入以下内容：
    - Key pair name：**DMSKeyPair**
    - 密钥对类型：保留 RSA。
    - 私钥文件格式：在 MacOS 或 Linux 上为 OpenSSH 选择 pem，在 Windows 上为 PuTTY 选择 ppk。

出现提示时保存密钥文件。

### Note

您也可以使用现有的 Amazon EC2 密钥对，而不必创建新的密钥对。

- f. 在网络设置部分中，选择编辑。选择以下设置：
  - VPC – 必填：选择具有您记下的 DMSVPC-vpc VPC 的 ID 的 VPC。
  - 子网：选择第一个公有子网。
  - 自动分配公有 IP：选择启用。

原样保留其余设置，然后选择启动实例。

## 填充您的源数据库

在本节中，您将找到源数据库和目标数据库的端点以供稍后使用，并使用以下工具填充源数据库：

- Git，用于下载填充源数据库的脚本。
- MariaDB 客户端，运行此脚本。

### 获取端点

查找并记下适用于玛丽亚数据库的 RDS 和适用于 PostgreSQL 的 RDS 数据库实例的终端节点，以备日后使用。

#### 查找数据库实例的端点

1. 登录 AWS Management Console 并打开 Amazon RDS 控制台，[网址为 https://console.aws.amazon.com/rds/](https://console.aws.amazon.com/rds/)。
2. 在导航窗格中，选择数据库。
3. 选择 dms-mariadb 数据库，并记下该数据库的端点值。
4. 对 dms-postgresql 数据库重复之前的步骤。

## 填充您的源数据库

接下来，连接到您的客户端实例，安装必要的软件，从 Git 下载 AWS 示例数据库脚本，然后运行脚本来填充源数据库。

#### 填充您的源数据库

1. 使用您在前面步骤中保存的主机名和公有密钥连接到客户端实例。

有关连接到 Amazon EC2 实例的更多信息，请参阅 Amazon EC2 用户指南中的[访问实例](#)。

#### Note

如果您使用的是 PuTTY，请在连接设置页面上启用 TCP 保持连接，这样您的连接就不会因为处于不活动状态而超时。

2. 安装 Git、MariaDB 和 PostgreSQL。根据需要确认安装。



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

3. 运行以下命令从中下载数据库创建脚本 GitHub。

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

4. 切换到 `aws-database-migration-samples/mysql/sampledb/v1/` 目录。
5. 运行以下命令。提供您之前记下的源 RDS 实例的端点，例如 `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. 运行数据库创建脚本。该脚本最多需要 45 分钟来创建架构和填充数据。您可以放心地忽略脚本显示的错误和警告。

## 使用 AWS SCT 将源架构迁移到目标数据库

在本节中，您将使用 AWS Schema Conversion Tool 将源架构迁移到目标数据库。或者，您可以使用 DMS Schema Conversion 来转换源数据库架构。有关更多信息，请参阅[开始使用 DMS 架构转换](#)。

### 使用 AWS SCT 将源架构迁移到目标数据库

1. 安装 AWS Schema Conversion Tool。有关更多信息，请参阅《AWS 用户指南》中的[安装、验证和更新 AWS SCT](#)。

下载适用于 MySQL 和 PostgreSQL 的 JDBC 驱动程序时，请记住驱动程序的保存位置，以在工具提示您输入驱动程序的位置时使用。

2. 打开 AWS Schema Conversion Tool。请选择文件，然后选择新建项目。
3. 在新建项目窗口中，设置以下值：
  - 将项目名称设置为 **DMSProject**。
  - 保持位置不变，以便将 AWS SCT 项目存储在默认文件夹中。

选择 OK (确定)。

4. 选择添加源以将源 MySQL 数据库添加到项目中，然后选择 MySQL 并选择下一步。
5. 在添加源页面上，设置以下值：
  - 连接名称：**source**
  - 服务器名称：输入您之前记下的 MySQL 数据库的端点。
  - 服务器端口：**3306**
  - 用户名：**admin**
  - 密码：**changeit**
6. 选择添加目标以将目标 Amazon RDS for PostgreSQL 数据库添加到项目中，然后选择 Amazon RDS for PostgreSQL。选择 下一步。
7. 在添加目标页面上，设置以下值：
  - 连接名称：**target**
  - 服务器名称：输入您之前记下的 PostgreSQL 数据库的端点。
  - 服务器端口：**5432**
  - 数据库：输入 PostgreSQL 数据库的名称。
  - 用户名：**postgres**
  - 密码：**changeit**
8. 在左侧窗格中，选择架构下的 dms\_sample。在右侧窗格中，选择您的目标 Amazon RDS for PostgreSQL 数据库。选择创建映射。您可以将多个映射规则添加到单个 AWS SCT 项目。有关映射规则的更多信息，请参阅[创建映射规则](#)。
9. 选择主视图。
10. 在左侧窗格中，选择架构下的 dms\_sample。打开上下文（右键单击）菜单并选择转换架构。确认该操作。

在工具转换了架构后，dms\_sample 架构将显示在右窗格中。
11. 在右侧窗格的架构下，打开 dms\_sample 的上下文（右键单击）菜单，然后选择应用到数据库。确认该操作。

验证架构迁移已完成。执行以下步骤。

## 检查架构迁移

1. 连接到您的 Amazon EC2 客户端。

2. 使用以下命令启动 PSQL 客户端。指定您的 PostgreSQL 数据库端点，并在出现提示时提供数据库密码。

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

3. 查询其中一个 (空) 表以验证 AWS SCT 已正确应用了架构。

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

## 为 AWS Database Migration Service 设置复制

在此主题中，您将在源数据库和目标数据库之间设置复制。

### 步骤 1：使用 AWS DMS 控制台创建复制实例

要开始使用 AWS DMS，请创建复制实例。

复制实例执行源端点和目标端点之间的实际数据迁移。您的实例需要足够的存储空间和处理能力，以便执行将数据从源数据库迁移到目标数据库的任务。此复制实例的大小应取决于要迁移的数据量和您的实例需要执行的任务。有关复制实例的更多信息，请参阅[使用 AWS DMS 复制实例](#)。

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

## 使用控制台创建复制实例

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 在导航窗格中，选择复制实例，然后选择创建复制实例。
3. 在创建复制实例页中，指定您的复制实例配置：
  - a. 对于 Name（名称），请输入 **DMS-instance**。
  - b. 对于描述，请输入复制实例的简短描述（可选）。
  - c. 对于实例类，请保持选中 dms.t3.medium。

该实例需要具有足够的存储空间、网络和处理能力用于进行迁移。有关如何选择实例类的更多信息，请参阅[为您的迁移选择正确的 AWS DMS 复制实例](#)。

- d. 对于引擎版本，请接受默认值。
- e. 对于多可用区，请选择开发或测试工作负载（单可用区）。
- f. 对于分配的存储空间（GiB），请接受默认值 50 GiB。

在 AWS DMS 中，存储主要由日志文件和缓存的事务使用。对于缓存事务，仅在需要将缓存的事务写入磁盘时消耗存储空间。因此，AWS DMS 不会使用大量存储空间。

- g. 对于网络类型，请选择 IPv4。
  - h. 对于 VPC，请选择 DMSVPC。
  - i. 对于复制子网组，请保留当前选择的复制子网组。
  - j. 清除可公开访问。
4. 选择高级安全和网络配置选项卡，根据需要设置网络和加密设置值。
    - a. 对于可用区，选择 us-west-2a。
    - b. 对于 VPC 安全组，如果尚未选择，请选择默认安全组。
    - c. 对于 AWS KMS key，保留（默认值）aws/dms 处于选中状态。
  5. 保留维护选项卡的设置不变。默认值为每个 AWS 区域 8 小时的时间段中随机选择的 30 分钟时段（随机选取周中的某天进行）。
  6. 选择 Create（创建）。

AWS DMS 创建复制实例来执行迁移。

## 步骤 2：指定源和目标端点

在创建复制实例时，您可以为之前创建的 Amazon RDS 指定源和目标数据存储端点。您可以单独创建每个端点。

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 DMS endpoint inherits the default DMS ARN. You cannot modify it after creation.

### 使用 AWS DMS 控制台指定源端点和数据库端点

1. 在控制台的导航窗格中，选择端点，然后选择创建端点。
2. 在创建端点页面上，选择源端点类型。选中选择 RDS 数据库实例复选框，然后选择 dms-mariadb 实例。
3. 在端点配置部分，为端点标识符输入 **dms-mysql-source**。
4. 对于源引擎，请保留 MySQL 选择。
5. 对于端点数据库的访问权限，请选择手动提供访问信息。验证端口、安全套接字层 (SSL) 模式、用户名和密码是否正确。
6. 选择测试端点连接 (可选) 选项卡。对于 VPC，请选择 DMSVPC。
7. 对于复制实例，请保留 dms-instance 选择。
8. 选择运行测试。

选择运行测试后，AWS DMS 使用您提供的详细信息创建端点并连接到该端点。如果连接失败，请编辑端点定义并再次测试连接。您也可以手动删除端点。

9. 测试成功后，选择创建端点。

10. 使用 AWS DMS 控制台指定目标数据库端点。为此，请使用以下设置重复前面的步骤：

- 端点类型：目标端点
- RDS 实例：dms-postgresql
- 端点标识符：**dms-postgresql-target**
- 目标引擎：保持 **PostgreSQL** 选中状态。

提供完端点的所有信息后，AWS DMS 创建源端点和目标端点以在数据库迁移期间使用。

### 步骤 3：创建任务并迁移数据

在此步骤中，您将创建一个任务，以在所创建的数据库之间迁移数据。

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](#)

### 创建迁移任务并开始数据库迁移

1. 在控制台导航窗格中，选择数据库迁移任务，然后选择创建任务。创建数据库迁移任务页面打开。
2. 在任务配置部分，指定以下任务选项：
  - 任务标识符：输入 **dms-task**。
  - 复制实例：选择您的复制实例 (dms-instance-vpc-**<vpc id>**)。
  - 源数据库端点：选择 dms-mysql-source。
  - 目标数据库端点：选择 dms-postgresql-target。
  - 迁移类型：选择迁移现有数据并复制持续更改。
3. 选择任务设置选项卡。设置以下设置：



- 目标表准备模式：不执行任何操作
  - 在完全加载完成后停止任务：不停止
4. 选择表映射选项卡，然后展开选择规则。选择添加新选择规则。设置以下设置：
    - 架构：输入架构
    - 架构名称：**dms\_sample**
  5. 选择迁移任务启动配置选项卡，然后选择创建时自动。
  6. 选择创建任务。

然后，AWS DMS 创建迁移任务并启动它。初始数据库复制需要大约 10 分钟。在 AWS DMS 完成数据迁移之前，请务必完成教程中的下一步。

## 步骤 4：测试复制

在本节中，您将在初始复制期间和之后将数据插入源数据库，并在目标数据库中查询插入的数据。

### 测试复制

1. 确保您的数据库迁移任务显示的状态为正在运行，但是您在上一步中启动的初始数据库复制尚未完成。
2. 连接到 Amazon EC2 客户端，并使用以下命令启动 MySQL 客户端。提供您的 MySQL 数据库端点。

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

3. 运行以下命令以将记录插入到源数据库中。

```
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. 退出 MySQL 客户端。

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

5. 在复制完成之前，在目标数据库中查询新记录。

在 Amazon EC2 实例中，使用以下命令，提供您的目标数据库端点来连接到目标数据库。

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

在系统提示时提供密码 (**changeit**)。

- 在复制完成之前，在目标数据库中查询新记录。

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

- 在迁移任务正在运行时，您可以实时监控数据库迁移的进度：

- 在 DMS 控制台导航窗格中，选择数据库迁移任务。
- 选择 dms-task。
- 选择表统计数据。

有关监控的更多信息，请参阅[监控 AWS DMS 任务](#)。

- 复制完成后，再次对目标数据库查询新记录。AWS DMS 在初始复制完成后迁移新记录。

```
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)
```

- 退出 psql 客户端。

```
dms_sample=> quit
```

- 重复步骤 1，再次连接到源数据库。

- 在 person 表中插入另一条记录。

```
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. 重复步骤 3 和 4 以断开源数据库的连接并连接到目标数据库。

13. 再次在目标数据库中查询复制的数据。

```
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)
```

## 步骤 5：清理 AWS DMS 资源

完成本入门教程后，您可以删除创建的资源。您可以使用 AWS 控制台删除它们。请确保在删除复制实例和端点之前删除迁移任务。

### 使用控制台删除迁移任务

1. 在 AWS DMS 控制台导航窗格中，选择数据库迁移任务。
2. 选择 dms-task。
3. 依次选择操作和删除。

### 使用控制台删除复制实例

1. 在 AWS DMS 控制台导航窗格中，选择复制实例。
2. 选择 DMS-instance。
3. 依次选择操作和删除。

AWS DMS 删除复制实例并将其从复制实例页面中移除。

### 使用控制台删除端点

1. 在 AWS DMS 控制台导航窗格中，选择端点。

2. 选择 dms-mysql-source。
3. 依次选择操作和删除。

删除 AWS DMS 资源后，请务必同时删除以下资源。有关删除其他服务中资源的帮助，请参阅每项服务的文档。

- 您的 RDS 数据库。
- 您的 RDS 数据库参数组。
- 您的 RDS 子网组。
- 与您的数据库和复制实例一起创建的任何 Amazon CloudWatch logs。
- 为您的 Amazon VPC 和 Amazon EC2 客户端创建的安全组。请确保将 launch-wizard-1 安全组的入站规则从默认中删除，这是删除这些安全组所必需的。
- Amazon EC2 客户端。
- Amazon VPC。
- Amazon EC2 客户端的 Amazon EC2 密钥对。

## 可用于 AWS Database Migration Service 的其他资源。

在本指南的下文中，您将了解如何使用 AWS DMS，在最广泛使用的商用和开源数据库之间迁移数据。

在准备和执行数据库迁移项目时，我们还建议您查看以下资源：

- [AWS DMS 分步迁移指南](#) – 本指南提供了分步演练，介绍将数据迁移到 AWS 的过程。
- [AWS DMS API 参考](#) – 本参考详细介绍 AWS Database Migration Service 的所有 API 操作。
- [适用于 AWS DMS 的 AWS CLI](#) – 本参考提供将 AWS Command Line Interface (AWS CLI) 和 AWS DMS 一起使用的信息。

# 使用 AWS DMS Fleet Advisor 发现和评估要进行迁移的数据库

您可以使用 DMS Fleet Advisor 从多个数据库环境中收集元数据和性能指标。这些收集的指标让您深入了解数据基础设施。[DMS Fleet Advisor](#) 从一个或多个中心位置，从您的本地数据库和分析服务器收集元数据和指标，无需将其安装在每台计算机上。目前，DMS Fleet Advisor 支持微软 SQL Server、MySQL、Oracle 和 PostgreSQL 数据库服务器的发现和指标收集。

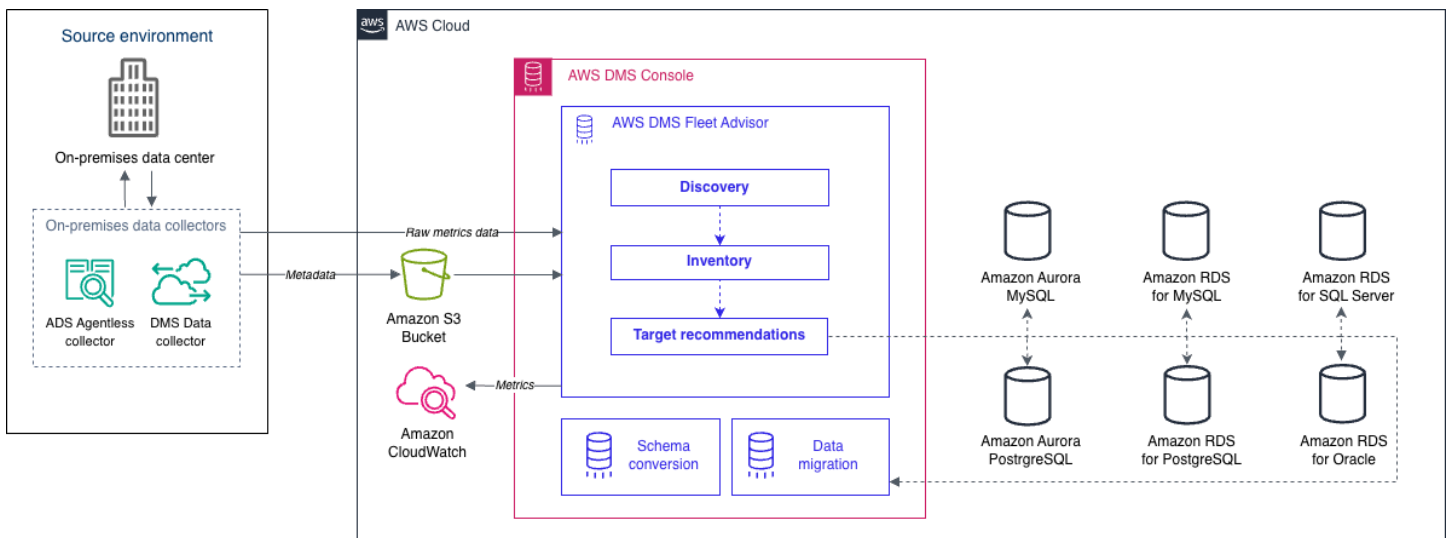
根据从网络中发现的数据，您可以构建清单来定义数据库服务器列表，以便进一步收集数据。AWS DMS 收集有关您的服务器、数据库和架构的信息后，您可以分析目标数据库迁移的可行性。

对于清单中您计划迁移到 AWS Cloud 的数据库，DMS Fleet Advisor 会生成大小合适的目标建议。为了生成目标建议，DMS Fleet Advisor 会考虑来自数据收集器的指标和首选设置。在 DMS Fleet Advisor 生成建议之后，您可以查看每个目标数据库配置的详细信息。组织的数据库工程师和管理员可以使用 DMS Fleet Advisor 目标建议，计划将其本地数据库迁移到 AWS。您可以探索不同的可用迁移选项，并将这些建议导出到 AWS Pricing Calculator 中以进一步优化成本。

有关支持的源数据库的列表，请参阅 [DMS Fleet Advisor 的源](#)。

有关 DMS Fleet Advisor 用来生成目标建议的数据库列表，请参阅 [DMS Fleet Advisor 的目标](#)。DMS Fleet Advisor 会生成类似的建议，例如，从源 Oracle 到目标 Oracle 数据库。DMS Fleet Advisor 还会生成异构建议，例如从源甲骨文或微软 SQL Server 迁移到适用于 PostgreSQL 或 Aurora PostgreSQL 数据库的目标 RDS 或 Aurora PostgreSQL 数据库。

下图说明了 AWS DMS Fleet Advisor 目标建议过程。



使用以下主题更好地了解如何使用 AWS DMS Fleet Advisor。

## 主题

- [支持 AWS 区域](#)
- [开始使用 DMS Fleet Advisor](#)
- [设置 AWS DMS Fleet Advisor](#)
- [使用数据收集器发现要迁移的数据库](#)
- [在 AWS DMS Fleet Advisor 中使用清单进行分析](#)
- [使用 AWS DMS Fleet Advisor 目标建议功能](#)
- [DMS 舰队顾问的局限性](#)

## 支持 AWS 区域

您可以在以下 AWS 区域使用 DMS Fleet Advisor。

区域名称	区域
美国东部 (弗吉尼亚州北部)	us-east-1
美国东部 (俄亥俄州)	us-east-2
美国西部 (加利福尼亚北部)	us-west-1
美国西部 (俄勒冈州)	us-west-2
亚太地区 (香港)	ap-east-1
亚太地区 (东京)	ap-northeast-1
亚太地区 (首尔)	ap-northeast-2
亚太地区 (大阪)	ap-northeast-3
亚太地区 (孟买)	ap-south-1
亚太地区 (新加坡)	ap-southeast-1

区域名称	区域
亚太地区（悉尼）	ap-southeast-2
亚太地区（雅加达）	ap-southeast-3
加拿大（中部）	ca-central-1
欧洲地区（法兰克福）	eu-central-1
欧洲地区（斯德哥尔摩）	eu-north-1
欧洲地区（爱尔兰）	eu-west-1
欧洲地区（伦敦）	eu-west-2
欧洲地区（巴黎）	eu-west-3
欧洲地区（米兰）	eu-south-3
加拿大（中部）	ca-central-1
南美洲（圣保罗）	sa-east-1
中东（巴林）	me-south-1
非洲（开普敦）	af-south-1

## 开始使用 DMS Fleet Advisor

您可以使用 DMS Fleet Advisor 来发现本地源数据库，以便迁移到 AWS Cloud。然后，可以在 AWS Cloud 中为每个本地数据库确定正确的迁移目标。可使用以下工作流程创建源数据库清单，并生成目标建议。

1. 创建 Amazon S3 存储桶、IAM 策略、角色和用户。有关更多信息，请参阅[创建所需资源](#)。
2. 创建具有 DMS 数据收集器所需最低权限的数据库用户。有关更多信息，请参阅[创建数据库用户](#)。
3. 创建和下载数据收集器。有关更多信息，请参阅[创建数据收集器](#)。
4. 在本地环境中安装数据收集器。接下来，配置数据收集器，确保它可以将收集的数据发送到 DMS Fleet Advisor。有关更多信息，请参阅[安装数据收集器](#)。

5. 发现数据环境中的操作系统服务器和数据库服务器。有关更多信息，请参阅[发现操作系统和数据库服务器](#)。
6. 收集数据库元数据和资源利用率指标。有关更多信息，请参阅[收集数据](#)。
7. 分析源数据库和架构。DMS Fleet Advisor 会对您的数据库进行大规模评估，以识别相似的架构。有关更多信息，请参阅[在 AWS DMS Fleet Advisor 中使用清单进行分析](#)。
8. 为源数据库生成、查看和保存目标建议的本地副本。有关更多信息，请参阅[目标建议](#)。

确定每个源数据库的迁移目标后，可以使用 DMS 架构转换将数据库架构转换为新平台。然后，可以使用 AWS DMS 迁移数据。有关更多信息，请参阅[使用 DMS 架构转换来转换数据库架构](#)和[什么是 AWS Database Migration Service ?](#)

[本视频](#)介绍了 DMS 架构转换用户界面，并协助您熟悉此服务的核心组件。

## 设置 AWS DMS Fleet Advisor

要设置 AWS DMS Fleet Advisor，请完成以下先决任务。

### 主题

- [为 AWS DMS Fleet Advisor 创建所需的 AWS 资源](#)
- [为 AWS DMS Fleet Advisor 创建数据库用户](#)

## 为 AWS DMS Fleet Advisor 创建所需的 AWS 资源

DMS Fleet Advisor 需要您账户中的一组 AWS 资源，用于转发和导入清单信息，以及更新 DMS 数据收集器的状态。

在首次收集数据并创建数据库和架构清单之前，请完成以下先决条件。

要配置 Amazon S3 存储桶和 IAM 资源，请执行下列操作之一：


- [使用 AWS CloudFormation 配置 Amazon S3 和 IAM 资源](#)（推荐）。
- [在 AWS Management Console 中配置 Amazon S3 和 IAM 资源](#)

## 使用 AWS CloudFormation 配置 Amazon S3 和 IAM 资源

CloudFormation 堆栈是可作为单个单元管理的一系列 AWS 资源。为了简化为 DMS Fleet Advisor 创建所需资源的过程，您可以使用 AWS CloudFormation 模板文件来创建 CloudFormation 堆栈。有关



更多信息，请参阅《AWS CloudFormation 用户指南》中的[在 AWS CloudFormation 控制台上创建堆栈](#)。

 Note

本节仅适用于使用独立的 DMS Fleet Advisor 收集器。要了解如何使用单个本地收集器收集有关数据库和服务器的信息，请参阅《[AWS Application Discovery Service 用户指南](#)》中的 [Application Discovery Service Agentless Collector](#)。

由创建的 Amazon S3 和 IAM 资源 CloudFormation

当您使用 CloudFormation 模板时，他们会创建包含以下资源的堆栈：AWS 账户

- 一个名为 `dms-fleetadvisor-data-accountId-region` 的 Amazon S3 存储桶
- 一个名为 `FleetAdvisorCollectorUser-region` 的 IAM 用户
- 一个名为 `FleetAdvisorS3Role-region` 的 IAM 服务角色
- 一个名为 `FleetAdvisorS3Role-region-Policy` 的访问策略
- 一个名为 `FleetAdvisorCollectorUser-region-Policy` 的访问策略
- 一个名为 `AWSServiceRoleForDMSFleetAdvisor` 的 IAM 服务相关角色 (SLR)

按照下面列出的步骤使用配置您的资源 CloudFormation。

- [步骤 1：下载 CloudFormation 模板文件](#)
- [第 2 步：使用配置亚马逊 S3 和 IAM CloudFormation](#)

步骤 1：下载 CloudFormation 模板文件

CloudFormation 模板是构成堆栈的 AWS 资源的声明。此模板存储为 JSON 文件。

下载 CloudFormation 模板文件

1. 打开以下链接之一的上下文（右键单击）菜单，然后选择将链接另存为：
  - 如果你打算使用 DMS Fleet Advisor，请选择 [dms-fleetadvisor-iam-slr-s3.zip](#)。如果你已经为 [DMS Fleet Advisor 创建了单反相机](#)，请选择 [3.zip dms-fleetadvisor-iam-s](#)

- 如果您计划使用 App Location Discovery Service (ADS) 无代理收集器但尚未为其创建 SLR，请选择 [slr-s3.zip](#)。如果您之前使用 ADS 为 DMS Fleet Advisor 创建过单反相机，请选择 [dms-fleetadvisor-ads-iam-s3.zip](#)。

2. 将该文件保存到您的计算机。

## 第 2 步：使用配置亚马逊 S3 和 IAM CloudFormation

当您使用 IAM CloudFormation 模板时，它会创建之前列出的 Amazon S3 和 IAM 资源。

### 要使用配置 Amazon S3 和 IAM CloudFormation

1. 打开 CloudFormation 控制台，[网址为 https://console.aws.amazon.com/cloudformation](https://console.aws.amazon.com/cloudformation)。
2. 通过在下拉列表中选择创建堆栈和使用新资源，启动“创建堆栈”向导。
3. 在 Create stack ( 创建堆栈 ) 页面上，执行以下操作：
  - a. 对于 Prepare template ( 准备模板 )，选择 Template is ready ( 模板就绪 )。
  - b. 对于 Template source(模板来源)，选择 Upload a template file(上传模板文件)。
  - c. 在“选择文件”中，导航至，然后选择 `dms-fleetadvisor-iam-slr-s3.json`、`-s3.json`、`dms-fleetadvisor-iam`、`dms-fleetadvisor-ads-iam-slr-s3.zip` 或 `dms-fleetadvisor-ads-iam-s3.zip`。
  - d. 请选择 Next ( 下一步 )。
4. 在指定堆栈详细信息页面中，请执行以下操作：
  - a. 对于堆栈名称，输入 **dms-fleetadvisor-iam-slr-s3**、**dms-fleetadvisor-iam-s3**、**dms-fleetadvisor-ads-iam-slr-s3** 或 **dms-fleetadvisor-ads-iam-s3**。
  - b. 选择 Next(下一步)。
5. 在 Configure stack options ( 配置堆栈选项 ) 页面上，请选择 Next ( 下一步 )。
6. 在 Review dms-fleetadvisor-iam-slr-s3、Review dms-fleetadvisor-iam-s3、Review dms-fleetadvisor-ads-iam-slr-s3 或 Review dms-fleetadvisor-ads-iam-s3 页面上，执行以下操作：
  - a. 选中我确认 AWS CloudFormation 可能使用自定义名称创建 IAM 资源复选框。
  - b. 选择提交。

CloudFormation 创建 DMS Fleet Advisor 所需的 S3 存储桶和 IAM 角色和用户。在左侧面板中，当 `dms-fleetadvisor-iam-slr-s3`、`3`、`-slr-s3`、`dms-fleetadvisor-iam-s3` 或 `dms-fleetadvisor-ads-iam-s3` 显示 CREATE\_COMPLETE 时，请继续下一步。

7. 在左侧面板中，选择 dms-fleetadvisor-iam-slr-s3、3、-s dms-fleetadvisor-ads-iam lr-s dms-fleetadvisor-iam-s3 或-s3。dms-fleetadvisor-ads-iam在右侧面板中，执行以下操作：
  - a. 选择堆栈信息。**##### ID ### arn: aws: cloudformation#####stack/-s3/ #####arn: aws: cloudformation#####stack/-s3/ #####arn: aws: cloudformation#####stack /-s3/ dms-fleetadvisor-iam-slr ##### arn: aws: cloudformation#####stack/-s3/ ### dms-fleetadvisor-iam-s dms-fleetadvisor-ads-iam dms-fleetadvisor-ads-iam。**
  - b. 选择资源。您将看到以下内容：
    - 一个名为 dms-fleetadvisor-data-*accountId-region* 的 Amazon S3 存储桶
    - 一个名为 FleetAdvisorS3Role-*region* 的服务角色
    - 一个名为 FleetAdvisorCollectorUser-*region* 的 IAM 用户
    - 一个名为 AWSServiceRoleForDMSFleetAdvisor 的 IAM SLR ( 如果您下载了 dms-fleet-advisor-iam-slr-s3.zip 或 dms-fleet-advisor-ads-iam-slr-s3.zip )。
    - 一个名为 FleetAdvisorS3Role-*region*-Policy 的访问策略
    - 一个名为 FleetAdvisorCollectorUser-*region*-Policy 的访问策略

## 在 AWS Management Console 中配置 Amazon S3 和 IAM 资源

### 创建 Amazon S3 存储桶

创建可以存储清单数据的 Amazon S3 存储桶。我们建议您在使用 DMS Fleet Advisor 之前，预先配置此 S3 存储桶。AWS DMS 将 DMS Fleet Advisor 清单数据存储在在此 S3 存储桶中。

有关创建 S3 存储桶的更多信息，请参阅《Amazon S3 用户指南》中的[创建第一个 S3 存储桶](#)。

#### Note

DMS Fleet Advisor 仅支持 SSE-S3 加密存储桶。

### 创建 Amazon S3 存储桶以存储本地数据环境信息

1. 登录到 AWS Management Console，然后通过以下网址打开 Amazon S3 控制台：<https://console.aws.amazon.com/s3/>。

2. 选择创建存储桶。
3. 在创建存储桶页面上，输入一个包含存储桶登录名的全局唯一名称，如 `fa-bucket-yoursignin`。
4. 选择在其中使用 DMS Fleet Advisor 的 AWS 区域。
5. 保留其余设置，然后选择创建存储桶。

## 创建 IAM 资源

在本节中，您将为数据收集器、IAM 用户和 DMS Fleet Advisor 创建 IAM 资源。

### 主题

- [为数据收集器创建 IAM 资源](#)
- [创建 DMS Fleet Advisor 服务相关角色](#)

### 为数据收集器创建 IAM 资源

为确保数据收集器正常工作，并将收集的元数据上传到 Amazon S3 存储桶，请创建以下策略。然后，创建具以下最低权限的 IAM 用户。有关 DMS 数据收集器的更多信息，请参阅[使用数据收集器发现要迁移的数据库](#)。

为 DMS Fleet Advisor 和数据收集器创建 IAM policy 以访问 Amazon S3

1. 登录到 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择策略。
3. 选择创建策略。
4. 在创建策略页面中，选择 JSON 选项卡。
5. 将以下 JSON 粘贴到编辑器中，替换示例代码。将 `fa_bucket` 替换为在上一节中创建的 Amazon S3 存储桶的名称。

```
{
  "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. 选择下一步：标签和下一步：审核。
7. 在名称\* 中输入 **FleetAdvisorS3Policy**，然后选择创建策略。

为 DMS 数据收集器创建 IAM policy 以访问 DMS Fleet Advisor

1. 登录到 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择策略。
3. 选择创建策略。
4. 在创建策略页面中，选择 JSON 选项卡。
5. 将以下 JSON 代码粘贴到编辑器中，替换示例代码。

```

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

```

6. 选择下一步：标签和下一步：审核。

7. 在名称\* 中输入 **DMSCollectorPolicy**，然后选择创建策略。

### 创建具最低权限的 IAM 用户以使用 DMS 数据收集器

1. 登录 AWS Management Console，单击 <https://console.aws.amazon.com/iam/> 打开 IAM 控制台。
2. 在导航窗格中，选择用户。
3. 选择添加用户。
4. 在添加用户页面上，在用户名\* 中输入 **FleetAdvisorCollectorUser**。对于选择 AWS 访问类型，选择访问密钥 – 编程访问。选择下一步: 权限。
5. 在设置权限部分中，选择直接附加现有策略。
6. 使用搜索控件查找并选择您之前创建的 DMS CollectorPolicy 和 FleetAdvisorS3Policy 策略。选择下一步：标签。
7. 在标签页面上，选择下一步：审核。
8. 在审核页面上，选择创建用户。在下个页面上，选择下载 .csv 以保存新的用户凭证。这些凭证可与 DMS Fleet Advisor 配合使用，以获得所需的最低访问权限。

### 为 DMS Fleet Advisor 和数据收集器创建 IAM 角色以访问 Amazon S3

1. 登录 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择角色。
3. 选择创建角色。
4. 在选择可信实体页面中，在可信实体类型下选择 AWS 服务。对于其他 AWS 服务的应用场景，选择 DMS。
5. 选中 DMS 复选框，然后选择下一步。
6. 在添加权限页面上，选择 FleetAdvisorS3 Policy。选择下一步。
7. 在命名、检查并创建页面上，在角色名称中输入 **FleetAdvisorS3Role**，然后选择创建角色。
8. 在角色页面上，在角色名称中输入 **FleetAdvisorS3Role**。选择 FleetAdvisorS3Role。
9. 在 FleetAdvisorS3Role 页面上，选择信任关系选项卡。选择编辑信任策略。
10. 在编辑信任策略页面上，将以下 JSON 粘贴到编辑器中，替换现有文本。

```
{
```

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

上述策略向 AWS DMS 用于从 Amazon S3 存储桶导入收集的数据的服务授予 `sts:AssumeRole` 权限。

## 11. 选择更新策略。

### 创建 DMS Fleet Advisor 服务相关角色

DMS Fleet Advisor 使用服务相关角色来管理您的中的亚马逊 CloudWatch 指标。AWS 账户 DMS Fleet Advisor 使用此服务相关角色代表您向 CloudWatch 发布收集的数据库性能指标。

### 为 DMS Fleet Advisor 创建服务相关角色

1. 登录 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择角色。然后，选择创建角色。
3. 在可信实体类型中选择 AWS 服务。
4. 对于其他 AWS 服务的应用场景，选择 DMS – Fleet Advisor。
5. 选中 DMS – Fleet Advisor 复选框，然后选择下一步。
6. 在添加权限页面上，选择下一步。
7. 在命名、检查并创建页面上，选择创建角色。

或者，您可以通过 AWS API 或 AWS CLI 创建此服务相关角色。有关更多信息，请参见 [为 AWS DMS Fleet Advisor 创建服务相关角色](#)。

为 DMS Fleet Advisor 创建服务相关角色后，可以在目标建议中看到源数据库的性能指标。此外，您还可以在自己的 CloudWatch 账户中看到这些指标。有关更多信息，请参见 [目标建议](#)。

### 创建 DMS Fleet Advisor 服务相关角色所需的 IAM policy

DMSFleetAdvisorCreateServiceLinkedRolePolicy 策略中指定了创建服务相关角色所需的最低权限。如果您无法创建服务相关角色，请为您的账户创建此 IAM 策略。

1. 登录到 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择策略。
3. 选择创建策略。
4. 在创建策略页面中，选择 JSON 选项卡。
5. 将以下 JSON 代码粘贴到编辑器中，替换示例代码。

```
{
  "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. 选择下一步：标签和下一步：审核。
7. 在名称\* 中输入 **DMSFleetAdvisorCreateServiceLinkedRolePolicy**，然后选择创建策略。



现在，您可以使用此策略，为 DMS Fleet Advisor 创建服务相关角色。

## 为 AWS DMS Fleet Advisor 创建数据库用户

本节描述了如何为源数据库创建具 DMS 数据收集器所需最低权限的用户。

本节包含以下主题：

- [将数据库用户用于 AWS DMS Fleet Advisor](#)
- [使用 MySQL 创建数据库用户](#)
- [使用 Oracle 创建数据库用户](#)
- [使用 PostgreSQL 创建数据库用户](#)
- [使用 Microsoft SQL Server 创建数据库用户](#)
- [删除数据库用户](#)

### 将数据库用户用于 AWS DMS Fleet Advisor

您可以将除 root 之外的数据库用户，用于 DMS 数据收集器。请在将数据库添加到清单之后，但在运行数据收集器之前，指定用户名和密码。有关将数据库添加到清单的更多信息，请参阅[管理受监控对象](#)。

DMS 数据收集器使用完成之后，可以删除所创建的数据库用户。有关更多信息，请参见[删除数据库用户](#)。

#### Important

在以下示例中，将 `{your_user_name}` 替换为您为数据库创建的数据库用户的名称。然后，将 `{your_password}` 替换为安全密码。

### 使用 MySQL 创建数据库用户

要在 MySQL 源数据库中创建数据库用户，请使用以下脚本。请确保根据 MySQL 数据库版本，保留 GRANT 语句的一个版本。

```
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;

```

## 使用 Oracle 创建数据库用户

要在 Oracle 源数据库中创建数据库用户，请使用以下脚本。

要运行此 SQL 脚本，请使用 SYSDBA 权限连接到 Oracle 数据库。运行此 SQL 脚本之后，通过使用此脚本创建的用户凭证连接到数据库。此外，还可以使用此用户的凭证运行 DMS 数据收集器。

以下脚本在 Oracle 多租户容器数据库 (CDB) 的用户名中添加了 C## 前缀。

```

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;
```

## 使用 PostgreSQL 创建数据库用户

要在 PostgreSQL 源数据库中创建数据库用户，请使用以下脚本。

```
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}";
```

## 使用 Microsoft SQL Server 创建数据库用户

要在 Microsoft SQL Server 源数据库中创建数据库用户，请使用以下脚本。

```
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}]

```

## 删除数据库用户

完成所有数据收集任务之后，可以删除为 DMS 数据收集器创建的数据库用户。您可以使用以下脚本，从数据库中删除具最低权限的用户。

要从 MySQL 数据库中删除用户，请运行以下脚本。

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

要从 Oracle 数据库中删除用户，请运行以下脚本。

```

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;

```

```
END IF;  
END;
```

要从 PostgreSQL 数据库中删除用户，请运行以下脚本。

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

要从 SQL Server 数据库中删除用户，请运行以下脚本。

```
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
```

## 使用数据收集器发现要迁移的数据库

要发现您的源数据基础架构，可以使用 [AWS Application Discovery Service Agentless Collector](#) 或 AWS DMS 数据收集器。ADS Agentless Collector 是一种本地应用程序，它通过无代理方法收集有关本地环境的信息，包括服务器配置文件信息（例如，操作系统、CPU 数量、RAM 量）、数据库元数据和利用率指标。使用开放虚拟化归档 (OVA) 文件在您的 VMware vCenter Server 环境中将 Agentless Collector 安装为虚拟机 (VM)。AWS DMS 数据收集器是安装在本地环境中的 Windows 应用程序。此应用程序可连接到数据环境，并从本地数据库和分析服务器收集元数据和性能指标。通过 ADS Agentless Collector 或 DMS 数据收集器收集数据库元数据和性能指标后，DMS Fleet Advisor 会生成一份可以迁移到 AWS Cloud 的服务器、数据库和架构的清单。

DMS 数据收集器是一个 Windows 应用程序，它使用 .NET 库、连接器和数据提供程序连接到您的源数据库以进行数据库发现和收集。

DMS 数据收集器在 Windows 上运行。不过，DMS 数据收集器可以从所有支持的数据库供应商收集数据，而无论它们运行于哪个操作系统服务器上。

DMS 数据收集器使用受保护的 RTPS 协议和 TLS 加密，与 DMS Fleet Advisor 建立安全连接。因此，默认情况下，数据在传输过程中会进行加密。

AWS DMS 具有您可以为 AWS 账户创建的最大数量的数据收集器。有关 AWS DMS 服务限额的信息，请参阅以下章节：[AWS Database Migration Service 的限额](#)。

## 主题

- [DMS 数据收集器的权限](#)
- [为 AWS DMS Fleet Advisor 创建数据收集器](#)
- [安装并配置数据收集器](#)
- [发现要监控的操作系统和数据库服务器](#)
- [管理受监控对象](#)
- [在 AWS DMS Fleet Advisor 中使用 SSL](#)
- [为 AWS DMS Fleet Advisor 收集数据](#)
- [DMS 数据收集器故障排除](#)

## DMS 数据收集器的权限

您为 DMS 数据收集器创建的数据库用户应具有读取权限。但是，在某些情况下，数据库用户需要 EXECUTE 权限。有关更多信息，请参见 [为 AWS DMS Fleet Advisor 创建数据库用户](#)。

DMS 数据收集器需要额外的权限才能运行发现脚本。

- 对于操作系统发现，DMS 数据收集器需要域服务器凭证，才能使用 LDAP 协议运行请求。
- 对于 Linux 中的数据库发现，DMS 数据收集器需要具 sudo SSH 授权的凭证。此外，还应当配置 Linux 服务器以允许运行远程 SSH 脚本。
- 对于 Windows 中的数据库发现，DMS 数据收集器需要具以下权限的凭证：运行 Windows Management Instrumentation (WMI) 和 WMI Query Language (WQL) 查询并读取注册表。此外，您还应将 Windows 服务器配置为允许运行远程 WMI、WQL 和 PowerShell 脚本。

## 为 AWS DMS Fleet Advisor 创建数据收集器

了解如何创建和下载 DMS 数据收集器。

在创建数据收集器之前，请使用 IAM 控制台为 DMS Fleet Advisor 创建服务相关角色。该角色允许委托人向 Amazon CloudWatch 发布指标数据点。DMS Fleet Advisor 使用此角色显示包含数据库指标的图表。有关更多信息，请参见 [为 AWS DMS Fleet Advisor 创建服务相关角色](#)。



## 创建和下载数据收集器

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。



选择使用 DMS Fleet Advisor 的区域。

2. 在导航窗格中，在发现下选择数据收集器。数据收集器页面将打开。
3. 选择创建数据收集器。创建数据收集器页面将打开。

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](#)

- 在常规配置部分的名称中，输入数据收集器的名称。

- 在连接部分中，选择浏览 S3。从显示的列表中选择预配置的 Amazon S3 存储桶。

AWS DMS 会将 DMS Fleet Advisor 清单数据存储在 S3 存储桶中。请确保 Amazon S3 存储桶位于 AWS DMS Fleet Advisor 当前运行所在的同一 AWS 区域中。

#### Note

DMS Fleet Advisor 仅支持 SSE-S3 加密存储桶。

- 在 IAM 角色列表中，从显示的列表中选择预配置的 IAM 角色。此角色授予 AWS DMS 访问指定 Amazon S3 存储桶的权限。

- 选择创建数据收集器。数据收集器页面将打开，且创建的数据收集器出现在列表中。

创建第一个数据收集器时，AWS DMS 会在 Amazon S3 存储桶中配置一个环境，该环境可格式化数据并存储属性，以便用于 DMS Fleet Advisor。

- 在信息横幅上选择下载本地收集器，以下载新创建的数据收集器。将显示一条消息，通知您正在进行下载。下载完成之后，可以访问 `AWS_DMS_Collector_Installer_<version_number>.msi` 文件。

现在，您可以在客户端上安装 DMS 数据收集器。有关更多信息，请参见 [安装并配置数据收集器](#)。

## 安装并配置数据收集器

了解如何安装 DMS 数据收集器，如何指定数据转发凭证，以及如何将 LDAP 服务器添加到项目。

下表描述了安装 DMS 数据收集器的硬件和软件要求。

最低	推荐
处理器：2 个内核，CPU 基准测试分数大于 8000	处理器：4 个内核，CPU 基准测试分数大于 11000
内存：8 GB	内存：16 GB
硬盘驱动器大小：256 GB	硬盘驱动器大小：512 GB
操作系统：Microsoft Windows Server 2012 或更高版本	操作系统：Windows Server 2016 或更高版本

## 在网络中的客户端上安装数据收集器

1. 运行 .MSI 安装程序。将显示 AWS DMSFleet Advisor 收集器设置向导页面。
2. 请选择 Next ( 下一步 )。将显示最终用户许可协议。
3. 阅读并接受最终用户许可协议。
4. 请选择 Next ( 下一步 )。将显示目标文件夹页面。
5. 选择下一步，在默认目录中安装数据收集器。

或者选择更改，进入其他安装目录。然后选择下一步。

6. 在桌面快捷方式页面上，选中要在桌面上安装图标的复选框。
7. 选择安装。数据收集器将安装在您选择的目录中。
8. 在已完成 DMS 收集器安装向导页面上，选择启动 AWS DMS 收集器，然后选择完成。

DMS 数据收集器使用 .NET 库、连接器和数据提供程序连接到源数据库。DMS 数据收集器安装程序会为服务器上所有支持的数据库自动安装此必需的软件。

安装完数据收集器后，可以输入 **http://localhost:11000/** 作为地址，从浏览器运行数据收集器。或者，从 Microsoft Windows 的开始菜单中，在程序列表中选择 AWS DMS 收集器。首次运行 DMS 数据收集器时，系统会要求您配置凭证。请创建用于登录数据收集器的用户名和密码。

在 DMS 数据收集器主页上，您可以找到有关准备和运行元数据收集的信息，包括以下状态条件：

- 数据收集的状态和运行状况。
- Amazon S3 存储桶及 AWS DMS 的可访问性，以便数据收集器可将数据转发到 AWS DMS。
- 已安装的数据库驱动程序的连接性。
- 用于执行初始发现的 LDAP 服务器凭证。

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

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

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

LDAP server host name	User name	Connection status
dc01.dbm.local	shareduser	Passed

DMS 数据收集器使用 LDAP 目录收集有关网络中计算机和数据库服务器的信息。轻型目录访问协议 (LDAP) 是一种开放标准应用程序协议。它用于通过 IP 网络访问和维护分布式目录信息服务。您可以将现有 LDAP 服务器添加到数据收集器项目中，以便您用于发现有关系统基础设施的信息。为此，请选择 +服务器选项，然后为域控制器指定完全限定域名 (FQDN) 和凭证。添加服务器后，验证连接检查。要开始发现过程，请参阅[发现要监控的操作系统和数据库服务器](#)。

## 配置数据转发凭证

安装数据收集器后，请确保此应用程序可以将收集的数据发送到 AWS DMS Fleet Advisor。

在 AWS DMS Fleet Advisor 中配置数据转发凭证

1. 在 DMS 数据收集器主页上，在数据转发部分中选择配置转发。将打开配置数据转发凭证对话框。
2. 选择要在其中使用 DMS Fleet Advisor 的 AWS 区域。
3. 输入之前在创建 IAM 资源时获得的 AWS 访问密钥 ID 和 AWS 秘密访问密钥。有关更多信息，请参见 [创建 IAM 资源](#)。
4. 选择浏览数据收集器。

如果您尚未在指定区域内创建数据收集器，请先创建数据收集器，然后再继续。有关更多信息，请参见 [创建数据收集器](#)。

5. 在选择数据收集器窗口中，从列表选择一个数据收集器，然后选择选择。
6. 在配置转发凭证对话框中，选择保存。

在 DMS 收集器主页上，在数据转发卡中，验证访问 Amazon S3 和访问 AWS DMS 是否设置为是。

如果您看到访问 Amazon S3 或访问 AWS DMS 的状态设置为否，请确保已创建了用于访问 Amazon S3 和 DMS Fleet Advisor 的 IAM 资源。在创建了具备所有必需权限的 IAM 资源之后，请重新配置数据转发。有关更多信息，请参见 [创建 IAM 资源](#)。

## 发现要监控的操作系统和数据库服务器

您可以使用 DMS 数据收集器，查找并列出现网中所有可用的服务器。发现网中所有可用的数据库服务器只是一个建议，并不是必需的。或者，您也可以手动添加或上传服务器列表，以供进一步的数据收集。有关手动添加服务器列表的更多信息，请参阅 [管理受监控对象](#)。

我们建议您先发现所有操作系统 (OS) 服务器，然后再发现这些服务器上的数据库。要发现操作系统服务器，你需要运行远程 PowerShell、安全外壳 (SSH) 和 Windows 管理工具 (WMI) 脚本和命令的权限，以及访问 Windows 注册表的权限。要发现网中的数据库服务器并从中收集元数据，您需要具备远程数据库连接的只读管理员权限。在继续发现之前，请确保添加了 LDAP 服务器。有关更多信息，请参见 [配置数据转发凭证](#)。

要开始使用 DMS 数据收集器，请完成以下任务：

- 发现网中的所有操作系统服务器。
- 将特定操作系统服务器添加为要监控的对象。
- 验证受监控操作系统服务器的连接。
- 发现操作系统服务器上运行的 Microsoft SQL Server、MySQL、Oracle 和 PostgreSQL 数据库。
- 添加用于数据收集的数据库服务器。
- 验证与受监控数据库的连接。

### 在网中发现可以监控的操作系统服务器

1. 在 DMS 数据收集器导航窗格中，选择发现。要显示导航窗格，请选择 DMS 数据收集器主页左上角的菜单图标。

将打开发现页面。

2. 确保选择了操作系统服务器选项卡，然后选择运行发现。将显示发现参数对话框。
3. 输入要用于扫描网的 LDAP 服务器。
4. 选择运行发现。页面将显示在网中发现的所有操作系统服务器的列表，而无论服务器是否在运行数据库。

我们建议您先运行所有操作系统服务器发现，然后再运行这些服务器上的数据库发现。凭证使得首先发现主机服务器，然后发现驻留其上的数据库成为可能。您需要先发现操作系统服务器，然后再对这些服务器上的数据库运行发现。请注意，用于 LDAP 服务器在网络中查找操作系统服务器的凭证，可能与发现特定操作系统服务器上的数据库所需的凭证不同。因此，我们建议您将操作系统服务器添加为受监控对象，验证凭证并在必要时进行更正，然后在继续操作之前检查连接性。

现在，您可以在网络中已发现操作系统服务器的列表中，选择要添加为受监控对象的服务器。

### 选择要作为受监控对象的操作系统服务器

1. 在发现页面上，选择操作系统服务器选项卡。
2. 在显示的已发现操作系统服务器的列表中，选中要监控的每个服务器旁边的复选框。
3. 选择添加到受监控对象。

您可以在监控对象页面上，查看要监控并验证连接的操作系统服务器的列表。

### 验证要监控的选定操作系统服务器的连接

1. 在 DMS 数据收集器导航窗格中，选择受监控对象。
2. 在受监控对象页面上，选择操作系统服务器选项卡。将显示要监控的已发现操作系统服务器的列表。
3. 选中列顶部的复选框，以选择所有列出的操作系统服务器。
4. 选择操作，然后选择验证连接。对于每个服务器对象，在连接状态列中查看结果。
5. 选择连接状态不是成功的服务器。接下来，选择操作，然后选择编辑。将打开编辑服务器对话框。
6. 验证信息是否正确，或根据需要进行编辑。完成后，选择保存。将打开覆盖凭证对话框。
7. 选择覆盖。DMS 数据收集器验证每个连接的状态并更新为成功。

现在，您可以发现选定要监控的服务器上所驻留的数据库。

### 发现服务器上运行的数据库

1. 在 DMS 数据收集器导航窗格中，选择发现。
2. 选择数据库服务器选项卡，然后选择运行发现。将打开发现参数对话框。
3. 在发现参数对话框中，在发现依据中选择受监控对象。对于服务器，选择要在其上运行数据库发现的操作系统服务器。

4. 选择运行发现。页面将显示您选择监控的操作系统服务器上所驻留的所有数据库的列表。

查看数据库地址、服务器名称和数据库引擎等信息，以协助您选择要监控的数据库。

### 选择要监控的数据库

1. 在发现页面上，选择数据库服务器选项卡。
2. 在显示的发现的数据库列表中，选中要监控的所有数据库旁边的复选框。
3. 选择添加到受监控对象。

现在，您可以验证选择要监控的数据库的连接。

### 验证与受监控数据库的连接

1. 在 DMS 数据收集器导航窗格中，选择受监控对象。
2. 在受监控对象页面上，选择数据库服务器选项卡。将显示选择要监控的已发现数据库服务器的列表。
3. 选中列顶部的复选框，以选择所有列出的数据库服务器。
4. 选择操作，然后选择验证连接。对于每个数据库，在连接状态列中查看结果。
5. 选择具有未定义（空白）状态或失败状态的连接。接下来，选择操作，然后选择编辑。将打开编辑受监控对象对话框。
6. 输入登录和密码凭证，然后选择保存。将打开更改凭证对话框。
7. 选择覆盖。DMS 数据收集器验证每个连接的状态并更新为成功。

在发现要监控的操作系统服务器和数据库后，您还可以执行操作来管理受监控对象。

## 管理受监控对象

在运行服务器发现过程时，您可以选择要监控的对象，如[发现操作系统和数据库服务器](#)中所述。并且，您还可以手动管理受监控对象，例如操作系统 (OS) 服务器和数据库服务器。您可以执行以下操作来管理受监控对象：

- 添加要监控的新对象
- 删除现有对象
- 编辑现有对象



- 导出和导入要监控的对象列表
- 检查与对象的连接
- 启动数据收集

例如，您可以手动添加要监控的对象。

#### 手动添加要监控的对象

1. 在受监控对象页面上，选择 +服务器。将打开添加受监控对象对话框。
2. 添加有关服务器的信息，然后选择保存。

您也可以使用 .csv 文件导入大量要监控的对象。可使用以下 .csv 文件格式，将对象列表导入 DMS 数据收集器。

```
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
```

#### 导入包含要监控的对象列表的 .csv 文件

1. 选择导入。将打开导入受监控对象页面。
2. 浏览至要导入的 .csv 文件，然后选择下一步。

您可以查看所有对象，并选择要从中开始收集元数据的对象。

## 将操作系统服务器与手动添加的数据库关联

DMS Fleet Advisor 无法直接从 MySQL 和 PostgreSQL 数据库收集性能指标。为了收集目标建议所需的指标，DMS Fleet Advisor 使用运行数据库的操作系统指标。

当您将 MySQL 和 PostgreSQL 数据库手动添加到受监控对象列表时，DMS 数据收集器无法识别运行这些数据库的操作系统服务器。因此，您应当将 MySQL 和 PostgreSQL 数据库与操作系统服务器相关联。

您无需手动将操作系统服务器与 DMS Fleet Advisor 自动发现的数据库相关联。

### 将操作系统服务器与数据库关联

1. 在 DMS 数据收集器导航窗格中，选择受监控对象。
2. 在受监控对象页面上，选择数据库服务器选项卡。将显示数据库服务器列表。
3. 选中手动添加的 MySQL 或 PostgreSQL 数据库服务器旁边的复选框。
4. 选择操作，然后选择编辑。将打开编辑数据库对话框。
5. 如果 DMS 数据收集器已经发现了运行此数据库的操作系统服务器，则选择自动检测。DMS 数据收集器会运行 SQL 脚本，自动识别运行数据库的操作系统服务器。然后，DMS 数据收集器将此操作系统服务器与数据库关联。跳过下一步，并保存您编辑的数据库配置。

如果 DMS 数据收集器无法自动识别数据库的操作系统服务器，请确保使用正确的凭证并提供数据库访问权限。或者，您可以选择手动添加操作系统服务器。

6. 要手动添加操作系统服务器，请选择 +添加操作系统服务器。将打开添加主机操作系统服务器对话框。

添加有关操作系统服务器的信息，然后选择保存。

7. 在编辑数据库对话框中，选择验证连接，以确保 DMS 数据收集器可以连接到操作系统服务器。
8. 验证连接后，选择保存。

如果您更改了源数据库的关联操作系统服务器，则 DMS Fleet Advisor 会使用更新后的指标来生成建议。但是，Amazon CloudWatch 图表显示的是您的数据库服务器的旧数据。有关 CloudWatch 图表的更多信息，请参阅[建议详细信息](#)。

## 在 AWS DMS Fleet Advisor 中使用 SSL

为了保护数据，AWS DMS Fleet Advisor 可以使用 SSL 访问您的数据库。

## 支持的数据库

AWS DMS Fleet Advisor 支持使用 SSL 访问以下数据库：

- Microsoft SQL Server
- MySQL
- PostgreSQL

## 设置 SSL

要使用 SSL 访问数据库，请配置数据库服务器以支持 SSL。有关更多信息，请参阅以下数据库文档：

- SQL Server：[启用与数据库引擎的加密连接](#)
- MySQL：[配置 MySQL 以使用加密连接](#)
- PostgreSQL：[使用 SSL 保护 TCP/IP 连接](#)

要使用 SSL 连接到数据库，请在手动添加服务器时，选择信任服务器证书和使用 SSL。对于 MySQL 数据库，可以使用自定义证书。要使用自定义证书，请选中验证 CA 复选框。有关添加服务器的信息，请参阅[管理受监控对象](#)。

## 检查适用于 SQL Server 的服务器证书颁发机构 (CA) 证书

如果要验证适用于 SQL Server 的服务器证书颁发机构 (CA) 证书，请在添加服务器时清除信任服务器证书。如果服务器使用知名的 CA，并且操作系统上默认安装了该 CA，那么验证应该可正常进行。如果 DMS Fleet Advisor 无法连接到数据库服务器，请安装数据库服务器使用的 CA 证书。有关更多信息，请参阅[配置客户端](#)。

## 为 AWS DMS Fleet Advisor 收集数据

要开始收集数据，请在受监控对象页面上选择对象，然后选择运行数据收集。DMS 数据收集器一次最多可从 100 个数据库收集数据。此外，DMS 数据收集器最多可使用八个并行线程来连接环境中的数据库。在这八个线程中，DMS 数据收集器最多可使用五个并行线程连接到单个数据库实例。

### Important

在开始收集数据之前，请查看 DMS 数据收集器主页上的软件检查部分。确认要监控的所有数据库引擎的状态均为已通过。如果某些数据库引擎的状态为失败，并且受监控对象列表中包

含有相应引擎的数据库服务器，请在继续操作之前修复此问题。您可以在软件检查部分列出的失败状态旁边找到提示。

DMS 数据收集器可以在两种模式下工作：单次运行或持续监控。开始数据收集后，将打开运行数据收集对话框。接下来，请选择以下两个选项中的一个。

### 元数据和数据库容量

DMS 数据收集器从数据库或操作系统服务器收集信息。包括架构、版本、发行版、CPU、内存和磁盘容量。DMS 数据收集器还收集和提供 IOPS、I/O 吞吐量和数据库服务器活动连接等指标。您可以根据这些信息，在 DMS Fleet Advisor 中计算目标建议。如果源数据库的预置过度或不足，那么目标建议也将预置过度或不足。

这是默认选项。

### 元数据、数据库容量和资源利用率

除了元数据和数据库容量信息之外，DMS 数据收集器还收集数据库或操作系统服务器的 CPU、内存和磁盘容量的实际利用率指标。DMS 数据收集器还收集和提供 IOPS、I/O 吞吐量和数据库服务器活动连接等指标。提供的目标建议将更加准确，因为它们基于实际的数据库工作负载。

如果选择此选项，则可设置数据收集周期。您可以在接下来的 7 天内收集数据，也可以设置自定义范围（1-60 天）。

数据收集开始后，您将被重定向到数据收集页面，您可以在其中查看收集查询的运行方式并监控实时进度。您可以在该页面或 DMS 数据收集器主页上，查看收集的总体运行状况。如果数据收集的总体运行状况低于 100%，则可能需要修复与收集有关的问题。

如果您在元数据和数据库容量模式下运行 DMS 数据收集器，那么可以在数据收集页面上看到已完成的查询数量。

如果您在元数据、数据库容量和资源利用率模式下运行 DMS 数据收集器，那么可以看到 DMS 数据收集器完成监控之前的剩余时间。

在数据收集页面上，可以看到每个对象的收集状态。如果某些方面无法正常工作，则会出现一条消息，显示发生了多少问题。您可以查看详细信息，以协助确定问题的解决方法。以下选项卡列出了潜在的问题：

- 按查询汇总 – 显示某些测试（如 Ping）的状态。可以在状态列中筛选结果。状态列中提供了消息，指示在数据收集过程中发生了多少故障。

- 按受监控对象汇总 – 显示每个对象的总体状态。
- 按查询类型汇总 – 显示收集器查询类型的状态，例如 SQL、Secure Shell (SSH) 或 Windows Management Instrumentation (WMI) 调用。
- 按问题汇总 – 显示发生的所有独特的问题，包括问题名称和每个问题发生的次数。

**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 address	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

要导出收集结果，请选择导出为 CSV。

确定并解决问题后，选择开始收集，然后重新运行数据收集过程。执行数据收集后，数据收集器使用安全连接将收集的数据上传至 DMS Fleet Advisor 清单。DMS Fleet Advisor 将信息存储在 Amazon S3 存储桶中。有关配置数据转发凭证的信息，请参阅[配置数据转发凭证](#)。

## 使用 AWS DMS Fleet Advisor 收集容量和资源利用率指标

您可以通过两种模式收集元数据和性能指标：单次运行或持续监控。DMS 数据收集器根据您选择的选项，跟踪数据环境中的不同指标。在单次运行期间，DMS 数据收集器仅跟踪来自数据库和操作系统服务器的元数据指标。在持续监控期间，DMS 数据收集器会跟踪资源的实际利用率。

AWS DMS 在单次运行 DMS 数据收集器期间收集以下元数据和指标。

- 操作系统服务器上的可用内存

- 操作系统服务器上的可用存储
- 数据库版本和发行版
- 操作系统服务器上的 CPU 数量
- 架构的数量
- 已存储程序的数量
- 表的数量
- 触发器的数量
- 视图的数量
- 架构结构

DMS Fleet Advisor 使用上述指标来构建数据库和操作系统服务器的清单。此外，DMS Fleet Advisor 还使用这些元数据和指标来分析您的源数据库架构。

DMS Fleet Advisor 可以使用在一次数据收集器运行期间收集的指标来生成目标建议。但是，在这种情况下，对于过度配置的源数据库，目标建议也被过度置备。因此，在维护资源方面会产生额外的费用。AWS Cloud 对于预置不足的源数据库，目标建议也预置不足，这可能会导致性能问题。我们建议使用持续监控来收集数据，方法是为 DMS 数据收集器选择元数据、数据库容量和资源利用模式。

在持续监控期间，AWS DMS 收集以下指标。您可以运行为期 1 到 60 天的 DMS 数据收集器。

- 数据库服务器上的 I/O 吞吐量
- 数据库服务器上每秒进行输入/输出操作的次数 (IOPS)
- 操作系统服务器使用的 CPU 数量
- 操作系统服务器上的内存使用情况
- 活跃的数据库和 OS 服务器连接数

DMS Fleet Advisor 使用上述指标来生成准确的目标建议，因此目标数据库可以满足性能需求。这样可以防止在维护资源时产生额外费用 AWS Cloud。

## AWS DMS Fleet Advisor 如何收集容量和资源利用率指标？

DMS Fleet Advisor 每分钟收集一次性能指标。

对于 Oracle 和 SQL Server，DMS Fleet Advisor 运行 SQL 查询以捕获每个数据库指标的值。

对于 MySQL 和 PostgreSQL，DMS Fleet Advisor 从运行数据库的操作系统服务器收集性能指标。在 Windows 中，DMS Fleet Advisor 运行 WMI Query Language (WQL) 脚本并接收 WMI 数据。在 Linux 中，DMS Fleet Advisor 运行捕获操作系统服务器指标的命令。

### Important

运行远程 SQL 脚本可能会影响生产数据库的性能。但是，数据收集查询不包含任何计算逻辑。因此，数据收集进程消耗的数据库资源不太可能超过 1%。

您可以查看数据收集器为收集指标而运行的所有查询。为此，请打开 `DMSCollector.Collections.json` 文件。您可以在 `etc` 文件夹中找到此文件，该文件夹位于数据收集器的安装文件夹中。默认路径为 `C:\ProgramData\Amazon\AWS DMS Collector\etc\DMSCollector.Collections.json`。

DMS 数据收集器使用本地文件系统作为收集的所有数据的临时存储空间。DMS 数据收集器以 JSON 格式存储收集的数据。您可以在离线模式下使用本地收集器，并在配置数据转发之前手动检查或验证收集的文件。您可以在 `out` 文件夹中看到收集的所有文件，该文件夹位于 DMS 数据收集器的安装文件夹中。默认路径为 `C:\ProgramData\Amazon\AWS DMS Collector\out`。

### Important

如果您在离线模式下运行 DMS 数据收集器，并将收集的数据存储在服务器上超过 14 天，则无法使用 Amazon CloudWatch 来显示这些指标。但是，DMS Fleet Advisor 仍使用这些数据来生成建议。有关 CloudWatch 图表的更多信息，请参阅[建议详细信息](#)。

您还可以在在线模式下检查或验证收集的数据文件。DMS 数据收集器会将所有数据转发到您在 DMS 数据收集器设置中指定的 Amazon S3 存储桶。

您可以使用 DMS 数据收集器从本地数据库收集数据。此外，还可以从 Amazon RDS 和 Aurora 数据库收集数据。但是，由于 Amazon RDS 或 Aurora 与本地数据库实例之间的差异，无法在云中成功运行所有 DMS 数据收集器查询。由于 DMS 数据收集器从主机操作系统收集 MySQL 和 PostgreSQL 数据库的利用率指标，因此这种方法不适用于 Amazon RDS 和 Aurora。

## DMS 数据收集器故障排除

在以下列表中，您可以找到在使用数据收集器收集数据期间遇到具体问题时可以采取的操作。

## 主题

- [与网络和服务器连接相关的数据收集问题](#)
- [与 Windows Management Instrumentation 相关的数据收集问题](#)
- [与 Windows 网页编辑器相关的数据收集问题](#)
- [与 SSL 相关的数据收集问题](#)

## 与网络和服务器连接相关的数据收集问题

NET : ping 请求期间发生异常。

检查计算机的名称，看看计算机是否处于无法解析为 IP 地址的状态。

例如，检查计算机是否已关闭、与网络断开连接或已停用。

NET : 超时

打开入站防火墙规则“文件和打印机共享 ( 回显请求 – ICMPv4-In )”。例如：

\* Inbound ICMPv4

净值：DestinationHostUnreachable

检查计算机的 IP 地址。具体而言，检查计算机是否与运行 DMS 数据收集器的计算机位于同一子网中，以及是否响应地址解析协议 (ARP) 请求。

如果计算机位于不同子网中，则无法将网关的 IP 地址解析为媒体访问控制 (MAC) 地址。

此外，检查计算机是否已关闭、与网络断开连接或已停用。

## 与 Windows Management Instrumentation 相关的数据收集问题

WMI : RPC 服务器不可用。( 来自 HRESULT 的异常 : 0x800706BA )

打开入站防火墙规则“Windows Management Instrumentation (DCOM-In)”。例如：

\* Inbound TCP/IP at local port 135.

此外，打开入站防火墙规则“Windows Management Instrumentation (WMI-In)”。例如：

\* Inbound TCP/IP at local port 49152 - 65535 , 适用于 Windows Server 2008 及更高版本。



\* Inbound TCP/IP at local port 1025 - 5000，适用于 Windows Server 2003 和更低版本。

WMI：访问被拒绝。（来自 HRESULT 的异常：0x80070005 ( E\_ACCESSDENIED ) )

尝试以下操作：

- 将 DMS 数据收集器用户添加到 Windows 组、分布式 COM 用户或管理员。
- 启动 Windows Management Instrumentation 服务，并将其启动类型设置为“自动”。
- 确保 DMS 数据收集器用户名采用了 \ 格式。

WMI：访问被拒绝

在根 WMI 命名空间上，向 DMS 数据收集器用户添加“远程启用”权限。

使用“高级”设置，并确保权限适用于“此命名空间和子命名空间”。

WMI：消息筛选器取消了调用。（来自 HRESULT 的异常：0x80010002...）

重新启动 Windows Management Instrumentation 服务。

## 与 Windows 网页编辑器相关的数据收集问题

WPC：找不到网络路径

打开入站防火墙规则“文件和打印机共享 (SMB-In)”。例如：

\* Inbound TCP/IP at local port 445.

此外，启动远程注册表服务，并将其启动类型设置为“自动”。

WPC：访问被拒绝

将 DMS 数据收集器用户添加到性能监视器用户或管理员组。

WPC：类别不存在

运行 loader /r 重建性能计数器缓存，然后重新启动计算机。

### Note

有关在使用 AWS Database Migration Service (AWS DMS) 迁移数据时所遇到问题的故障排除信息，请参阅[故障排除和诊断支持](#)。

## 与 SSL 相关的数据收集问题

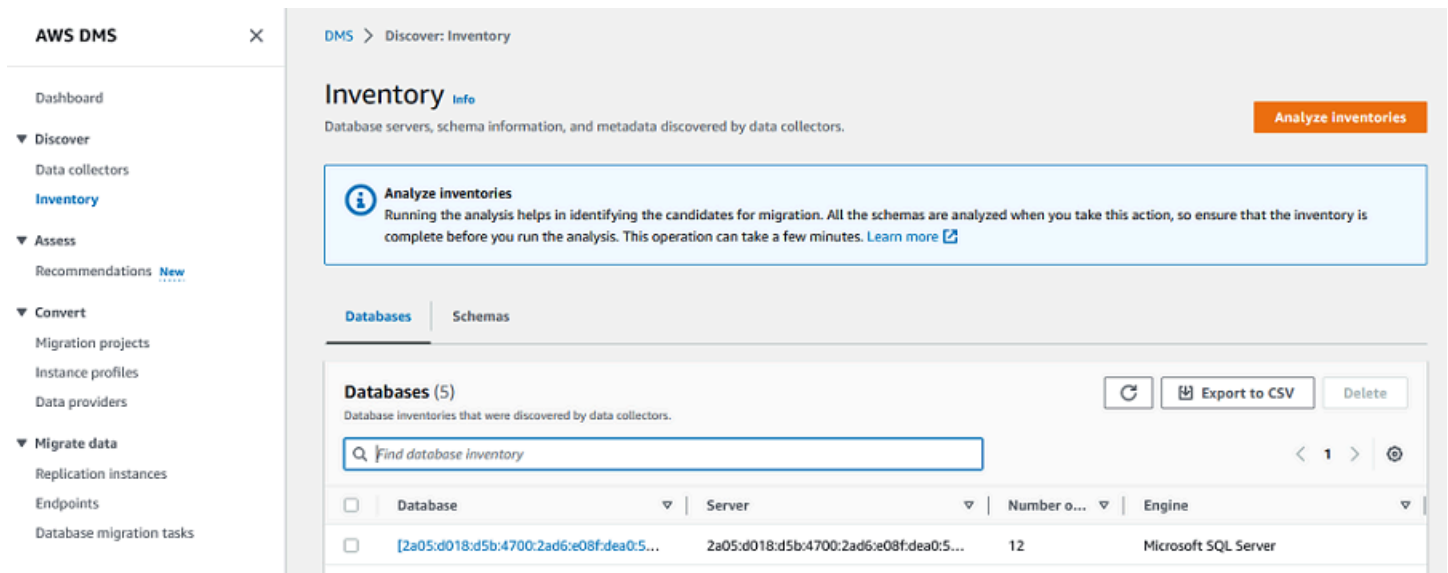
### SSL 错误

数据库需要安全的 SSL 连接，但您尚未为连接打开验证 CA 和使用 SSL 选项。打开这些选项，并确保本地操作系统安装了数据库所使用的证书颁发机构。有关更多信息，请参阅 [设置 SSL](#)。

## 在 AWS DMS Fleet Advisor 中使用清单进行分析

要检查潜在数据库迁移的可行性，您可以使用发现的数据库和架构清单。您可以使用清单中的信息，了解哪些数据库和架构适合迁移。

您可以在控制台上访问数据库和架构清单。为此，在控制台上选择清单。



DMS Fleet Advisor 会分析数据库架构，以确定不同架构的相似性。此分析并不会比较对象的实际代码。DMS Fleet Advisor 仅比较架构对象（例如函数和过程）的名称，以识别不同数据库架构中的相似对象。

### 主题

- [使用数据库清单进行分析](#)
- [使用架构清单进行分析](#)

## 使用数据库清单进行分析

要查看网络中发现的从中收集数据的所有服务器上的所有数据库列表，请使用以下步骤。

## 查看从中收集数据的网络服务器上的数据库列表

1. 在控制台上选择清单。

将打开清单页面。

2. 选择数据库选项卡。

将显示发现的数据库列表。

The screenshot shows the 'Inventory' page in the AWS console. It features a header with 'Inventory Info' and a description: 'Database servers, schema information, and metadata discovered by data collectors.' There is an 'Analyze inventories' button in the top right. Below the header is a blue box with an information icon and text: 'Analyze inventories. Running the analysis helps in identifying the candidates for migration. All the schemas are analyzed when you take this action, so ensure that the inventory is complete before you run the analysis. This operation can take a few minutes. Learn more'.

Below this is a navigation bar with 'Databases' and 'Schemas' tabs. The 'Databases' tab is active, showing a list of 7 database inventories. The list has a search bar and pagination controls. The table below shows the following data:

Database	Server	Number of s...	Engine	Engine version	Engine ...
WinServ2016.d...	-	No data	PostgreSQL	-	-
VM-MSSQL14-...	10.11.1.10	44	Microsoft SQL ...	2014 (Extended support)	Enterprise
MSSQL01.dbm...	-	No data	Microsoft SQL ...	2019 (Mainstream support)	Express

3. 选择分析清单以确定架构属性，例如相似性和复杂性。该过程所需的时间取决于要分析的对象数量，但不会超过一小时。分析结果可在清单页面的架构选项卡上找到。

DMS Fleet Advisor 会分析所有已发现数据库中的架构，以定义其对象的交叉点。分析结果以百分比表示。DMS Fleet Advisor 将交集超过 50% 的架构视为重复架构。原始架构被标识为找到重复项的架构。这有助于识别要先转换或迁移的原始架构。

对整个清单进行统一分析，以识别重复的架构。

## 使用架构清单进行分析

您可以查看网络中从中收集数据的服务器上发现的数据库架构列表。请执行以下过程。

## 查看从中收集数据的网络服务器上的架构列表

1. 在控制台上选择清单。将打开清单页面。
2. 选择架构选项卡。将显示架构列表。
3. 在列表中选择架构以查看有关信息，包括服务器、数据库、大小和复杂性。

对于每个架构，可以查看对象摘要，其中提供了有关对象类型、对象数量、对象大小和代码行的信息。

4. (可选) 选择分析清单以识别重复的架构。DMS Fleet Advisor 分析数据库架构以定义其对象的交集。
5. 可以将清单信息导出为 .csv 文件，以便进一步查看。

The screenshot shows the 'Schema inventory' page in the AWS DMS console. The page title is 'Inventory' and it includes a sub-header 'Database servers, schema information, and metadata discovered by data collectors.' There is an 'Analyze inventories' button. Below this, there is a section for 'Schemas (13)' with a search bar and an 'Export to CSV' button. The main content is a table of schemas.

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

要识别要迁移的架构并确定迁移目标，可以使用 AWS Schema Conversion Tool (AWS SCT) 或 DMS 架构转换。有关更多信息，请参阅[在 AWS SCT 中使用新项目向导](#)。

确定要迁移的架构后，可以使用 AWS SCT 或 DMS 架构转换来转换架构。有关 DMS 架构转换的更多信息，请参阅[使用 DMS 架构转换来转换数据库架构](#)。

## 使用 AWS DMS Fleet Advisor 目标建议功能

要探究并选择最佳迁移目标，您可以在 DMS Fleet Advisor 中为本地源数据库生成目标建议。建议包括一个或多个可能的 AWS 目标引擎，您可以选择用于本地源数据库的迁移。根据这些可能的目标引擎，DMS Fleet Advisor 建议将单个目标引擎作为大小合适的迁移目的地，并将该目标指示为 DMS 建议的目标。为了确定这个大小合适的迁移目标，DMS Fleet Advisor 将使用数据收集器收集的清单元数据和指标。

在开始迁移之前，您可以使用建议来发现迁移选项、节省成本并降低风险。可以将建议导出为逗号分隔值 (CSV) 文件，并与主要利益相关者共享以协助做出决策。您可以将建议导出到 AWS Pricing Calculator，以进一步优化维护成本。有关更多信息，请参阅 <https://calculator.aws/#/>。

您无法在 DMS Fleet Advisor 中修改目标建议。因此，不能使用 DMS Fleet Advisor 进行假设分析。假设分析是一个更改目标参数以观察此变更对建议的定价估算有何影响的过程。您可以在 AWS Pricing Calculator 中运行假设分析，方法是使用建议的目标参数作为 AWS Pricing Calculator 的起点。有关更多信息，请参阅 <https://calculator.aws/#/>。

我们建议您考虑将 DMS Fleet Advisor 建议作为迁移计划的起点。然后，您可以决定更改建议的实例参数，以优化数据库工作负载的成本或性能。

### 主题

- [建议的目标实例](#)
- [DMS Fleet Advisor 如何确定建议的目标实例规格？](#)
- [使用 AWS DMS Fleet Advisor 生成目标建议](#)
- [使用 AWS DMS Fleet Advisor 探究目标建议的详细信息](#)
- [使用 AWS DMS Fleet Advisor 导出目标建议](#)
- [使用 AWS DMS Fleet Advisor 发现和分析迁移限制](#)
- [目标建议故障排除](#)

## 建议的目标实例

对于目标建议，DMS Fleet Advisor 考虑以下通用型、内存优化型和可突增性能 Amazon RDS 数据库实例。

- db.m5
- db.m6i
- db.r5

- db.r6i
- db.t3
- db.x1
- db.x1e
- db.z1d

有关 Amazon RDS 数据库实例类的更多信息，请参阅《Amazon RDS 用户指南》中的[数据库实例类](#)。

## DMS Fleet Advisor 如何确定建议的目标实例规格？

DMS Fleet Advisor 可以根据数据库容量或利用率生成建议。

- 如果选择根据数据库容量生成建议，则 DMS Fleet Advisor 会将现有数据库容量映射到最接近的实例类的规格。
- 如果选择根据资源利用率生成建议，则 DMS Fleet Advisor 会确定 CPU、内存、IO 吞吐量和 IOPS 等指标的第 95 个百分位数值。第 95 个百分位数表示收集的数据中有 95% 低于该值。然后，DMS Fleet Advisor 将这些值映射到最接近的实例类的规格。

为了确定目标数据库的大小，DMS Fleet Advisor 会收集有关源数据库大小的信息。然后，DMS Fleet Advisor 建议对目标存储使用相同的大小。如果源数据库存储过度预置，则目标存储的建议大小也将过度预置。

如果要使用 AWS DMS 迁移数据，那么可能需要提高目标数据库实例的 IOPS 预置。DMS Fleet Advisor 在生成目标建议时，该服务仅考虑源数据库指标。DMS Fleet Advisor 不会考虑运行数据迁移任务可能需要的额外 IOPS。有关更多信息，请参见[迁移任务运行缓慢](#)。

为了估计 IOPS 成本，DMS Fleet Advisor 使用您的源 IOPS 使用情况 one-to-one 映射作为基准。DMS Fleet Advisor 将峰值负载视为基准值，并将 IOPS 定价利用率视为 100%。

对于 PostgreSQL 和 MySQL 源数据库，DMS Fleet Advisor 可能会在目标建议中包括 Aurora 和 Amazon RDS 数据库实例。如果 Aurora 配置映射到源需求，则 DMS Fleet Advisor 会将此选项标记为建议选项。

## 使用 AWS DMS Fleet Advisor 生成目标建议

完成数据库和分析实例集的数据收集和清单后，您可以在 DMS Fleet Advisor 中生成目标建议。为此，请选择源数据库，并配置 DMS Fleet Advisor 目标建议功能用于确定目标实例大小的设置。此外，DMS Fleet Advisor 目标建议功能使用从源数据库收集的容量和利用率指标。

## 生成目标建议

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。

确保选择了在其中使用 DMS Fleet Advisor 的 AWS 区域。

2. 在导航窗格中，在评测下选择建议，然后选择生成建议。
3. 在选择源数据库面板中，选中要迁移到 AWS Cloud 的数据库名称对应的复选框。

对于搜索源数据库，输入数据库名称以筛选清单。

DMS Fleet Advisor 一次最多可以为 100 个数据库生成建议。

4. 对于可用性和持久性，选择首选的部署选项。

要计算生产数据库的目标建议，请选择生产（多可用区）。DMS Fleet Advisor 将在目标建议中包括位于不同可用区的两个数据库实例。这个多可用区部署选项可提供高可用性、数据冗余和故障转移支持。

如果 Aurora 是推荐的目标引擎，而可用性和持久性是多可用区部署，则目标建议包括读取器和写入器数据库实例。

要计算用于开发或测试的数据库的目标建议，请选择开发/测试（单可用区）。DMS Fleet Advisor 将在您的目标建议中包含单个数据库实例。这个单可用区部署选项可降低维护成本。

5. 对于目标实例大小，选择 DMS Fleet Advisor 用于计算目标建议的首选选项。

要根据源数据库或操作系统服务器配置计算目标建议，请选择总容量。DMS Fleet Advisor 使用源数据库或操作系统服务器的总 CPU、内存和磁盘容量等指标，生成目标建议。然后，DMS Fleet Advisor 将数据库容量指标映射到最接近的 Amazon RDS 数据库实例类的规格。

要根据源数据库或操作系统服务器的实际利用率计算目标建议，请选择资源利用率。DMS Fleet Advisor 使用源数据库或操作系统服务器的 CPU、内存和磁盘容量的利用率指标，生成目标建议。根据利用率指标，DMS Fleet Advisor 计算每个指标的第 95 个百分位数。第 95 个百分位数表示该时间段内 95% 的数据低于此值。然后，DMS Fleet Advisor 将这些值映射到最接近的 Amazon RDS 数据库实例类。

我们建议您使用资源利用率选项，以获得更准确的建议。为此，请确保收集了总容量和资源利用率指标。

6. 选择生成。

DMS Fleet Advisor 会为选定的数据库生成目标建议。对于成功生成的建议，DMS Fleet Advisor 将状态设置为已计算。此外，DMS Fleet Advisor 还使用 AWS Pricing Calculator，以确定建议的目标数据库实例的每月估计成本。现在，您可以详细浏览生成的建议。有关更多信息，请参见 [建议详细信息](#)。

要估算数据清单的每月总成本，请选中计划迁移到云的数据库的复选框。DMS Fleet Advisor 会显示 AWS Cloud 中目标数据库的每月估计总成本和摘要。DMS Fleet Advisor 使用 AWS 价目表 查询 API 提供定价详情，仅供您参考之用。实际费用取决于各种因素，包括 AWS 服务的实际使用情况。有关 AWS 服务定价的更多信息，请参阅 [云服务定价](#)。

## 使用 AWS DMS Fleet Advisor 探究目标建议的详细信息

在 DMS Fleet Advisor 生成目标建议后，您可以在建议表中查看建议的迁移目标的关键参数。这些关键参数包括目标引擎、实例类、虚拟 CPU 数量、内存、存储和存储类型。除了这些参数，DMS Fleet Advisor 还会显示此建议迁移目标的每月估计成本。

每个建议中可能包括一个或多个可能的 AWS 目标引擎。如果建议中包括多个目标引擎，AWS DMS 会将其中一个标记为建议的目标引擎。此外，AWS DMS 会在建议表中显示此建议选项的参数和每月估计成本。

要对目标建议与源数据库的利用率和容量进行比较，可详细探究建议。此外，您可以查看所选建议的迁移限制。这些限制包括不支持的数据库功能、操作项和其他迁移注意事项。

### 详细探究建议

1. 使用 DMS Fleet Advisor 生成目标建议。有关更多信息，请参见 [生成目标建议](#)。
2. 从建议表中选择建议名称。将打开建议页面。
3. 如果建议中包括多个目标选项，那么在目标建议中选择目标选项。
4. 展开源利用率和容量部分。DMS Fleet Advisor 将显示以下指标的资源利用率图表。
  - CPU 数
  - 内存
  - I/O 吞吐量
  - 每秒进行输入/输出操作的次数 (IOPS)
  - 存储
  - 活跃的数据库服务器连接数

可使用这些图表，对来自 DMS 数据收集器的源数据库指标与所选目标引擎的指标进行比较。



如果您在展开“来源利用率和容量”部分后看不到图表，请确保您已授予您的 IAM 用户查看 Amazon CloudWatch 控制面板的权限。有关更多信息，请参阅[亚马逊 CloudWatch 用户指南中的使用亚马逊 CloudWatch 控制面板](#)。

5. 选择包含所选目标引擎名称的链接。将打开目标详细信息页面。
6. 要将目标推荐导出为 CSV，请从“操作”下拉列表中选择“导出到 CSV”选项。
7. 要将目标推荐导出到 AWS Pricing Calculator，请从“操作”下拉列表中选择“优化成本” AWS Pricing Calculator 选项。
8. 在配置部分中，对源数据库参数的值与目标引擎参数进行比较。对于目标引擎，DMS Fleet Advisor 会显示云资源的每月估计成本。DMS Fleet Advisor 使用 AWS 价目表 查询 API 提供定价详情，仅供您参考之用。实际费用取决于各种因素，包括 AWS 服务的实际使用情况。有关 AWS 服务定价的更多信息，请参阅 <https://aws.amazon.com/pricing/> 云服务定价。
9. 在迁移限制部分中，查看迁移限制。我们建议您在将源数据库迁移到 AWS Cloud 时考虑这些限制。

## 使用 AWS DMS Fleet Advisor 导出目标建议

生成目标建议后，可以将建议列表的副本保存为逗号分隔值 (CSV) 文件。

### 生成目标建议

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。

确保选择了在其中使用 DMS Fleet Advisor 的 AWS 区域。

2. 在导航窗格中，在评测下选择建议，然后选择要包含在 CSV 文件中的建议。
3. 选择导出为 CSV，输入文件名并在电脑上选择要保存此文件的文件夹。
4. 打开 CSV 文件。

建议的 CSV 文件包含以下信息。

- CreatedDate— DMS Fleet Advisor 创建目标引擎建议的日期。
- Databaseld— DMS Fleet Advisor 为其创建此建议的源数据库的标识符。
- DeploymentOption— 推荐的 Amazon RDS 数据库实例的部署选项。
- EngineEdition— 推荐的目标亚马逊 RDS 引擎版本。

- EngineName— 目标引擎的名称。
- InstanceMemory— 推荐的 Amazon RDS 数据库实例上的内存量。
- InstanceSizingType— 目标实例的大小。
- InstanceType— 推荐的目标 Amazon RDS 实例类型。
- InstanceVcpu— 推荐的 Amazon RDS 数据库实例上的虚拟 CPU 数量。
- Preferred – 布尔标记，表明建议的是此目标选项。
- Status – 目标引擎建议的状态。
- StorageIops— 在推荐的 Amazon RDS 数据库实例上每秒完成的 I/O 操作数 (IOPS)。
- StorageSize— 推荐的 Amazon RDS 数据库实例的存储大小。
- StorageType— 推荐的 Amazon RDS 数据库实例的存储类型。
- WorkloadType— 目标引擎的部署选项，例如多可用区或单可用区部署。

## 使用 AWS DMS Fleet Advisor 发现和分析迁移限制

您可以使用 DMS 数据收集器来发现目标引擎不支持的数据库功能。为了选择正确的迁移目标，您应当考虑这些限制。

DMS 数据收集器会发现特定的源数据库功能。然后，DMS Fleet Advisor 会从迁移到指定目标的角度分析源功能，并提供有关限制的更多信息，并包括解决或避免此限制的建议措施。此外，DMS Fleet Advisor 还会计算这些限制的影响。

限制列表可在目标引擎详细信息页面中找到。从左侧导航菜单的“推荐”页面导航到此页面。从目标列表中选择要检查的目标引擎。限制列表位于页面底部。

下表包括 Amazon RDS for MySQL 不支持的 MySQL 数据库功能。

限制	描述	影响
身份验证插件	Amazon RDS 不支持 MySQL 身份验证插件。	低
将错误记录到系统日志	Amazon RDS 不支持将错误日志写入系统日志。	低
全局事务标识符	您可以将全局事务标识符与所有 RDS for MySQL 5.7	低

限制	描述	影响
	版、RDS for MySQL 8.0.26 版及更高的 MySQL 8.0 版配合使用。	
组复制	Amazon RDS 不支持 MySQL 组复制插件。	低
InnoDB 表空间加密	Amazon RDS 不支持 InnoDB 表空间加密。	低
InnoDB 保留字	InnoDB 是 Amazon RDS for MySQL 的保留字。您不能将此名称用于 MySQL 数据库。	低
钥匙环插件	Amazon RDS 不支持 MySQL 密钥环插件。	低
密码验证插件	Amazon RDS 不支持 MySQL <code>validate_password</code> 插件。	低
永久保存的系统变量	Amazon RDS 不支持 MySQL 永久保存的系统变量。	低
限制访问	Amazon RDS 限制了对需要高级权限的某些系统程序和表的访问。此外，Amazon RDS 不允许使用 Telnet、Secure Shell ( SSH ) 或 Windows 远程桌面连接直接从主机访问数据库实例。	低
Rewriter 查询重写插件	Amazon RDS 不支持 MySQL Rewriter 查询重写插件。	低
半同步复制	Amazon RDS 不支持 MySQL 半同步复制。	低

限制	描述	影响
可传输的表空间	Amazon RDS 不支持 MySQL 可传输的表空间功能。	低
X 插件	Amazon RDS 不支持 MySQL X 插件。	低

下表包括 Amazon RDS for Oracle 不支持的 Oracle 数据库功能。

限制	描述	影响
Active Data Guard	您不能将 Active Data Guard 与 Oracle 多租户容器数据库 (CDB) 配合使用。	中
Automatic Storage Management	Amazon RDS 不支持 Oracle Automatic Storage Management (Oracle ASM)。	中
Database Activity Streams	对于单租户架构，Amazon RDS 不支持 Oracle Database Activity Streams。	高
文件大小限制	RDS for Oracle 数据库实例上单个文件的最大大小为 16 TiB。	中
FTP 和 SFTP	Amazon RDS 不支持 FTP 和 SFTP。	中
混合分区表	Amazon RDS 不支持 Oracle 混合分区表。	高
Oracle Data Guard	对于单租户架构，Amazon RDS 不支持 Oracle Data Guard。	高

限制	描述	影响
Oracle Database Vault	Amazon RDS 不支持 Oracle Database Vault。	高
Oracle DBA 权限库	对于 Oracle DBA 权限，Amazon RDS 存在限制。有关更多信息，请参阅 <a href="#">Oracle DBA 权限限制</a> 。	高
Oracle Enterprise Manager	对于单租户架构，Amazon RDS 不支持 Oracle Enterprise Manager。	高
Oracle Enterprise Manager Agent	对于单租户架构，Amazon RDS 不支持 Oracle Enterprise Manager Agent。	中
Oracle Enterprise Manager 云控制管理存储库	不能将 Amazon RDS for Oracle 数据库实例用于 Oracle Enterprise Manager 云控制管理存储库。	高
Oracle Flashback Database	Amazon RDS 不支持 Oracle Flashback Database 功能。	高
Oracle Label Security	对于单租户架构，Amazon RDS 不支持 Oracle Label Security。Oracle Label Security 只能与多租户容器数据库 (Oracle CDB) 配合使用。	高
Oracle Messaging Gateway	Amazon RDS 不支持 Oracle Messaging Gateway。	高
Oracle Real Application Clusters	Amazon RDS 不支持 Oracle Real Application Clusters (Oracle RAC)。	高

限制	描述	影响
Oracle Real Application Testing	Amazon RDS 不支持 Oracle Real Application Testing。	高
Oracle Snapshot Standby 数据库	Amazon RDS 不支持 Oracle Snapshot Standby 数据库。	高
公共同义词	对于 Oracle 提供的架构，Amazon RDS 不支持公共同义词。	中
不支持的功能的架构	Amazon RDS 不支持需要系统权限的 Oracle 功能和组件的架构。	高
纯统一审计	Amazon RDS 不支持纯统一审计。您可以在混合模式下使用统一审计。	中
Workspace Manager	Amazon RDS 不支持 Oracle Database Workspace Manager WMSYS 架构。	高

下表包括 Amazon RDS for PostgreSQL 不支持的 PostgreSQL 数据库功能。

限制	描述	影响
并发连接	与 RDS for PostgreSQL 实例之间的最大并发连接数受 <code>max_connections</code> 参数的限制。	低
最新版本	Amazon RDS 不会自动应用主要版本升级。要执行主要版本升级，请手动修改数据库实例。有关更多信息，请参阅 <a href="#">为</a>	低

限制	描述	影响
	<a href="#">PostgreSQL 选择主要版本升级</a> 。	
预留连接	Amazon RDS 保留了多达 3 个供系统维护使用的连接。如果为用户连接参数指定一个值，请为您希望使用的连接数目加上 3。	低
支持的扩展	RDS for PostgreSQL 支持有限数量的 PostgreSQL 数据库引擎扩展。您可以在 PostgreSQL 版本的默认数据库参数组中找到支持的扩展列表。您还可以通过显示 <code>rds.extensions</code> 参数，使用 <code>psql</code> 查看当前扩展列表。	低
表空间拆分或隔离	不能使用表空间进行 I/O 拆分或隔离。在 RDS for PostgreSQL 中，所有存储都在单个逻辑卷上。	低

下表包括 Amazon RDS for SQL Server 不支持的 SQL Server 数据库功能。

限制	描述	影响
备份到 Microsoft Azure Blob Storage	RDS for SQL Server 不支持备份到 Microsoft Azure Blob Storage。	中
缓冲池扩展	RDS for SQL Server 不支持缓冲池扩展。	高

限制	描述	影响
自定义密码策略	RDS for SQL Server 不支持自定义密码策略。	中
数据质量服务	RDS for SQL Server 不支持 SQL Server 数据质量服务 (DQS)。	高
数据库日志传输	RDS for SQL Server 不支持数据库日志传输。	高
数据库名称	数据库名称有以下限制：不能以 rdsadmin 开头，不能以空格或制表符开头或结尾，不能包含创建新行的任何字符，不能包含单引号 (')。	中
数据库快照	RDS for SQL Server 不支持数据库快照。您只能在 Amazon RDS 中使用数据库实例快照。	中
扩展的存储过程	RDS for SQL Server 不支持扩展的存储过程，包括 xp_cmdshell 。	高
文件表	RDS for SQL Server 不支持文件表。	中
FILESTREAM 支持	RDS for SQL Server 不提供 FILESTREAM 支持。	中
链接的服务器	对于链接的服务器，RDS for SQL Server 提供有限的支持。	高
机器学习和 R 服务	RDS for SQL Server 不支持机器学习和 R 服务，因为需要操作系统访问权限才能安装这些服务。	高



限制	描述	影响
维护计划	RDS for SQL Server 不支持维护计划。	高
性能数据收集器	RDS for SQL Server 不支持性能数据收集器。	高
基于策略的管理	RDS for SQL Server 不支持基于策略的管理。	中
PolyBase	适用于 SQL Server 的 RDS 不支持 PolyBase。	高
复制	RDS for SQL Server 不支持复制。	中
资源管理器	RDS for SQL Server 不支持资源管理器。	高
服务器级别触发器	RDS for SQL Server 不支持服务器级别触发器。	中
服务代理端点	RDS for SQL Server 不支持服务代理端点。	高
SSAS	考虑在 RDS for SQL Server 上运行 SQL Server Analysis Services (SSAS) 所适用的限制。有关更多信息，请参阅 <a href="#">限制</a> 。	低
SSIS	考虑在 RDS for SQL Server 上运行 SQL Server Integration Services (SSIS) 所适用的限制。有关更多信息，请参阅 <a href="#">限制</a> 。	低

限制	描述	影响
SSRS	考虑在 RDS for SQL Server 上运行 SQL Server Reporting Services (SSRS) 所适用的限制。有关更多信息，请参阅 <a href="#">限制</a> 。	低
SQL Server 数据库实例的存储大小	SQL Server 通用型 (SSD) 存储和预调配 IOPS 存储实例的最大存储大小为 16 TiB。  SQL Server 磁性存储实例的最大存储大小为 1 TiB。	高
拉伸数据库	RDS for SQL Server 不支持 SQL Server 拉伸数据库功能。	中
T-SQL 端点	RDS for SQL Server 不支持使用 CREATE_ENDPOINT 的所有操作。	高
TRUSTWORTHY 数据库属性	RDS for SQL Server 不支持 TRUSTWORTHY 数据库属性，因为该属性需要 sysadmin 角色。	中

下表列出了建议问题。DMS Fleet Advisor 分析源数据库和目标数据库功能，并提供这些迁移限制。Blocker 影响的限制意味着 DMS Fleet Advisor 无法为源数据库生成目标建议。

限制	描述	影响
未找到相应的实例	AWS DMS找不到可以作为源数据库指标组合大小合适的迁移目标的目标实例。	Blocker

限制	描述	影响
IOPS 找不到相应的实例	源数据库使用的 IOPS 数量超过了可能的目标数据库实例的最大 IOPS 数。	Blocker
RAM 找不到相应的实例	源数据库使用了 GB 的 RAM，这超过了可能的目标数据库实例的最大 RAM 大小。	Blocker
未按存储大小找到合适的实例	源数据库使用了 TB 的存储空间，超过了可能的目标数据库实例的最大存储大小。	Blocker
按版本找不到相应的实例	源数据库有一个版本，Amazon RDS 不支持该版本。	Blocker
CPU 内核未找到相应的实例	源数据库有许多 CPU 内核，超过了可能的目标数据库实例的最大 CPU 内核数。	Blocker
按版本找不到相应的实例	您的源数据库有版本，AWS DMS 但无法识别。	Blocker
CPU 参数未定义	DMS 数据收集器未收集有关您的源数据库使用的 CPU 的信息。确保在数据收集器中收集了数据转发所需的指标并配置了凭据。请参阅 <a href="#">配置数据转发凭据</a> 。	Blocker
内存参数未定义	DMS 数据收集器未收集有关您的源数据库使用的内存的信息。确保在数据收集器中收集了数据转发所需的指标并配置了凭据。请参阅 <a href="#">配置数据转发凭据</a> 。	Blocker

限制	描述	影响
存储大小参数未定义	DMS 数据收集器未收集有关您的源数据库使用的存储大小的信息。确保在数据收集器中收集了数据转发所需的指标并配置了凭据。请参阅 <a href="#">配置数据转发凭证</a> 。	Blocker
存储 IOPS 参数未定义	DMS 数据收集器未收集源数据库使用的存储 IOPS 指标。确保在数据收集器中收集了数据转发所需的指标并配置了凭据。	Blocker
数据不足	DMS 数据收集器收集的数据不足以生成目标推荐。确保在数据收集器中配置了用于数据转发的凭据。请参阅 <a href="#">配置数据转发凭证</a> 。	Blocker
数据库版本未定义	DMS 数据收集器未收集有关您的源数据库版本的信息。确保在数据收集器中收集了数据转发所需的指标并配置了凭据。请参阅 <a href="#">配置数据转发凭证</a> 。	Blocker
未知错误	DMS Fleet Advisor 无法为您的源数据库生成目标建议。	Blocker
数据库版本未定义	DMS Fleet Advisor 未收集有关您的源数据库版本的信息。DMS Fleet Advisor 建议您为源数据库使用最新的数据库版本。如果您选择此建议，则必须升级数据库版本。查看为源数据库生成的目标建议，并确保这些建议符合您的要求。	高

限制	描述	影响
增加 RDS 设置中的数据库连接数	您的源数据库需要一定数量的连接。默认情况下，Amazon RDS 数据库实例的可用连接数是不同的。请务必在创建 RDS 数据库实例时更改此默认值。为此，请更新 <code>max_connections</code> 参数组中的参数值。	中
目标版本兼容	源数据库的目标建议使用不同的数据库版本。您的源数据库版本支持与推荐的目标版本相同的功能。但是，选择这个新的数据库版本可能会增加您的开支。	中
存储吞吐量参数未定义	DMS 数据收集器未收集源数据库使用的存储吞吐量指标。查看为源数据库生成的目标建议，并确保这些建议符合您的要求。	中
数据库连接编号参数未定义	DMS 数据收集器未收集有关您的源数据库使用的连接数的信息。查看为源数据库生成的目标建议，并确保这些建议符合您的要求。或者，也可以申请增加配额。	中
数据库降级版本	您的源数据库在比 Amazon RDS 数据库更高的版本上运行。要降级数据库版本，请确保不要使用较低版本中未实现的功能。或者，也可以使用 Amazon EC2 作为迁移目标。	中

限制	描述	影响
目标版本不同	源数据库的目标建议使用不同的数据库版本。您的源数据库版本与推荐的目标版本兼容。但是，推荐的目标数据库版本不支持源数据库版本的某些功能。选择这个新的数据库版本可能会增加您的开支。	中
从不支持的版本升级	<p>您的源数据库已达到支持终止阶段。要使用最新的数据库引擎版本作为目标，请在迁移前升级数据库。或者，也可以使用 Amazon EC2 作为迁移目标。</p> <p>根据数据库引擎的不同，请使用以下链接之一了解更多信息：</p> <p><a href="#">升级 MySQL</a></p> <p><a href="#">升级 SQL 服务器</a></p> <p><a href="#">升级 OracleDB</a></p> <p><a href="#">升级 PostgreSQL</a></p>	中

## 目标建议故障排除

在以下列表中，您可以找到在使用 DMS Fleet Advisor 目标建议功能期间遇到问题时应采取的措施。

### 主题

- [我看不到目标建议的价格估算](#)
- [我看不到资源利用率图表](#)
- [我看不到指标收集状态](#)

## 我看不到目标建议的价格估算

对于状态为成功的建议，如果您看到每月估计费用为无数据，请确保已授予 IAM 用户访问 AWS 价目表服务 API 的权限。为此，必须创建包含 `pricing:GetProducts` 权限的策略，并将其添加到 IAM 用户，如[创建 IAM 资源](#)中所述。

对于状态为失败的建议，DMS Fleet Advisor 不会计算每月估计费用。

## 我看不到资源利用率图表

如果您在展开来源利用率和容量部分后看到“无法加载指标”消息，请确保您已授予您的 IAM 用户查看 Amazon CloudWatch 控制面板的权限。为此，必须将所需策略添加至 IAM 用户，如[创建 IAM 资源](#)中所述。

或者，可以创建包含

`cloudwatch:GetDashboard`、`cloudwatch:ListDashboards`、`cloudwatch:PutDashboard` 和 `cloudwatch>DeleteDashboards` 权限的自定义策略。有关更多信息，请参阅[亚马逊 CloudWatch 用户指南中的使用亚马逊 CloudWatch 控制面板](#)。

## 我看不到指标收集状态

如果您在选择生成建议时，看到指标收集为无可用数据，请确保收集了数据。有关更多信息，请参见[AWS DMS Fleet Advisor 收集数据](#)。

如果您在收集数据后遇到此问题，请务必授予您的 IAM 用户访问亚马逊的 `cloudwatch:Get*` 权限 CloudWatch。DMS Fleet Advisor 使用服务相关角色代表您发布收集的 CloudWatch 数据库性能指标。请务必创建一个与 DMS Fleet Advisor 配合使用的服务相关角色。有关更多信息，请参阅[创建 IAM 资源](#)。

## DMS 舰队顾问的局限性

使用 DMS 舰队顾问时的限制包括以下几点：

- DMS 舰队顾问生成 one-to-one 建议。对于每个源数据库，DMS Fleet Advisor 确定单个目标引擎。DMS Fleet Advisor 不处理多租户服务器，也不提供在单个目标数据库实例上运行多个数据库的建议。
- DMS Fleet Advisor 不提供有关可用数据库版本升级的建议。
- DMS Fleet Advisor 一次最多可以为 100 个数据库生成建议。

- 如果您安装的是 Windows 应用程序 DMS 数据收集器，请确保同时安装 .NET Framework 4.8 和 PowerShell 6.0 及更高版本。有关硬件要求，请参阅[安装数据收集器](#)。
- DMS 数据收集器需要具备在域服务器上使用 LDAP 协议运行请求的权限。
- DMS 数据收集器需要在 Linux 中运行的 sudo SSH 脚本。
- DMS 数据收集器需要权限才能在 Windows 中运行远程 PowerShell、Windows 管理工具 (WMI)、WMI 查询语言 (WQL) 和注册表脚本。
- 对于 MySQL 和 PostgreSQL，DMS Fleet Advisor 无法收集数据库的性能指标。不过，DMS Fleet Advisor 会收集操作系统服务器指标。因此，您无法根据运行于 Amazon RDS 和 Aurora 上的 MySQL 和 PostgreSQL 数据库的使用率指标生成建议。



# 使用 DMS 架构转换来转换数据库架构

AWS Database Migration Service (AWS DMS) 中的 DMS 架构转换使不同类型的数据库之间的数据库迁移更具可预测性。使用 DMS 架构转换来评测源数据提供程序的迁移复杂性，然后用来转换数据库架构和代码对象。然后，您可以将转换后的代码应用于目标数据库。

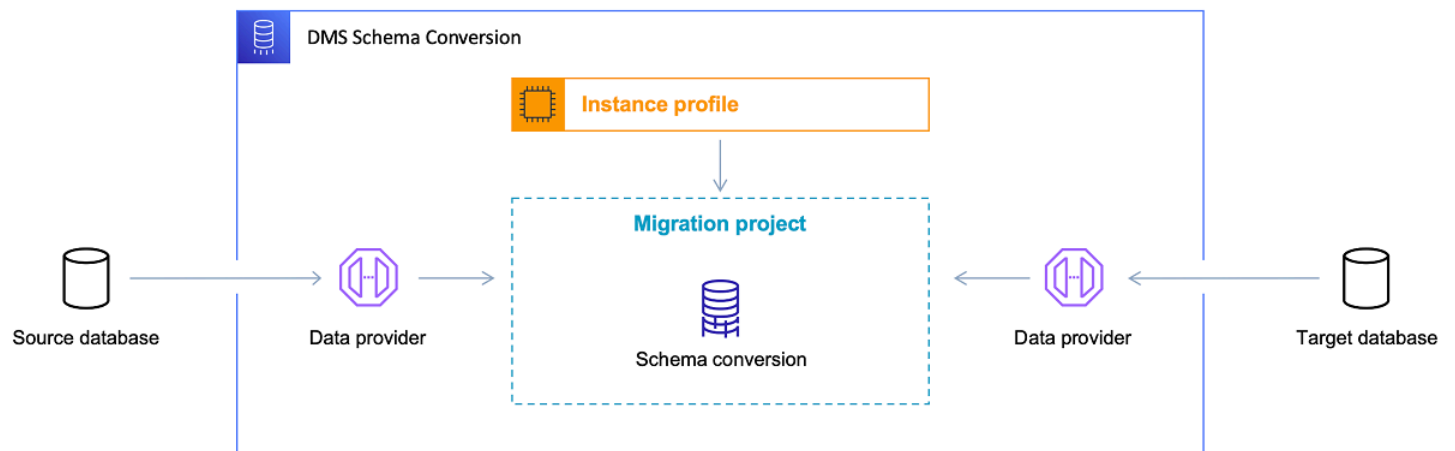
DMS 架构转换会自动将源数据库架构和大部分数据库代码对象转换为与目标数据库兼容的格式。此转换包括表、视图、存储过程、函数、数据类型、同义词等。任何 DMS 架构转换无法自动转换的对象都会被明确标记。要完成迁移，您可以手动转换这些对象。

简而言之，[DMS 架构转换](#)由以下三个组件运行：实例配置文件、数据提供程序和迁移项目。实例配置文件指定网络和安全设置。数据提供程序存储数据库连接凭证。迁移项目包含数据提供程序、实例配置文件和迁移规则。AWS DMS 使用数据提供器和实例配置文件来设计转换数据库架构和代码对象的流程。

有关支持的源数据库的列表，请参阅[DMS 架构转换的源](#)。

有关支持的目标数据库列表，请参阅[DMS 架构转换的目标](#)。

下图阐明了 DMS 架构转换过程。



通过以下主题更好地了解如何使用 DMS 架构转换。

## 主题

- [支持的 AWS 区域](#)
- [架构转换功能](#)
- [架构转换限制](#)
- [开始使用 DMS 架构转换](#)

- [为 DMS 架构转换设置网络](#)
- [在 DMS 架构转换中创建源数据提供程序](#)
- [在 DMS 架构转换中创建目标数据提供程序](#)
- [在 DMS 架构转换中管理迁移项目](#)
- [使用 DMS 架构转换创建数据库迁移评测报告](#)
- [使用 DMS 架构转换](#)
- [在 DMS 架构转换中使用扩展包](#)

## 支持的 AWS 区域

您可以在下面 AWS 区域创建 DMS 架构转换迁移项目。在其他区域，您可以使用 AWS Schema Conversion Tool。有关的更多信息 AWS SCT，请参阅 [AWS Schema Conversion Tool 用户指南](#)。

区域名称	区域
美国东部 ( 弗吉尼亚州北部 )	us-east-1
美国东部 ( 俄亥俄州 )	us-east-2
美国西部 ( 俄勒冈州 )	us-west-2
亚太地区 ( 东京 )	ap-northeast-1
亚太地区 ( 新加坡 )	ap-southeast-1
亚太地区 ( 悉尼 )	ap-southeast-2
欧洲地区 ( 法兰克福 )	eu-central-1
欧洲地区 ( 斯德哥尔摩 )	eu-north-1
欧洲地区 ( 爱尔兰 )	eu-west-1

## 架构转换功能

DMS 架构转换提供以下功能：

- DMS 架构转换会自动管理您的数据库迁移项目所需的 AWS Cloud 资源。这些资源包括实例配置文件、数据提供程序和 AWS Secrets Manager 机密。它们还包括 AWS Identity and Access Management (IAM) 角色、Amazon S3 存储桶和迁移项目。
- 您可以使用 DMS 架构转换来连接到源数据库、读取元数据和创建数据库迁移评测报告。然后，您可以将报告保存到 Amazon S3 存储桶。借助这些报告，您将大致了解您的架构转换任务，以及 DMS 架构转换无法自动转换为目标数据库的项目的详细信息。数据库迁移评测报告有助于评估 DMS 架构转换可以自动执行多少迁移项目。此外，这些报告还有助于估算完成转换所需的手动工作量。有关更多信息，请参阅[使用 DMS 架构转换创建数据库迁移评测报告](#)。
- 连接到源和目标数据提供程序后，DMS 架构转换可以将您的现有源数据库架构转换为目标数据库引擎。您可以从源数据库中选择要转换的任何架构项目。在 DMS 架构转换中转换数据库代码后，您可以查看源代码和转换后的代码。此外，您还可以将转换后的 SQL 代码保存到 Amazon S3 存储桶。
- 在转换源数据库架构之前，您可以设置转换规则。您可以使用转换规则更改列的数据类型、将对象从一个架构移动到另一个架构以及更改对象的名称。您可以将转换规则应用到数据库、架构、表和列。有关更多信息，请参阅[设置转换规则](#)。
- 您可以更改转换设置以提高转换后代码的性能。这些设置特定于每个转换对，并且取决于您在代码中使用的源数据库的功能。有关更多信息，请参阅[指定架构转换设置](#)。
- 在某些情况下，DMS 架构转换无法将源数据库功能转换为等效的 Amazon RDS 功能。对于这些情况，DMS 架构转换会在目标数据库中创建一个扩展包，以模拟未转换的功能。有关更多信息，请参阅[使用扩展包](#)。
- 您可以将转换后的代码和扩展包架构应用于目标数据库。有关更多信息，请参阅[应用转换后的代码](#)。
- DMS 架构转换支持最新 AWS SCT 版本中的所有功能。有关更多信息，请参阅[AWS SCT 的最新发行说明](#)。
- 在 DMS 将转换后的 SQL 代码迁移到目标数据库之前，您可以对其进行编辑。有关更多信息，请参阅[编辑和保存转换后的 SQL 代码](#)。

## 架构转换限制

DMS 架构转换是 () 的 AWS Schema Conversion Tool 网络版本。AWS SCT 与 AWS SCT 桌面应用程序相比，DMS 架构转换支持的数据库平台更少，提供的功能也更有限。要转换数据仓库架构、大数据框架、应用程序 SQL 代码和 ETL 流程，请使用 AWS SCT。有关的更多信息 AWS SCT，请参阅[AWS Schema Conversion Tool 用户指南](#)。

使用 DMS 架构转换进行数据库架构转换时，存在以下限制：

- 您无法保存迁移项目并在离线模式下使用。

- 您无法在 DMS 架构转换的迁移项目中编辑源代码的 SQL 代码。要编辑源数据库的 SQL 代码，请使用常规 SQL 编辑器。选择从数据库刷新，将更新的代码添加到您的迁移项目中。
- DMS 架构转换中的迁移规则不支持更改列排序规则。此外，您不能使用迁移规则将对象移动到新架构。
- 您不能对源数据库树和目标数据库树应用筛选器以仅显示符合筛选子句的数据库对象。
- DMS Schema Conversion 扩展包不包含在转换后的代码中模拟电子邮件发送、作业调度和其他功能的 AWS Lambda 函数。
- DMS 架构转换不使用客户管理的 KMS 密钥来访问任何客户 AWS 资源。例如，DMS 架构转换不支持使用客户管理的 KMS 密钥访问 Amazon S3 中的客户数据。

## 开始使用 DMS 架构转换

要开始使用 DMS 架构转换，可使用以下教程。在本教程中，您可以学习设置 DMS 架构转换、创建迁移项目以及连接到数据提供程序。然后，您可以学习评测迁移的复杂性，并将源数据库转换为与目标数据库兼容的格式。然后，还可以将转换后的代码应用于目标数据库。

以下教程介绍了先决任务，并演示了如何将 Amazon RDS for SQL Server 数据库转换为 Amazon RDS for MySQL。您可以使用任何支持的源数据提供程序和目标数据提供程序。有关更多信息，请参阅 [DMS 架构转换的源数据提供程序](#)。

[有关 DMS 架构转换的更多信息，请阅读 Oracle 到 PostgreSQL 和 SQL Server 到 MySQL step-by-step 迁移的迁移指南。](#)

[本视频](#)介绍了 DMS 架构转换用户界面，并协助您熟悉此服务的核心组件。

### 主题

- [使用 DMS 架构转换的先决条件](#)
- [步骤 1：创建实例配置文件](#)
- [步骤 2：配置数据提供程序](#)
- [步骤 3：创建迁移项目](#)
- [步骤 4：创建评测报告](#)
- [步骤 5：转换源代码](#)
- [步骤 6：应用转换后的代码](#)
- [步骤 7：清理和故障排除](#)

## 使用 DMS 架构转换的先决条件

要设置 DMS 架构转换，请完成以下任务。然后，您可以设置实例配置文件、添加数据提供程序并创建迁移项目。

### 主题

- [基于 Amazon VPC 创建 VPC](#)
- [创建 Amazon S3 存储桶](#)
- [将数据库凭据存储在 AWS Secrets Manager](#)
- [创建 IAM 角色](#)

### 基于 Amazon VPC 创建 VPC

在此步骤中，您将在中创建虚拟私有云 (VPC) AWS 账户。此 VPC 基于亚马逊虚拟私有云 (亚马逊 VPC) 服务，包含您的 AWS 资源。

#### 为 DMS 架构转换创建 VPC

1. 登录 AWS Management Console 并打开亚马逊 VPC 控制台，[网址为 https://console.aws.amazon.com/vpc/](https://console.aws.amazon.com/vpc/)。
2. 选择创建 VPC。
3. 在创建 VPC 页面上，输入以下设置：
  - 要创建的资源 – VPC 等
  - 名称标签自动生成 – 选择自动生成，然后输入一个全局唯一名称。例如，输入 **sc-vpc**。
  - IPv4 CIDR 块 – **10.0.1.0/24**
  - NAT 网关 – 在 1 个可用区中
  - VPC 端点 – 无
4. 保留其余默认设置，然后选择创建 VPC。
5. 选择子网，记下公共子网和专用子网 ID。

要连接到 Amazon RDS 数据库，请使用包含公共子网的子网组。

要连接到本地数据库，请使用包含专用子网的子网组。有关更多信息，请参阅 [步骤 1：创建实例配置文件](#)。

6. 选择 NAT 网关。选择您的 NAT 网关，并记下弹性 IP 地址。

配置您的网络，确保它 AWS DMS 可以从此 NAT 网关的公有 IP 地址访问您的源本地数据库。有关更多信息，请参阅 [使用互联网连接到 VPC](#)。

当您在 Amazon RDS 上创建实例配置文件和目标数据库时，请使用此 VPC。

## 创建 Amazon S3 存储桶

要存储迁移项目的信息，可创建 Amazon S3 存储桶。DMS 架构转换使用此 Amazon S3 存储桶来保存条目，如评测报告、转换后的 SQL 代码、有关数据库架构对象的信息等等。

为 DMS 架构转换创建 Amazon S3 存储桶

1. 登录 AWS Management Console 并打开 Amazon S3 控制台，[网址为 https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/)。
2. 选择创建存储桶。
3. 在创建存储桶页面上，为 S3 存储桶选择一个全局唯一名称。例如，输入 **sc-s3-bucket**。
4. 对于 AWS 区域，选择区域。
5. 对于存储桶版本控制，选择启用。
6. 保留其余默认设置，然后选择创建存储桶。

## 将数据库凭据存储在 AWS Secrets Manager

将您的源数据库和目标数据库凭据存储在中 AWS Secrets Manager。确保将这些秘密复制到您的 AWS 区域。在迁移项目中，DMS 架构转换将使用这些密钥连接到数据库。

将您的数据库凭据存储在 AWS Secrets Manager

1. 登录 AWS Management Console 并打开 AWS Secrets Manager 控制台，[网址为 https://console.aws.amazon.com/secretsmanager/](https://console.aws.amazon.com/secretsmanager/)。
2. 选择 存储新密钥。
3. 将打开选择密钥类型页面。对于密钥类型，选择要存储的数据库凭证类型：
  - 适用于 Amazon RDS 数据库的凭证 – 选择此选项，存储适用于 Amazon RDS 数据库的凭证。对于凭证，输入数据库的凭证。对于数据库，选择您的数据库。
  - 适用于其他数据库的凭证 – 选择此选项，存储适用于 Oracle 或 SQL Server 源数据库的凭证。对于凭证，输入数据库的凭证。

- 其他类型的密钥 – 选择此选项，仅存储用于连接数据库的用户名和密码。选择添加行，以添加两个键值对。请务必使用 **username** 和 **password** 作为密钥名称。对于与这些密钥相关的值，输入数据库的凭证。
4. 对于加密密钥，选择 Secrets Manager 用于加密密 AWS KMS 钥值的密钥。选择下一步。
  5. 在配置密钥页面上，输入描述性的密钥名称。例如，输入 **sc-source-secret** 或 **sc-target-secret**。
  6. 选择复制密钥，然后在 AWS 区域中选择区域。选择下一步。
  7. 在配置旋转页面，选择下一步。
  8. 在审核页面上，审核密钥详细信息，然后选择存储。

要存储源数据库和目标数据库的凭证，请重复上述步骤。

## 创建 IAM 角色

创建要在迁移项目中使用的 AWS Identity and Access Management (IAM) 角色。DMS 架构转换使用这些 IAM 角色，访问 Amazon S3 存储桶以及存储在 AWS Secrets Manager 中的数据库凭证。

创建 IAM 角色以提供对 Amazon S3 存储桶的访问权限

1. 登录 AWS Management Console 并打开 IAM 控制台，[网址为 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/)。
2. 在导航窗格中，选择角色。
3. 选择创建角色。
4. 在选择可信实体页面上，选择 AWS 服务。选择 DMS。
5. 选择下一步。将打开添加权限页面。
6. 在筛选策略中，输入 **S3**。选择亚马逊 S3 FullAccess。
7. 选择下一步。将打开命名、检查并创建页面。
8. 对于角色名称，输入一个描述性名称。例如，输入 **sc-s3-role**。选择创建角色。
9. 在角色页面上，在角色名称中输入 **sc-s3-role**。选择 sc-s3-role。
10. 在 sc-s3-role 页面上，选择信任关系选项卡。选择编辑信任策略。
11. 在编辑信任策略页面上，编辑角色的信任关系，以使用 `schema-conversion.dms.amazonaws.com` 服务主体作为可信实体。
12. 选择更新信任策略。

## 创建提供访问权限的 IAM 角色 AWS Secrets Manager

1. 登录 AWS Management Console 并打开 IAM 控制台，[网址为 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/)。
2. 在导航窗格中，选择角色。
3. 选择创建角色。
4. 在选择可信实体页面上，选择 AWS 服务。选择 DMS。
5. 选择下一步。将打开添加权限页面。
6. 在筛选策略中，输入 **Secret**。选择 SecretsManagerReadWrite。
7. 选择下一步。将打开命名、检查并创建页面。
8. 对于角色名称，输入一个描述性名称。例如，输入 **sc-secrets-manager-role**。选择创建角色。
9. 在角色页面上，在角色名称中输入 **sc-secrets-manager-role**。选择 sc-secrets-manager-role。
10. 在 sc-secrets-manager-role 页面上，选择信任关系选项卡。选择编辑信任策略。
11. 在编辑信任策略页面上，编辑要使用的角色 `schema-conversion.dms.amazonaws.com` 和您的 AWS DMS 区域服务主体作为可信实体的信任关系。该 AWS DMS 区域服务主体采用以下格式。

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

将 *region-name* 替换为您的区域名称，如 `us-east-1`。

以下代码示例显示了 `us-east-1` 区域的主体。

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

以下代码示例显示了用于访问 AWS DMS 架构转换的信任策略。

```
{
  "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. 选择更新信任策略。

## 步骤 1：创建实例配置文件

在创建实例配置文件之前，请为您的实例配置文件配置子网组。有关为 AWS DMS 迁移项目创建子网组的更多信息，请参阅[创建子网组](#)。

您可以根据以下过程所述创建实例配置文件。在此实例配置文件中，可以为 DMS 架构转换项目指定网络和安全设置。

要创建实例配置文件

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择实例配置文件，然后选择创建实例配置文件。
3. 对于名称，为实例配置文件输入唯一名称。例如，输入 **sc-instance**。
4. 对于网络类型，选择 IPv4 以创建仅支持 IPv4 寻址的实例配置文件。要创建支持 IPv4 和 IPv6 寻址的实例配置文件，可选择双堆栈模式。
5. 对于虚拟私有云 (VPC)，选择在先决条件步骤中创建的 VPC。
6. 对于子网组，为实例配置文件选择子网组。要连接到 Amazon RDS 数据库，请使用包含公有子网的子网组。要连接到本地数据库，请使用包含私有子网的子网组。
7. 选择创建实例配置文件。

要创建迁移项目，请使用此实例配置文件。

## 步骤 2：配置数据提供程序

接下来，创建描述源数据库和目标数据库的数据提供程序。对于每个数据提供程序，可以指定数据存储类型和位置信息。您不会将数据库凭证存储在数据提供程序中。

### 为本地源数据库创建数据提供程序

1. 登录并 AWS Management Console 打开 AWS DMS 控制台。
2. 在导航窗格中，选择数据提供程序，然后选择创建数据提供程序。
3. 对于名称，为源数据提供程序输入唯一名称。例如，输入 **sc-source**。
4. 对于引擎类型，为数据提供程序选择数据库引擎类型。
5. 提供源数据库的连接信息。连接参数取决于源数据库引擎。有关更多信息，请参阅 [创建实例提供程序](#)。
6. 对于安全套接字层 (SSL) 模式，选择 SSL 强制执行类型。
7. 选择创建数据提供程序。

### 为目标 Amazon RDS 数据库创建数据提供程序

1. 登录 AWS Management Console 并打开 AWS DMS 控制台。
2. 在导航窗格中，选择数据提供程序，然后选择创建数据提供程序。
3. 对于配置，选择 RDS 数据库实例。
4. 对于来自 RDS 的数据库，选择浏览，然后选择您的数据库。DMS 架构转换会自动检索有关引擎类型、服务器名称和端口的信息。
5. 对于名称，为目标数据提供程序输入唯一名称。例如，输入 **sc-target**。
6. 对于数据库名称，输入数据库的名称。
7. 对于安全套接字层 (SSL) 模式，选择 SSL 强制执行类型。
8. 选择创建数据提供程序。

## 步骤 3：创建迁移项目

现在您可以创建迁移项目。在迁移项目中，可以指定源数据提供程序和目标数据提供程序，以及实例配置文件。

## 创建迁移项目

1. 选择迁移项目，然后选择创建迁移项目。
2. 对于名称，为迁移项目输入唯一名称。例如，输入 **sc-project**。
3. 对于实例配置文件，选择 **sc-instance**。
4. 对于源，选择浏览，然后选择 **sc-source**。
5. 对于密钥 ID，选择 **sc-source-secret**。
6. 对于 IAM 角色，选择 **sc-secrets-manager-role**。
7. 对于目标，选择浏览，然后选择 **sc-target**。
8. 对于密钥 ID，选择 **sc-target-secret**。
9. 对于 IAM 角色，选择 **schema-conversion-role**。
10. 选择创建迁移项目。

## 步骤 4：创建评测报告

要评测迁移的复杂性，可创建数据库迁移评测报告。此报告包括 DMS 架构转换无法自动转换的所有数据库对象的列表。

### 创建评测报告

1. 选择迁移项目，然后选择 **sc-project**。
2. 选择架构转换，然后选择启动架构转换。
3. 在源数据库窗格中，选择要评测的数据库架构。此外，选中此架构名称的复选框。
4. 在源数据库窗格中，从操作菜单中选择评测。将显示评测对话框。
5. 在对话框中选择评测，以确认选择。

摘要选项卡中显示了 DMS 架构转换可以为数据库存储对象和数据库代码对象自动转换的项目数量。

6. 选择操作项，查看 DMS 架构转换无法自动转换的所有数据库对象的列表。审核每一项的建议操作。
7. 要保存评测报告的副本，选择导出结果。接下来，选择以下格式之一：CSV 或 PDF。将显示导出对话框。
8. 选择导出以确认选择。
9. 选择 S3 存储桶。将打开 Amazon S3 控制台。

10. 选择下载以保存评测报告。

## 步骤 5：转换源代码

您可以使用以下过程转换源数据库架构。然后，可以将转换后的架构作为 SQL 脚本保存到文本文件中。

### 转换数据库架构

1. 在源数据库窗格中，选择要转换的数据库架构。此外，选中此架构名称的复选框。
2. 在源数据库窗格中，从操作菜单中选择转换。将显示转换对话框。
3. 在对话框中选择转换，以确认选择。
4. 在源数据库窗格中选择数据库对象。DMS 架构转换会显示此对象的源代码和转换后的代码。您可以使用编辑 SQL 功能编辑数据库对象转换后的 SQL 代码。有关更多信息，请参阅 [编辑和保存转换后的 SQL 代码](#)。
5. 在目标数据库窗格中，选择转换后的数据库架构。此外，选中此架构名称的复选框。
6. 对于操作，选择另存为 SQL。将显示保存对话框。
7. 选择另存为 SQL 以确认选择。
8. 选择 S3 存储桶。将打开 Amazon S3 控制台。
9. 选择下载以保存 SQL 脚本。

## 步骤 6：应用转换后的代码

DMS 架构转换不会立即将转换后的代码应用于目标数据库。要更新目标数据库，可以使用上一步中创建的 SQL 脚本。或者，使用以下过程应用 DMS 架构转换中转换后的代码。

### 应用转换后的代码

1. 在目标数据库窗格中，选择转换后的数据库架构。此外，选中此架构名称的复选框。
2. 对于操作，选择应用更改。将显示应用更改对话框。
3. 选择应用以确认选择。

## 步骤 7：清理和故障排除

您可以使用 Amazon CloudWatch 查看或共享您的 DMS 架构转换日志。

## 查看 DMS 架构转换日志

1. 登录 AWS Management Console 并打开 CloudWatch 控制台，[网址为 https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/)。
2. 依次选择日志、日志组。

DMS 架构转换日志组的名称以 `dms-tasks-sct` 开头。可以按创建时间对日志组进行排序，以查找 DMS 架构转换日志组。

此外，日志组的名称包括迁移项目的 Amazon 资源名称 (ARN)。可以在 DMS 架构转换的迁移项目页面上，查看项目的 ARN。请确保在首选项中选择了 ARN。

3. 选择日志组的名称，然后选择日志流的名称。
4. 对于操作，选择导出结果以保存 DMS 架构转换日志。

在 DMS 架构转换中完成架构转换后，请清理资源。

## 清理 DMS 架构转换资源

1. 登录 AWS Management Console 并打开 AWS DMS 控制台。
2. 在导航窗格中，选择迁移项目。
  - a. 选择 **sc-project**。
  - b. 选择架构转换，然后选择关闭架构转换。
  - c. 选择删除并确认选择。
3. 在导航窗格中，选择实例配置文件。
  - a. 选择 **sc-instance**。
  - b. 选择删除并确认选择。
4. 在导航窗格中，选择数据提供程序。
  - a. 选择 **sc-source** 和 **sc-target**。
  - b. 选择删除并确认选择。

此外，请务必清理您创建的其他 AWS 资源，例如您的 Amazon S3 存储桶、中的数据库密钥 AWS Secrets Manager、IAM 角色和虚拟私有云 (VPC)。

## 为 DMS 架构转换设置网络

DMS 架构转换基于 Amazon VPC 服务，在虚拟私有云 (VPC) 中创建架构转换实例。在创建实例配置文件时，可指定要使用的 VPC。可以使用账户和 AWS 区域的默认 VPC，也可以创建新的 VPC。

可以使用不同的网络配置，为源数据库和目标数据库设置通过 DMS 架构转换进行的交互。这些配置取决于源数据提供程序的位置以及网络设置。以下主题描述了常见的网络配置。

### 主题

- [为源和目标数据提供程序使用单个 VPC](#)
- [为源和目标数据提供程序使用多个 VPC](#)
- [使用 AWS Direct Connect 或 VPN 配置连接到 VPC 的网络](#)
- [使用互联网连接到 VPC](#)
- [使用没有互联网网关的环境](#)

## 为源和目标数据提供程序使用单个 VPC

DMS 架构转换最简单的网络配置是单个 VPC 配置。此时，源数据提供程序、实例配置文件和目标数据提供程序都位于同一 VPC 中。您可以使用此配置转换 Amazon EC2 实例上的源数据库。

要使用此配置，请确保实例配置文件使用的 VPC 安全组可以访问数据提供程序。例如，对于网络地址转换 (NAT) 网关，您可以允许 VPC 无类别域间路由 (CIDR) 范围或弹性 IP 地址。

## 为源和目标数据提供程序使用多个 VPC

如果源数据提供程序和目标数据提供程序位于不同 VPC 中，那么您可以在其中一个 VPC 中创建实例配置文件。然后，可以使用 VPC 对等连接将这两个 VPC 链接起来。您可以使用此配置转换 Amazon EC2 实例上的源数据库。

VPC 对等连接是两个 VPC 之间的连接，使用各 VPC 的专用 IP 地址激活路由，就像位于同一个网络中一样。您可以在自己的 VPC 之间、自己的 VPC 与另一个 AWS 账户中的 VPC 或与其他 AWS 区域中的 VPC 之间创建 VPC 对等连接。有关 VPC 对等连接的更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 对等连接](#)。

要实现 VPC 对等连接，请按照《Amazon VPC 用户指南》的 [使用 VPC 对等连接](#) 中的说明进行操作。请确保一个 VPC 的路由表中包含另一个 VPC 的 CIDR 块。例如，假设 VPC A 使用目标 10.0.0.0/16，VPC B 使用目标 172.31.0.0。在这种情况下，VPC A 的路由表应包含 172.31.0.0，VPC

B 的路由表必须包含 10.0.0.0/16。有关更多详细信息，请参阅《Amazon VPC 对等连接指南》中的[更新 VPC 对等连接的路由表](#)。

## 使用 AWS Direct Connect 或 VPN 配置连接到 VPC 的网络

远程网络可以使用多种选项连接到 VPC，例如 AWS Direct Connect 或者软件或硬件 VPN 连接。您可以使用这些选项，通过将内部网络扩展到 AWS Cloud，集成现有的现场服务。您可以集成现场服务，例如监控、身份验证、安全、数据或其他系统。通过使用此类网络扩展，您可以将现场服务无缝连接到 AWS 托管的资源，如 VPC。您可以使用此配置转换本地源数据库。

在此配置中，VPC 安全组必须包含一个路由规则，即，将以 VPC CIDR 范围或特定 IP 地址为目标的流量发送到主机。此主机必须能够将流量从 VPC 桥接到本地 VPN 中。在这种情况下，NAT 主机包括自己的安全组设置。这些设置必须允许流量从 VPC CIDR 范围或安全组进入 NAT 实例。有关更多信息，请参阅《AWS Site-to-Site VPN 用户指南》中的[创建 Site-to-Site VPN 连接](#)。

## 使用互联网连接到 VPC

如果您不使用 VPN 或 AWS Direct Connect 连接 AWS 资源，那么可以使用互联网连接。此配置涉及带互联网网关的 VPC 中的专用子网。网关包含目标数据提供程序和实例配置文件。您可以使用此配置转换本地源数据库。

要将互联网网关添加到 VPC，请参阅《Amazon VPC 用户指南》中的[连接互联网网关](#)。

VPC 路由表必须包含一个路由规则，即，将并非以 VPC 为目标的流量默认发送到互联网网关。在此配置中，与数据提供程序的连接似乎来自 NAT 网关的公共 IP 地址。有关更多信息，请参阅《Amazon VPC 用户指南》中的[VPC 路由表](#)。

## 使用没有互联网网关的环境

要在不使用互联网网关的情况下创建架构转换环境，可执行以下操作。

1. 按照 [开始使用](#) 教程中的步骤 1-3 进行操作，但进行以下更改：
  - 选择专用子网，而不是公共子网。
  - 在创建实例过程中，对于分配公共 IP，选择否。
2. 打开 Amazon VPC 控制台。
3. 选择端点，然后选择创建端点。
4. 在创建端点页面上，执行以下操作：

- 对于服务类别，选择 AWS 服务。
  - 在服务列表中，选择 `com.amazonaws.{region}.secretsmanager`
  - 在 VPC 部分中，选择您创建的 VPC。
  - 为 VPC 选择子网。
  - 为 VPC 选择安全组。
  - 对于策略，保留选中完全访问权限。
5. 完成[开始使用](#)教程的其余部分。

## 在 DMS 架构转换中创建源数据提供程序

在迁移项目中，您可以使用微软 SQL Server、Oracle 或 PostgreSQL 数据库作为 DMS 架构转换的源数据提供者。您的源数据提供程序可以是自管理引擎，可运行在本地，也可运行在 Amazon EC2 实例上。

请确保已配置网络，允许源数据提供程序与 DMS 架构转换之间的交互。有关更多信息，请参阅 [为 DMS 架构转换设置网络](#)。

### 主题

- [使用 Microsoft SQL Server 数据库作为 DMS 架构转换的源](#)
- [使用 Oracle 数据库作为 DMS 架构转换的源](#)
- [使用 Oracle 数据仓库数据库作为 DMS 架构转换源](#)
- [在 DMS 架构转换中使用 PostgreSQL 数据库作为源](#)
- [在 DMS 架构转换中使用 MySQL 数据库作为源](#)

## 使用 Microsoft SQL Server 数据库作为 DMS 架构转换的源

您可以使用 SQL Server 数据库作为 DMS 架构转换的迁移源。

您可以使用 DMS 架构转换，将数据库代码对象从 SQL Server 转换为以下目标：

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



有关支持的 SQL Server 数据库版本的信息，请参阅 [DMS 架构转换的源数据提供程序](#)。

有关在源 SQL Server 数据库中使用 DMS 架构转换的更多信息，请参阅 SQL Server 到 [MySQL 的迁移 step-by-step 演练](#)。

## 将 Microsoft SQL Server 用作源的权限

下面列出了使用 Microsoft SQL Server 作为源所需的权限：

- VIEW DEFINITION
- VIEW DATABASE STATE

VIEW DEFINITION 权限允许具有公共访问权限的用户查看对象定义。DMS 架构转换使用 VIEW DATABASE STATE 权限检查 SQL Server 企业版的功能。

对要转换其架构的每个数据库重复以上授权。

此外，授予针对 master 数据库的以下权限：

- VIEW SERVER STATE
- VIEW ANY DEFINITION

DMS 架构转换使用 VIEW SERVER STATE 权限收集服务器设置和配置。请确保已授予查看数据提供程序的 VIEW ANY DEFINITION 权限。

要读取有关 Microsoft 分析服务的信息，请在 master 数据库上运行以下命令。

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

在上例中，将 `<user_name>` 占位符替换为您之前授予其所需权限的用户名。

要阅读有关 SQL Server 代理的信息，请将您的用户添加到 SQL AgentUser 角色中。在 msdb 数据库上运行以下命令。

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

在上例中，将 `<SQLAgentRole>` 占位符替换为 SQL Server Agent 角色的名称。然后，将 `<user_name>` 占位符替换为您之前授予其所需权限的用户名。有关更多信息，请参阅 [Amazon RDS 用户指南中的向 SQL AgentUser 角色添加用户](#)。

要检测日志传输，请在 msdb 数据库上授予 SELECT on dbo.log\_shipping\_primary\_databases 权限。

要使用数据定义语言 ( DDL ) 复制的通知方法，请在源数据库上授予 RECEIVE ON `<schema_name>.<queue_name>` 权限。在本例中，将 `<schema_name>` 占位符替换为数据库的架构名称。然后，将 `<queue_name>` 占位符替换为队列表的名称。

## 使用 Oracle 数据库作为 DMS 架构转换的源

您可以使用 Oracle 数据库作为 DMS 架构转换的迁移源。

要连接到 Oracle 数据库，请使用 Oracle 系统 ID ( SID )。要查找 Oracle SID，请向 Oracle 数据库提交以下查询：

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

您可以使用 DMS 架构转换，将数据库代码对象从 Oracle 数据库转换为以下目标：

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

有关支持的 Oracle 数据库版本的信息，请参阅[DMS 架构转换的源数据提供程序](#)。

有关对源 Oracle 数据库使用 DMS 架构转换的更多信息，请参阅 [Oracle 到 PostgreSQL 的迁移演练](#)。step-by-step

### 将 Oracle 用作源的权限

将 Oracle 作为源需要以下权限：

- CONNECT
- SELECT\_CATALOG\_ROLE
- SELECT ANY DICTIONARY
- SELECT ON SYS.ARGUMENT\$

## 使用 Oracle 数据仓库数据库作为 DMS 架构转换源

在 DMS 架构转换中，您可以使用 Oracle 数据仓库数据库作为迁移源，将数据库代码对象和应用程序代码转换为 Amazon Redshift。

有关支持的 Oracle 数据库版本的信息，请参阅[DMS 架构转换的源数据提供程序](#)。有关对源 Oracle 数据库使用 DMS 架构转换的更多信息，请参阅 [Oracle 到 PostgreSQL 的迁移演练](#)。step-by-step

### 将 Oracle 数据仓库数据库用作源的权限

以下是将 Oracle 数据仓库用作源所需的权限：

- CONNECT
- SELECT\_CATALOG\_ROLE
- SELECT ANY DICTIONARY

### Oracle 数据仓库到 Amazon Redshift 的转换设置

有关编辑 DMS 架构转换设置的信息，请参阅[为迁移项目指定架构转换设置](#)。

Oracle 数据仓库到 Amazon Redshift 转换设置包括以下各项：

- 在转换后的代码中为所选严重性级别及更高级别的操作项添加注释：此设置限制转换后的代码中操作项的注释数量。DMS 会在转换后的代码中为所选严重性级别及更高级别的操作项添加注释。

例如，要最大限度地减少转换后的代码中的注释数量，请选择仅错误。要在转换后的代码中包含所有操作项的注释，请选择所有消息。

- 目标 Amazon Redshift 集群的最大表数：通过此设置可以设置 DMS 可以应用于目标 Amazon Redshift 集群的最大表数。Amazon Redshift 具有限制了不同集群节点类型使用表数的配额。此设置支持以下值：
  - 自动：DMS 会根据节点类型确定要应用于目标 Amazon Redshift 集群的表数。
  - 设置值：手动设置表数。

即使表数超过了您的 Amazon Redshift 集群可以存储的限值，DMS 也会转换您的所有源表。DMS 会将转换后的代码存储在您的项目中，不会将其应用于目标数据库。如果应用转换后的代码时达到了 Amazon Redshift 集群的表配额，DMS 会显示一条警告消息。此外，DMS 还会将表应用于目标 Amazon Redshift 集群，直到表的数量达到限值。

有关 Amazon Redshift 表配额的信息，请参阅 [Amazon Redshift 中的配额和限制](#)。

- 使用 UNION ALL 视图：此设置允许您设置 DMS 可以为单个源表创建的最大目标表数。

Amazon Redshift 不支持表分区。要模拟表分区并加快查询运行速度，DMS 可以将源表的每个分区迁移到 Amazon Redshift 中的单独表中。然后，DMS 会创建一个视图，其中包含它创建的所有目标表中的数据。

DMS 会自动确定源表中的分区数量。根据源表分区的类型，此数量可能会超过您可以应用于 Amazon Redshift 集群的表配额。为避免达到此配额，请输入 DMS 可以为单个源表的分区创建的最大目标表数。默认选项为 368 个表，它表示一年中 366 天的分区和以及 NO RANGE 和 UNKNOWN 分区的两个表。

- Oracle 代码中使用的日期类型格式元素与 Amazon Redshift 中的日期时间格式字符串相似：使用此设置可以转换数据类型格式函数（例如 TO\_CHAR、TO\_DATE 和 TO\_NUMBER）以及 Amazon Redshift 不支持的日期时间格式元素。默认情况下，DMS 使用扩展包函数来模拟转换后的代码中这些不支持的格式元素。

与 Amazon Redshift 中的日期时间格式字符串相比，Oracle 中的日期时间格式模型包含更多元素。如果源代码仅包含 Amazon Redshift 支持的日期时间格式元素，设置此值可以避免在转换后的代码中使用扩展包函数。避免使用扩展函数可以使转换后的代码运行得更快。

- Oracle 代码中使用的数字格式元素与 Amazon Redshift 中的数字格式字符串相似：使用此设置可以转换 Amazon Redshift 不支持的数字数据类型格式函数。默认情况下，DMS 使用扩展包函数来模拟转换后的代码中这些不支持的格式元素。

与 Amazon Redshift 中的数字格式字符串相比，Oracle 中的数字格式模型包含更多的元素。如果源代码仅包含 Amazon Redshift 支持的数字格式元素，设置此值可以避免在转换后的代码中使用扩展包函数。避免使用扩展函数可以使转换后的代码运行得更快。

- 使用 NVL 函数模拟 Oracle LEAD 和 LAG 函数的行为：如果您的源代码不使用 LEAD 和 LAG 函数中偏移量的默认值，DMS 可以使用 NVL 函数来模拟这些函数。默认情况下，DMS 会为每次使用 LEAD 和 LAG 函数引发一个操作项。使用 NVL 模拟这些函数可以使转换后的代码运行得更快。
- 模拟主键和唯一键的行为：进行此设置可使 DMS 模拟目标 Amazon Redshift 集群上的主键和唯一键约束的行为。Amazon Redshift 不强制使用主键和唯一键约束，它们仅用于提供信息。如果源代码使用主键或唯一键约束，进行此设置可确保 DMS 模拟其行为。
- 使用压缩编码：进行此设置可将压缩编码应用于 Amazon Redshift 表列。DMS 使用默认 Redshift 算法自动分配压缩编码。有关压缩编码的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的[压缩编码](#)。

默认情况下，Amazon Redshift 不对定义为排序键和分配键的列应用压缩。要对这些列应用压缩，请设置为 KEY 列使用压缩编码。只有设置使用压缩编码时，才能选择此选项。

## 在 DMS 架构转换中使用 PostgreSQL 数据库作为源

您可以在 DMS 架构转换中使用 PostgreSQL 数据库作为迁移源。

您可以使用 DMS 架构转换将数据库代码对象从 PostgreSQL 数据库转换为以下目标：

- MySQL
- Aurora MySQL

下面列出了将 PostgreSQL 用作源所需的权限：

- 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>

## 在 DMS 架构转换中使用 MySQL 数据库作为源

您可以在 DMS 架构转换中使用 MySQL 数据库作为迁移源。

您可以使用 DMS 架构转换将数据库代码对象从 MySQL 数据库转换为以下目标：

- PostgreSQL
- Aurora PostgreSQL

下面是将 MySQL 用作源所需的权限：

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

## MySQL 到 PostgreSQL 的转换设置

有关编辑 DMS 架构转换设置的信息，请参阅[为迁移项目指定架构转换设置](#)。

MySQL 到 PostgreSQL 的转换设置包括以下内容：

- 转换后的 SQL 代码中的注释：设置此设置可在转换后的代码中为所选严重性及更高的操作项添加注释。

有效值：

- 仅限错误
- Errors and warnings ( 错误和警告 )
- 所有消息

## 在 DMS 架构转换中创建目标数据提供程序

在 DMS 架构转换的迁移项目中，您可以使用 MySQL 和 PostgreSQL 数据库作为目标数据提供程序。目标数据提供程序可以是 Amazon EC2、Amazon RDS 或 Amazon Aurora 实例。

主题

- [使用 MySQL 数据库作为 DMS 架构转换的目标](#)
- [使用 PostgreSQL 数据库作为 DMS 架构转换的目标](#)
- [在 DMS 架构转换中使用 Amazon Redshift 集群作为目标](#)

## 使用 MySQL 数据库作为 DMS 架构转换的目标

您可以使用 MySQL 数据库作为 DMS 架构转换的迁移目标。

有关支持的目标数据库的信息，请参阅 [DMS 架构转换的目标数据提供程序](#)。

### 使用 MySQL 作为目标所需的权限

使用 MySQL 作为目标需要以下权限：

- 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.\*

您可以使用以下代码示例创建数据库用户并授予权限。

```
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';
```

在前面的示例中，将 *user\_name* 替换为用户名。然后，将 *your\_password* 替换为安全密码。

要使用 Amazon RDS for MySQL 或 Aurora MySQL 作为目标，请将 `lower_case_table_names` 参数设置为 1。此值意味着 MySQL 服务器在处理表、索引、触发器和数据库等对象名称的标识符时不区

分大小写。如果目标实例中已开启二进制日志记录，请将 `log_bin_trust_function_creators` 参数设置为 1。在这种情况下，您无需使用 `DETERMINISTIC`、`READS SQL DATA` 或 `NO SQL` 特性创建存储函数。要配置这些参数，可创建新的数据库参数组或修改现有数据库参数组。

## 使用 PostgreSQL 数据库作为 DMS 架构转换的目标

您可以使用 PostgreSQL 数据库作为 DMS 架构转换的迁移目标。

有关支持的目标数据库的信息，请参阅 [DMS 架构转换的目标数据提供程序](#)。

### 将 PostgreSQL 作为目标所需的权限

要将 PostgreSQL 作为目标，DMS 架构转换需要 `CREATE ON DATABASE` 权限。对于要在 DMS 架构转换的迁移项目中使用的每个数据库，请创建一个用户，并向该用户授予此权限。

要使用 Amazon RDS for PostgreSQL 作为目标，DMS 架构转换需要 `rds_superuser` 角色。

要使用转换后的公共同义词，请使用以下命令更改数据库的默认搜索路径。

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

在本例中，将 `<db_name>` 占位符替换为数据库的名称。

在 PostgreSQL 中，只有架构所有者或 `superuser` 才能删除架构。架构的所有者可以删除架构及其包含的所有对象，即使该所有者并不拥有架构的某些对象。

如果使用不同的用户将不同架构转换并应用到目标数据库，那么当 DMS 架构转换无法删除架构时，您可能会收到错误消息。为避免此类错误消息，请使用 `superuser` 角色。

## 在 DMS 架构转换中使用 Amazon Redshift 集群作为目标

您可以使用 Amazon Redshift 数据库作为 DMS 架构转换的迁移目标。有关支持的目标数据库的信息，请参阅 [DMS 架构转换的目标数据提供程序](#)。

### 将 Amazon Redshift 作为目标需具备的权限

使用 Amazon Redshift 作为 DMS 架构转换目标需要具备以下权限：

- `CREATE ON DATABASE`：允许 DMS 在数据库中创建新架构。



- CREATE ON SCHEMA：允许 DMS 在数据库架构中创建对象。
- GRANT USAGE ON LANGUAGE：允许 DMS 在数据库中创建新函数和过程。
- GRANT SELECT ON ALL TABLES IN SCHEMA pg\_catalog：为用户提供有关 Amazon Redshift 集群的系统信息。
- GRANT SELECT ON pg\_class\_info：为用户提供有关表分配方式的信息。

您可以使用以下代码示例创建数据库用户并授予权限。将示例值替换为您的值。

```
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;
```

对每个目标架构重复 GRANT CREATE ON SCHEMA 操作，您将在其中应用转换后的代码或迁移数据。

您可以在目标 Amazon Redshift 数据库上应用扩展包。扩展包是一个附加模块，用于模拟将对象转换为 Amazon Redshift 时所需的源数据库函数。有关更多信息，请参阅[在 DMS 架构转换中使用扩展包](#)。

## 在 DMS 架构转换中管理迁移项目

为架构转换创建实例配置文件和兼容的数据提供程序之后，可创建迁移项目。有关更多信息，请参阅[创建迁移项目](#)。

要在 DMS 架构转换中使用这个新项目，请在迁移项目页面上，从列表中选择此项目。接下来，在架构转换选项卡上，选择启动架构转换。

首次启动 DMS 架构转换需要进行一些设置。AWS Database Migration Service(AWS DMS) 启动架构转换实例最长需要 15 分钟。此过程还会从源数据库和目标数据库读取元数据。首次成功启动后，可以更快地访问 DMS 架构转换。

Amazon 将在您完成迁移项目后的三天内，终止该项目使用的架构转换实例。您可以从用于 DMS 架构转换的 Amazon S3 存储桶中，检索转换后的架构和评测报告。

## 为 DMS 架构转换指定迁移项目设置

创建迁移项目并启动架构转换之后，可以指定迁移项目设置。您可以更改转换设置，以提高转换后代码的性能。此外，还可以自定义架构转换视图。

转换设置取决于源数据库平台和目标数据库平台。有关更多信息，请参阅 [创建源数据提供程序](#) 和 [创建目标数据提供程序](#)。

要指定在源数据库和目标数据库窗格中显示的架构和数据库，可使用树视图设置。您可以隐藏空架构、空数据库、系统数据库，以及用户定义的数据库和架构。

在树视图中隐藏数据库和架构

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后在架构转换选项卡上选择启动架构转换。
4. 选择设置。将打开设置页面：
5. 在树视图部分中，执行以下操作：
  - 选择隐藏空架构以隐藏空的架构。
  - 选择隐藏空数据库以隐藏空的数据库。
  - 对于系统数据库或架构，按名称选择系统数据库和架构以进行隐藏。
  - 对于用户定义的数据库或架构，输入要隐藏的用户定义的数据库和架构的名称。选择添加。名称不区分大小写。

要添加多个数据库或架构，可使用逗号分隔其名称。要添加多个名称相似的对象，可使用百分号 (%) 作为通配符。此通配符可替换数据库或架构名称中任意数量的任何符号。

对于源和目标部分，重复上述步骤。

6. 选择应用，然后选择架构转换。

## 使用 DMS 架构转换创建数据库迁移评测报告

DMS 架构转换的一个重要环节是它会生成一份报告，以帮助您转换架构。此数据库迁移评测报告汇总了所有架构转换任务。它还详细介绍了无法转换为目标数据库实例的数据库引擎的架构的操作项目。

您可以在 AWS DMS 控制台中查看报告，或者以 PDF 或逗号分隔值 (CSV) 文件格式保存此报告的副本。

该迁移评估报告包括以下内容：

- 执行摘要
- 建议，包括服务器对象的转换、备份建议和链接的服务器更改

当 DMS 架构转换无法自动转换您的项目时，报告会估算为您的目标数据库实例编写等效项目代码所需的工作量。

主题

- [创建数据库迁移评测报告](#)
- [查看数据库迁移评测报告](#)
- [保存数据库迁移评测报告](#)

## 创建数据库迁移评测报告

创建迁移项目后，请使用以下过程创建数据库迁移评测报告。

创建数据库迁移评估报告

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后选择架构转换。
4. 选择启动架构转换。将打开架构转换页面。
5. 在源数据库窗格中，选择您要评测的数据库架构或架构项目。要在报告中包含多个对象，请确保选择所有项目。
6. 选中要评测的所有架构对象的复选框后，您必须为所选对象选择父节点。源数据库窗格中的操作菜单现已可用。
7. 在操作菜单中，选择评测。您将看到确认对话框。
8. 在对话框中选择评测，以确认选择。

## 查看数据库迁移评测报告

创建评测报告后，DMS 架构转换会在以下选项卡中添加信息：

- 摘要
- 操作项目

摘要选项卡显示 DMS 架构转换可以自动转换的项目数量。

操作项目选项卡显示 DMS 架构转换无法自动转换的项目，并提供有关如何管理这些项目的建议。

### 评估报告摘要

摘要选项卡显示了来自数据库迁移评测报告的摘要信息。它显示了 DMS 架构转换可以自动转换为数据库存储对象和数据库代码对象的项目数量。

在大多数情况下，DMS 架构转换无法自动将所有架构项目转换为目标数据库引擎。该摘要选项卡提供在目标数据库实例中创建等同于源数据库的架构项目所需工作量的估算。

要查看数据库存储对象（例如表、序列、约束、数据类型等）的转换摘要，请选择数据库存储对象。

要查看数据库代码对象（如过程、函数、视图、触发器等）的转换摘要，请选择数据库代码对象。

要更改评测报告的范围，请在源数据库树中选择所需的节点。DMS 架构转换会更新评测报告摘要以匹配所选范围。

### 评测报告操作项目

操作项目选项卡包含 DMS 架构转换无法自动转换为与目标数据库引擎兼容的格式的项目列表。对于每个操作项目，DMS 架构转换均提供问题描述和建议的操作。DMS 架构转换对相似的操作项目分组并显示出现次数。

要查看相关数据库对象的代码，请在列表中选择一個操作项目。

## 保存数据库迁移评测报告

创建数据库迁移评测报告后，您可以将此报告的副本另存为 PDF 或逗号分隔值 (CSV) 文件。

将数据库迁移评测报告保存为 PDF 文件

1. 选择导出，然后选择 PDF。查看对话框，然后选择导出为 PDF。

2. DMS 架构转换使用您的 PDF 文件创建一个档案，并将该档案存储在您的 Amazon S3 存储桶中。要更改 Amazon S3 存储桶，请在您的实例配置文件中编辑架构转换设置。
3. 打开 Amazon S3 存储桶中的评测报告文件。

将数据库迁移评测报告另存为 CSV 文件

1. 选择导出，然后选择 CSV。查看对话框，然后选择导出为 CSV。
2. DMS 架构转换使用 CSV 文件创建一个档案，并将该档案存储在您的 Amazon S3 存储桶中。要更改 Amazon S3 存储桶，请在您的实例配置文件中编辑架构转换设置。
3. 打开 Amazon S3 存储桶中的评测报告文件。

PDF 文件包含摘要和操作项信息。

将评测报告导出为 CSV 时，DMS 架构转换会创建三个 CSV 文件。

第一个 CSV 文件包含有关行动项目的以下信息：

- 类别
- 出现
- 行动项目
- 主题
- 组
- 描述
- 文档参考
- 建议的操作
- 行
- 位置
- 源
- 目标
- 服务器 IP 地址和端口
- 数据库
- 架构

第二个 CSV 文件的名称中包含 Action\_Items\_Summary 后缀，并包含以下信息：

- 架构
- 行动项目
- 出现次数
- 学习曲线工作量，即设计一种方法来转换各个操作项目所需的工作量
- 按照设计的方法，转换出现的操作项目的工作量，这显示了转换各个操作项目所需的工作量
- 操作项目描述
- 建议的操作

指示所需工作量水平的值基于加权量表，范围从低（最小）到高（最大）。

第三个 CSV 文件的名称中包含 Summary，并包含以下信息：

- 类别
- 对象数
- 已自动转换的对象
- 需要简单操作的对象
- 需要中等复杂操作的对象
- 需要复杂操作的对象
- 代码总行数

## 使用 DMS 架构转换

DMS 架构转换会将现有数据库架构和大部分数据库代码对象，转换为与目标数据库兼容的格式。

DMS 架构转换可以自动执行将联机事务处理（OLTP）数据库架构转换为 Amazon RDS for MySQL 或 RDS for PostgreSQL 的大部分过程。源数据库和目标数据库引擎包含许多不同的特性和功能，DMS 架构转换尽可能尝试创建等效架构。对于无法直接转换的数据库对象，DMS 架构转换提供了一系列操作供您执行。

要转换数据库架构，请使用以下过程：

- 在转换数据库架构之前，设置在转换过程中更改数据库对象名称的转换规则。

- 创建数据库迁移评测报告，以评测迁移的复杂性。此报告提供了有关 DMS 架构转换无法自动转换的架构元素的详细信息。
- 转换源数据库存储和代码对象。DMS 架构转换会创建转换后的数据库对象的本地版本。您可以在迁移项目中访问这些转换后的对象。
- 将转换后的代码另存为 SQL 文件，以查看、编辑或处理转换操作项。或者，将转换后的代码直接应用于目标数据库。

要转换数据仓库架构，请使用桌面 AWS Schema Conversion Tool。有关更多信息，请参阅《AWS Schema Conversion Tool 用户指南》中的[将数据仓库架构转换为 Amazon Redshift](#)。

## 主题

- [在 DMS 架构转换中设置转换规则](#)
- [在 DMS 架构转换中转换数据库架构](#)
- [为迁移项目指定架构转换设置](#)
- [在 DMS 架构转换中刷新数据库架构](#)
- [在 DMS 架构转换中保存和应用转换后的代码](#)

## 在 DMS 架构转换中设置转换规则

在使用 DMS 架构转换来转换源数据库架构之前，您可以设置转换规则。转换规则可以执行诸如将对象名称更改为小写或大写、添加或删除前缀或后缀，以及重命名对象之类的操作。例如，假设源架构中有一组名为 test\_TABLE\_NAME 的表。您可以设置一条规则，将前缀 test\_ 在目标架构中更改为前缀 demo\_。

您可以创建执行以下任务的转换规则：

- 添加、删除或替换前缀
- 添加、删除或替换后缀
- 更改列的数据类型
- 将对象名称更改为小写或大写
- 重命名对象

您可以为以下对象创建转换规则：

- 架构

- 表
- 列

## 创建转换规则

DMS 架构转换将转换规则存储为迁移项目的一部分。您可以在创建迁移项目时设置转换规则，也可以稍后对其进行编辑。

您可以在项目中添加多个转换规则。在转换过程中，DMS 架构转换应用转换规则的顺序与规则的添加顺序相同。

### 创建转换规则

1. 在创建迁移项目页面上，选择添加转换规则。有关更多信息，请参阅 [创建迁移项目](#)。
2. 对于规则目标，选择要应用此规则的数据库对象的类型。
3. 对于源架构，选择输入架构。然后，输入此规则适用的源架构、表和列的名称。您可以输入确切名称以选择一个对象，也可以输入模式来选择多个对象。可使用百分号 ( % ) 作为通配符，替换数据库对象名称中任意数量的任何符号。
4. 对于操作，选择要执行的任务。
5. 根据规则类型的不同，输入一个或两个其他值。例如，要重命名对象，请输入对象的新名称。要替换前缀，则输入旧前缀和新前缀。
6. 选择添加转换规则以添加其他转换规则。

添加完规则后，选择创建迁移项目。

要复制现有的转换规则，可选择复制。要编辑现有的转换规则，可从列表中选择此规则。要删除现有的转换规则，选择删除。

## 编辑转换规则

您可以在迁移项目中新增、删除转换规则，或编辑现有的转换规则。由于 DMS 架构转换在架构转换启动期间应用转换规则，因此请务必先关闭架构转换，然后在编辑规则后再次启动架构转换。

### 编辑转换规则

1. 登录并打开 AWS DMS 控制台 AWS Management Console，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。



2. 选择迁移项目，然后选择您的迁移项目。
3. 选择架构转换，然后选择关闭架构转换。
4. AWS DMS 关闭架构转换后，选择 Modify 以编辑您的迁移项目设置。
5. 对于转换规则，请选择下列操作之一：
  - 选择复制，以复制现有的转换规则并将其添加到列表末尾。
  - 选择删除，以删除现有的转换规则。
  - 选择现有的转换规则进行编辑。
6. 完成编辑后，选择保存更改。
7. 在迁移项目页面上，从列表中选择您的项目。选择架构转换，然后选择启动架构转换。

## 在 DMS 架构转换中转换数据库架构

创建迁移项目并连接到源数据库和目标数据库之后，您可以将源数据库对象转换为与目标数据库兼容的格式。DMS 架构转换将以树状视图格式，将源数据库架构显示在左侧面板中。

数据库树的每个节点都是延迟加载的。如果您在树状视图中选择节点，那么 DMS 架构转换会在此时请求源数据库提供架构信息。要更快地加载架构信息，可选择您的架构，然后从操作菜单中选择加载元数据。然后，DMS 架构转换读取数据库元数据，并将信息存储在 Amazon S3 存储桶中。现在，您可以更快地浏览数据库对象。

您可以转换整个数据库架构，也可以从源数据库中选择任何架构项进行转换。如果您选择的架构项依赖于父项，则 DMS 架构转换也会为父项生成架构。例如，如果您选择表进行转换，DMS 架构转换会创建转换后的表以及包含该表的数据库架构。

### 转换数据库对象

您可以使用 DMS 架构转换，转换整个数据库架构或单独的数据库对象。

#### 转换整个数据库架构

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后选择架构转换。
4. 选择启动架构转换。将打开架构转换页面。

5. 在源数据库窗格中，选中架构名称的复选框。
6. 在迁移项目的左侧窗格中选择此架构。DMS 架构转换用蓝色突出显示架构名称，并激活操作菜单。
7. 对于操作，选择转换。将显示转换对话框。
8. 在对话框中选择转换，以确认选择。

### 转换源数据库对象

1. 登录并打开 AWS DMS 控制台 AWS Management Console，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后选择架构转换。
4. 选择启动架构转换。将打开架构转换页面。
5. 在源数据库窗格中，选择您的源数据库对象。
6. 选中要转换的所有对象的复选框后，在左侧窗格中选择所有所选对象的父节点。

DMS 架构转换用蓝色突出显示父节点，并激活操作菜单。

7. 对于操作，选择转换。将显示转换对话框。
8. 在对话框中选择转换，以确认选择。

例如，要转换 10 个表中的两个，请选中要转换的两个表的复选框。请注意，操作菜单处于非活动状态。选择了表节点之后，DMS 架构转换用蓝色突出显示其名称，并激活操作菜单。然后，您可以从此菜单中选择转换。

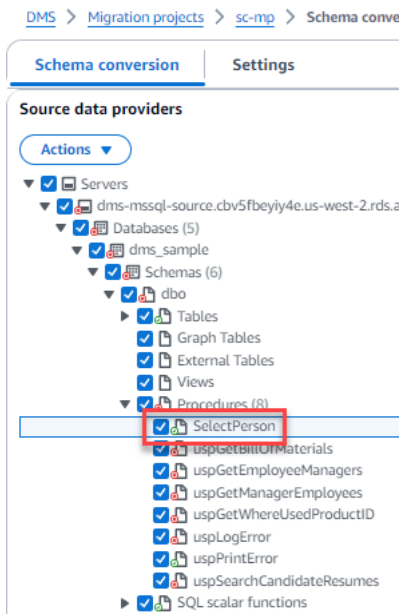
同样，要转换两个表和三个过程，请选中对象名称的复选框。然后，选择架构节点以激活操作菜单，然后选择转换架构。

### 编辑和保存转换后的 SQL 代码

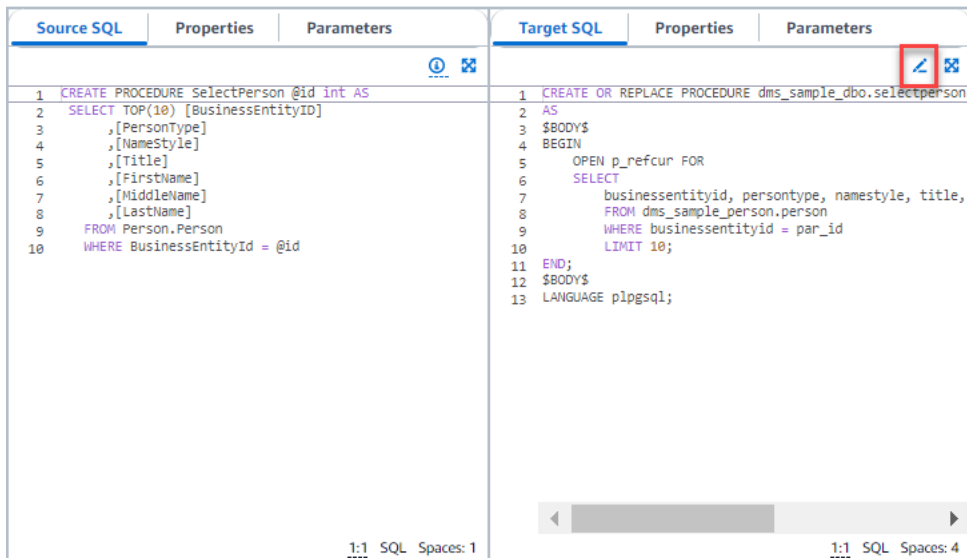
架构转换页面允许您编辑数据库对象中转换的 SQL 代码。按照以下过程编辑转换后的 SQL 代码、应用更改，然后进行保存。

#### 编辑、应用更改并保存转换后的 SQL 代码

1. 在“架构转换”页面中，打开“源数据提供者”窗格中的树视图以显示代码对象。



2. 在源数据提供者窗格中，选择操作，转换。确认该操作。
3. 转换完成后，要查看转换后的 SQL，请根据需要展开中心窗格。要编辑转换后的 SQL，请在“目标 SQL”窗格中选择“编辑”图标。



4. 编辑目标 SQL 后，通过选择页面顶部的复选图标来确认所做的更改。确认该操作。
5. 在目标数据提供者窗格中，选择操作，应用更改。确认该操作。
6. DMS 将编辑后的过程写入目标数据存储。

## 查看转换后的数据库对象

转换源数据库对象之后，可以在项目的左侧窗格中选择对象。然后，您可以查看该对象的源代码和转换后的代码。DMS 架构转换会为您在左窗格中选择的对象，自动加载转换后的代码。还可以查看所选对象的属性或参数。

DMS 架构转换会将转换后的代码自动存储为迁移项目的一部分。它不会将这些代码更改应用于目标数据库。有关将转换后的代码应用于目标数据库的更多信息，请参阅[应用转换后的代码](#)。要从迁移项目中删除转换后的代码，请在右侧窗格中选择目标架构，然后从操作中选择从数据库刷新。

转换源数据库对象之后，可以在中间下方的窗格中查看转换摘要和操作项。创建评测报告时，可以看到相同的信息。评测报告对于识别和解析 DMS 架构转换无法转换的架构项很有用。您可以将评测报告摘要和转化操作项列表保存为 CSV 文件。有关更多信息，请参阅[数据库迁移评测报告](#)。

## 为迁移项目指定架构转换设置

创建迁移项目后，可以在 DMS 架构转换中指定转换设置。配置架构转换设置可提高转换后的代码的性能。

### 编辑转换设置

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目。选择架构转换，然后选择启动架构转换。
4. 选择设置。将打开设置页面：
5. 在转换部分中，更改设置。
6. 选择应用，然后选择架构转换。

对于所有转化对，您可以限制转换后的代码中含操作项的注释数量。要限制转换后的代码中的注释数量，请在迁移项目中打开转换设置。

对于转换后的 SQL 代码中的注释，选择操作项的严重性级别。DMS 架构转换会在转换后的代码中，为所选严重性级别及更高级别的操作项添加注释。例如，要最大限度地减少转换后的代码中的注释数量，可选择仅错误。

要在转换后的代码中包含所有操作项的注释，可选择所有消息。

每对源数据库和目标数据库的其他转换设置都不同。

## 主题

- [Oracle 到 MySQL 的转换设置](#)
- [Oracle 到 PostgreSQL 的转换设置](#)
- [SQL Server 到 MySQL 的转换设置](#)
- [SQL Server 到 PostgreSQL 的转换设置](#)
- [PostgreSQL 到 MySQL 的转换设置](#)
- [适用于 z/OS 的 DB2 到 DB2 LUW 的转换设置](#)

## Oracle 到 MySQL 的转换设置

在 DMS 架构转换中，Oracle 到 MySQL 的转换设置包括以下内容：

- 源 Oracle 数据库可以使用 ROWID 伪列。MySQL 不支持类似功能。DMS 架构转换可以在转换后的代码中模拟 ROWID 伪列。为此，请打开生成行 ID 选项。

如果源 Oracle 代码未使用 ROWID 伪列，请关闭生成行 ID 选项。这种情况下，转换后的代码运行更快。

- 源 Oracle 代码可以包含 TO\_CHAR、TO\_DATE 和 TO\_NUMBER 函数以及 MySQL 不支持的参数。默认情况下，DMS 架构转换会在转换后的代码中模拟这些参数的用法。

当源 Oracle 代码缺少 MySQL 不支持的参数时，您可以使用本机 MySQL TO\_CHAR、TO\_DATE 和 TO\_NUMBER 函数。这种情况下，转换后的代码运行更快。为此，可选择以下值：

- 使用本机 MySQL TO\_CHAR 函数
- 使用本机 MySQL TO\_DATE 函数
- 使用本机 MySQL TO\_NUMBER 函数
- 数据库和应用程序可以运行于不同时区内。默认情况下，DMS 架构转换在转换后的代码中模拟时区。但是，当数据库和应用程序使用相同时区时，您不需要这种模拟。在这种情况下，请选择在数据库和应用程序使用相同时区的情况下提高转换后的代码的性能。

## Oracle 到 PostgreSQL 的转换设置

在 DMS 架构转换中，Oracle 到 PostgreSQL 的转换设置包括以下内容：

- AWS DMS 可以在 PostgreSQL 上将 Oracle 实例化视图转换为表或实例化视图。对于实例化视图，请选择如何转换源实例化视图。

- 源 Oracle 数据库可以使用 ROWID 伪列。PostgreSQL 不支持类似功能。DMS 架构转换可以使用 `bigint` 或 `character varying` 数据类型，在转换后的代码中模拟 ROWID 伪列。为此，对于行 ID，请选择使用 `bigint` 数据类型模拟 ROWID 伪列或使用可变字符数据类型模拟 ROWID 伪列。

如果源 Oracle 代码不使用 ROWID 伪列，请选择不生成。这种情况下，转换后的代码运行更快。

- 源 Oracle 代码可以包含 `TO_CHAR`、`TO_DATE` 和 `TO_NUMBER` 函数以及 PostgreSQL 不支持的参数。默认情况下，DMS 架构转换会在转换后的代码中模拟这些参数的用法。

当源 Oracle 代码缺少 PostgreSQL 不支持的参数时，您可以使用本机 PostgreSQL `TO_CHAR`、`TO_DATE` 和 `TO_NUMBER` 函数。这种情况下，转换后的代码运行更快。为此，可选择以下值：

- 使用本机 PostgreSQL `TO_CHAR` 函数
- 使用本机 PostgreSQL `TO_DATE` 函数
- 使用本机 PostgreSQL `TO_NUMBER` 函数
- 数据库和应用程序可以运行于不同时区内。默认情况下，DMS 架构转换在转换后的代码中模拟时区。但是，当数据库和应用程序使用相同时区时，您不需要这种模拟。在这种情况下，请选择在数据库和应用程序使用相同时区的情况下提高转换后的代码的性能。
- 要在转换后的代码中继续使用序列，可选择使用源端生成的最后一个值填充转换后的序列。
- 在某些情况下，源 Oracle 数据库在 `NUMBER` 数据类型的主键列或外键列中，可能只存储整数值。在这些情况下，AWS DMS 可以将这些列转换为 `BIGINT` 数据类型。这种方法可提高转换后的代码的性能。为此，可选择将 `NUMBER` 数据类型的主键列和外键列转换为 `BIGINT` 数据类型。请确保源代码的这些列中不包含浮点值，以避免数据丢失。
- 要跳过源代码中已停用的触发器和约束，可选择仅转换活动触发器和约束。
- 可以使用 DMS 架构转换，转换被称为动态 SQL 的字符串变量。数据库代码可以更改这些字符串变量的值。要确保 AWS DMS 始终转换此字符串变量的最新值，请选择“转换在调用例程中创建的动态 SQL 代码”。
- PostgreSQL 版本 10 及更早版本不支持过程。如果你不熟悉在 PostgreSQL 中使用过程，可以将 Oracle 过程转换为 PostgreSQL AWS DMS 函数。为此，可选择将过程转换为函数。
- 要查看有关已发生操作项的其他信息，您可以向扩展包中添加特定函数。为此，可选择添加引发用户定义的异常的扩展包函数。然后，选择引发用户定义的异常的严重性级别。请务必在转换源数据库对象后，应用扩展包架构。有关扩展包的更多信息，请参阅[使用扩展包](#)。
- 源 Oracle 数据库可以包含具自动生成的名称的约束。如果源代码使用这些名称，请确保选择保留系统生成的约束的名称。如果源代码使用了这些约束，但未使用其名称，请清除此选项以提高转换速度。

- 如果源数据库和目标数据库运行于不同时区内，那么模拟 SYSDATE 内置 Oracle 函数的函数返回的值会与源函数有所不同。要确保源函数和目标函数返回相同的值，请选择设置源数据库的时区。
- 您可以在转换后的代码中使用 oraflce 扩展中的函数。为此，对于 Oraflce 内置例程，请选择要使用的函数。有关 orace 的更多信息，请参阅 [oraflce on](#)。GitHub

## SQL Server 到 MySQL 的转换设置

在 DMS 架构转换中，SQL Server 到 MySQL 的转换设置包括以下内容：

- 源 SQL Server 数据库可以将 EXEC 的输出存储在表中。DMS 架构转换创建临时表和一个额外的过程来模拟此功能。要使用此模拟，可选择创建额外的例程处理开放数据集。

## SQL Server 到 PostgreSQL 的转换设置

在 DMS 架构转换中，SQL Server 到 PostgreSQL 的转换设置包括以下内容：

- 在 SQL Server 中，可以在不同表中使用名称相同的索引。但是，在 PostgreSQL 中，在架构中使用的所有索引名称必须是唯一的。要确保 DMS 架构转换为所有的索引生成唯一名称，可选择为索引生成唯一名称。
- PostgreSQL 版本 10 及更早版本不支持过程。如果你不熟悉在 PostgreSQL 中使用过程，可以将 SQL Server 过程转换为 Postgre AWS DMS SQL 函数。为此，可选择将过程转换为函数。
- 源 SQL Server 数据库可以将 EXEC 的输出存储在表中。DMS 架构转换创建临时表和一个额外的过程来模拟此功能。要使用此模拟，可选择创建额外的例程处理开放数据集。
- 您可以定义模板，用于转换后的代码中的架构名称。对于架构名称，请选择下列选项之一：
  - DB – 使用 SQL Server 数据库名称作为 PostgreSQL 中的架构名称。
  - SCHEMA – 使用 SQL Server 架构名称作为 PostgreSQL 中的架构名称。
  - DB\_SCHEMA – 使用 SQL Server 数据库名称和架构名称的组合作为 PostgreSQL 中的架构名称。
- 您可以保留源对象名称的字母大小写。为避免将对象名称转换为小写，可选择保留对象名称的大小写。仅当您在目标数据库中启用了区分大小写选项时，此选项才适用。
- 您可以保留源数据库中的参数名称。DMS 架构转换可以在转换后的代码中，为参数名称添加双引号。为此，可选择保留原始参数名称。
- 您可以从源数据库中保留一定长度的例程参数。DMS 架构转换会创建域并使用它们来指定例程参数的长度。为此，请选择保留参数长度。

## PostgreSQL 到 MySQL 的转换设置

DMS 架构转换中的 PostgreSQL 到 MySQL 转换设置包括以下内容：

- 转换后的 SQL 代码中的注释：此设置包括在转换后的代码中对所选严重性及更高的操作项的注释。此设置支持以下值：
  - 仅错误
  - Errors and warnings ( 错误和警告 )
  - 所有消息

## 适用于 z/OS 的 DB2 到 DB2 LUW 的转换设置

DMS 架构转换中的 DB2 for z/OS 到 DB2 LUW 转换设置包括以下内容：

- 转换后的 SQL 代码中的注释：此设置包括在转换后的代码中对所选严重性及更高的操作项的注释。此设置支持以下值：
  - 仅错误
  - Errors and warnings ( 错误和警告 )
  - 所有消息

## 在 DMS 架构转换中刷新数据库架构

创建迁移项目之后，DMS 架构转换会将有关源架构和目标架构的信息存储在此项目中。DMS 架构转换使用延迟加载，仅在需要时加载元数据，例如，当您在数据库树中选择节点时。您可以使用预先加载，更快地加载架构信息。为此，可选择您的架构，然后从操作中选择加载元数据。

将对象自动或手动加载到迁移项目之后，DMS 架构转换将不再使用延迟加载。因此，当您更改数据库中的对象（如表和过程）时，请务必在迁移项目中刷新它们。

要从数据库刷新架构，可选择要刷新的对象，然后从操作中选择从数据库刷新。您可以刷新源数据库和目标数据库架构中的数据库对象：

- 源 – 如果更新源数据库架构，那么可选择从数据库刷新，用源数据库中的最新架构替代项目中的架构。
- 目标 – 如果更新目标数据库的架构，那么 DMS 架构转换将用目标数据库中的最新架构替代项目中的架构。DMS 架构转换会用目标数据库中的代码替代转换后的代码。选择从数据库刷新之前，请确保已将转换后的代码应用于目标数据库。否则，请再次转换源数据库架构。



## 在 DMS 架构转换中保存和应用转换后的代码

DMS 架构转换在转换源数据库对象之后，不会立即将转换后的代码应用于目标数据库。相反，DMS 会将转换后的代码存储在项目中，直到您准备好将其应用到目标数据库。

在应用转换后的代码之前，您可以更新源数据库代码，并再次转换更新的对象，以处理现有的操作项。有关 DMS 架构转换无法自动转换的项的更多信息，请参阅[使用 DMS 架构转换创建数据库迁移评测报告](#)。有关在 DMS 架构转换的迁移项目中刷新源数据库对象的更多信息，请参阅[刷新数据库架构](#)。

您可以将代码作为 SQL 脚本保存到文件中，而不是在 DMS 架构转换中将转换后的代码直接应用于数据库。您可以查看这些 SQL 脚本，必要时进行编辑，然后手动将这些 SQL 脚本应用于目标数据库。

### 将转换后的代码保存为 SQL 文件

您可以将转换后的架构作为 SQL 脚本保存到文本文件中。您可以修改转换后的代码，以处理 DMS 架构转换无法自动转换的操作项。然后，可以在目标数据库上运行更新的 SQL 脚本，将转换后的架构应用于目标数据库。

### 将转换后的架构保存为 SQL 脚本

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后选择架构转换。
4. 选择启动架构转换。将打开架构转换页面。
5. 在右侧窗格中，选择目标数据库架构，或选择要保存的转换后的对象。请确保 DMS 架构转换用蓝色突出显示父节点名称，并为目标数据库激活操作菜单。
6. 对于操作，选择另存为 SQL。将显示保存对话框。
7. 选择另存为 SQL 以确认选择。

DMS 架构转换使用 SQL 文件创建存档，并将该存档存储在 Amazon S3 存储桶中。

8. (可选) 在实例配置文件中编辑架构转换设置，为存档更改 S3 存储桶。
9. 从 S3 存储桶打开 SQL 脚本。

### 应用转换后的代码

在您准备好将转换后的代码应用于目标数据库后，请从项目的右侧窗格中选择数据库对象。您可以将更改应用于整个数据库架构或选定的数据库架构对象。

选择数据库对象之后，DMS 架构转换用蓝色突出显示所选节点或父节点的名称。然后，激活操作菜单。对于操作，选择应用更改。在显示的对话框中，选择应用以确认选择，将转换后的代码应用于目标数据库。

## 应用扩展包架构

首次将转换后的架构应用于目标数据库时，DMS 架构转换也可能应用扩展包架构。扩展包架构可模拟源数据库的系统功能，这些功能是在目标数据库上运行转换后的代码所必需的。如果转换后的代码使用扩展包的功能，请确保应用扩展包架构。

要将扩展包手动应用于目标数据库，可在操作中选择应用更改。在显示的对话框中，选择确认以将扩展包应用于目标数据库。

我们建议您不要修改扩展包架构，以免转换后的代码出现意外结果。

有关更多信息，请参阅 [在 DMS 架构转换中使用扩展包](#)。

## 在 DMS 架构转换中使用扩展包

DMS 架构转换中的扩展包是一个附加模块，用于模拟目标数据库不支持的源数据库函数。使用扩展包可确保转换后的代码产生与源代码相同的结果。您需要先转换数据库架构，然后才能安装扩展包。

每个扩展包都包含一个数据库架构。此架构包括 SQL 函数、过程、表和视图，用于模拟源数据库中特定的联机事务处理 (OLTP) 对象或不受支持的内置函数。

在转换源数据库时，DMS 架构转换会向目标数据库添加一个额外的架构。此架构用于实现源数据库的 SQL 系统功能，这些功能是在目标数据库上运行转换后的代码所必需的。这个额外的架构称为扩展包架构。

扩展包架构按照源数据库命名，如下所示：

- Microsoft SQL Server – aws\_sqlserver\_ext
- Oracle – aws\_oracle\_ext

可以通过以下两种方式应用扩展包：

- 当您应用转换后的代码时，DMS 架构转换可以自动应用扩展包。DMS 架构转换先应用扩展包，然后再应用所有其他架构对象。
- 您可以手动应用扩展包。为此，请在目标数据库树中选择扩展包架构，然后依次选择应用和应用扩展包。

# 使用以下命令将数据库迁移到其 Amazon RDS 等效数据库中 AWS DMS

AWS Database Migration Service (AWS DMS) 中的同构数据迁移简化了自行管理的本地数据库向亚马逊关系数据库服务 (Amazon RDS) 等效数据库的迁移。例如，您可以使用同构数据迁移，将本地 PostgreSQL 数据库迁移到 Amazon RDS for PostgreSQL 或 Aurora PostgreSQL。对于同构数据迁移，AWS DMS 使用本机数据库工具提供简单且高性能 like-to-like 的迁移。

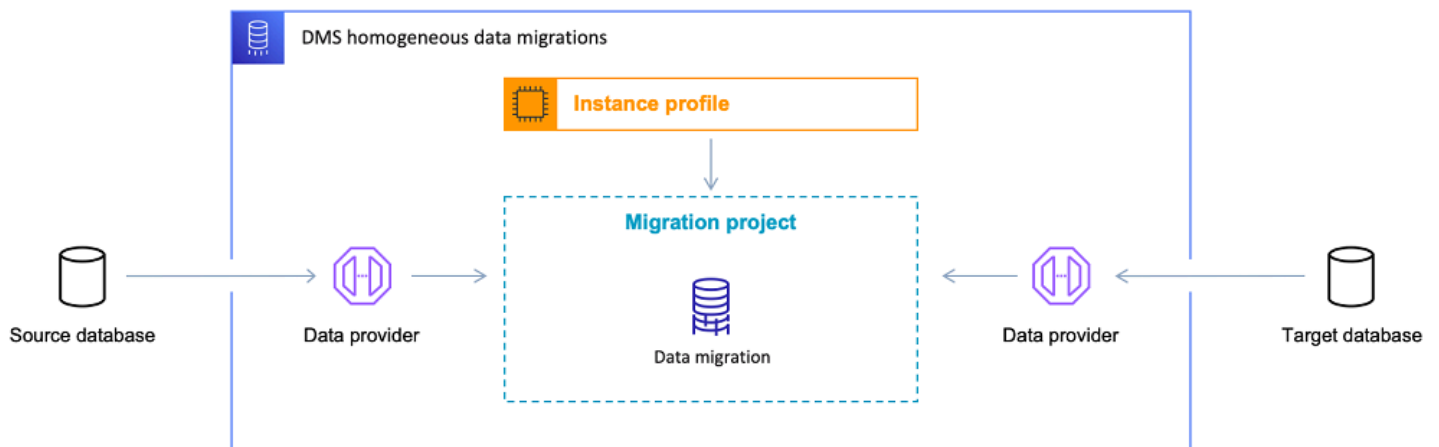
同构数据迁移是无服务器的，这意味着它 AWS DMS 会自动扩展迁移所需的资源。通过同构数据迁移，您可以迁移数据、表分区、数据类型和辅助对象，例如函数、存储过程等。

总体而言，同构数据迁移使用实例配置文件、数据提供程序和迁移项目来操作。当您使用相同类型的兼容源和目标数据提供程序创建迁移项目时，将在其中 AWS DMS 部署一个运行数据迁移的无服务器环境。接下来，AWS DMS 连接到源数据提供器，读取源数据，将文件转储到磁盘上，然后使用本机数据库工具恢复数据。有关实例配置文件、数据提供器和迁移项目的更多信息，请参阅[与数据提供者、实例配置文件和迁移项目合作 AWS DMS](#)。

有关支持的源数据库的列表，请参阅 [DMS 同构数据迁移的源](#)。

有关支持的目标数据库列表，请参阅 [DMS 同构数据迁移的目标](#)。

下图说明同构数据迁移的工作方式。



以下各节提供了有关使用同构数据迁移的信息。

## 主题

- [支持的 AWS 区域](#)
- [功能](#)

- [同构数据迁移限制](#)
- [AWS DMS 中的同构数据迁移过程概述](#)
- [在 AWS DMS 中设置同构数据迁移](#)
- [为同构数据迁移创建源数据提供器 AWS DMS](#)
- [为同构数据迁移创建目标数据提供器 AWS DMS](#)
- [在中运行同构数据迁移 AWS DMS](#)
- [对 AWS DMS 中的同构数据迁移进行故障排除](#)

## 支持的 AWS 区域

你可以在下面运行同构数据迁移。AWS 区域

区域名称	区域
美国东部 ( 弗吉尼亚州北部 )	us-east-1
美国东部 ( 俄亥俄州 )	us-east-2
美国西部 ( 俄勒冈州 )	us-west-2
亚太地区 ( 东京 )	ap-northeast-1
亚太地区 ( 新加坡 )	ap-southeast-1
亚太地区 ( 悉尼 )	ap-southeast-2
欧洲地区 ( 法兰克福 )	eu-central-1
欧洲地区 ( 斯德哥尔摩 )	eu-north-1
欧洲地区 ( 爱尔兰 )	eu-west-1

## 功能

同构数据迁移提供以下功能：

- AWS DMS 自动管理同构数据迁移所需的计算和存储资源。AWS Cloud AWS DMS 当您开始数据迁移时，会在无服务器环境中部署这些资源。
- AWS DMS 使用本机数据库工具在相同类型的数据库之间启动全自动迁移。
- 您可以使用同构数据迁移来迁移您的数据和辅助对象，例如分区、函数、存储过程等。
- 您可以在以下三种迁移模式下运行同构数据迁移：完全加载、持续复制以及带持续复制的完全加载。
- 对于同构数据迁移，您可以使用本地数据库、Amazon EC2、Amazon RDS 数据库作为源。您可以选择 Amazon RDS 或 Amazon Aurora 作为同构数据迁移的迁移目标。

## 同构数据迁移限制

使用同构数据迁移时存在以下限制：

- 同构数据迁移仅支持 MongoDB 和 Amazon DocumentDB 迁移的选择规则。DMS 不支持其他数据库引擎的选择规则。此外，您无法设置转换规则来更改列的对象类型、将对象从一个架构移动到另一个架构或者更改对象的名称。
- 同构数据迁移不提供用于数据验证的内置工具。
- 在 PostgreSQL 中使用同构数据迁移时，会将视图作为表 AWS DMS 迁移到目标数据库。
- 在持续数据复制过程中，同构数据迁移不捕获架构级别的更改。如果您在源数据库中创建了一个新表，则 AWS DMS 无法迁移此表。要迁移此新表，请重新启动数据迁移。
- 您不能在中使用同构数据迁移将数据从较高的数据库版本迁移 AWS DMS 到较低的数据库版本。
- 您不能在 CLI 或 API 中使用同构数据迁移。
- 同构数据迁移不支持与 VPC 辅助 CIDR 范围中的数据库实例建立连接。
- 您不能使用 8081 端口从数据提供程序进行同构迁移。
- 同构数据迁移不支持迁移加密的 MySQL 数据库和表。

## AWS DMS 中的同构数据迁移过程概述

您可以使用 AWS DMS 中的同构数据迁移，在两个相同类型的数据库之间迁移数据。可使用以下工作流程创建和运行数据迁移。

1. 创建所需的 AWS Identity and Access Management (IAM) 策略和角色。有关更多信息，请参阅[创建 IAM 资源](#)。
2. 配置源数据库和目标数据库，并创建具有 AWS DMS 中同构数据迁移所需的最低权限的数据库用户。有关更多信息，请参阅[创建源数据提供程序](#)和[创建目标数据提供程序](#)。

3. 将源数据库和目标数据库凭证存储在 AWS Secrets Manager 中。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的[步骤 1：创建密钥](#)。
4. 在 AWS DMS 控制台中创建子网组、实例配置文件和数据提供程序。有关更多信息，请参阅[创建子网组](#)、[创建实例配置文件](#)和 [创建实例提供程序](#)。
5. 使用在上一步中创建的资源创建迁移项目。有关更多信息，请参阅 [创建迁移项目](#)。
6. 创建、配置和启动数据迁移。有关更多信息，请参阅[创建数据迁移](#)。
7. 完成完全加载或正在进行的复制后，可以切换为开始使用新的目标数据库。
8. 清除资源。在您完成迁移后，Amazon 将在三天内终止迁移项目中的数据迁移。但是，您需要手动删除诸如实例配置文件、数据提供程序、IAM policy 和角色，以及 AWS Secrets Manager 中的密钥等资源。

有关 AWS DMS 中同构数据迁移的更多信息，请阅读[从 PostgreSQL 到 Amazon RDS for PostgreSQL 的迁移](#)分步迁移指南。

[本视频](#)介绍了 AWS DMS 中的同构数据迁移用户界面，并协助您熟悉此功能。

## 在 AWS DMS 中设置同构数据迁移

要在 AWS DMS 中设置同构数据迁移，请完成以下先决任务。

### 主题

- [为 AWS DMS 中的同构数据迁移创建所需的 IAM 资源](#)
- [为 AWS DMS 中的同构数据迁移设置网络](#)

## 为 AWS DMS 中的同构数据迁移创建所需的 IAM 资源

要运行同构数据迁移，必须在账户中创建 IAM policy 和 IAM 角色，才能与其他 AWS 服务进行交互。在本节中，您将创建这些必需的 IAM 资源。

### 主题

- [为 AWS DMS 中的同构数据迁移创建 IAM policy](#)
- [为 AWS DMS 中的同构数据迁移创建 IAM 角色](#)

## 为 AWS DMS 中的同构数据迁移创建 IAM policy

为了访问数据库并迁移数据，AWS DMS 会为同构数据迁移创建无服务器环境。在此环境中，AWS DMS 需要访问 VPC 对等连接、路由表、安全组和其他 AWS 资源。此外，AWS DMS 将每个数据迁移的日志、指标和进度存储在 Amazon CloudWatch 中。要创建数据迁移项目，AWS DMS 需要访问这些服务。

在此步骤中，创建 IAM policy，向 AWS DMS 提供 Amazon EC2 和 CloudWatch 资源的访问权限。接下来，创建 IAM 角色并附加此策略。

### 为 AWS DMS 中的同构数据迁移创建 IAM policy

1. 登录到 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择策略。
3. 选择创建策略。
4. 在创建策略页面中，选择 JSON 选项卡。
5. 将下面的 JSON 粘贴到编辑器中。

```
{
  "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. 选择下一步：标签和下一步：审核。
7. 在名称\*中输入 **HomogeneousDataMigrationsPolicy**，然后选择创建策略。

## 为 AWS DMS 中的同构数据迁移创建 IAM 角色

在此步骤中，创建 IAM 角色，向 AWS DMS 提供访问 AWS Secrets Manager、Amazon EC2 和 CloudWatch 的权限。

## 为 AWS DMS 中的同构数据迁移创建 IAM 角色

1. 登录到 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择角色。
3. 选择创建角色。
4. 在选择可信实体页面中，在可信实体类型下选择 AWS 服务。对于其他 AWS 服务的应用场景，选择 DMS。
5. 选中 DMS 复选框，然后选择下一步。
6. 在添加权限页面上，选择之前创建的 HomogeneousDataMigrationsPolicy。同时选择 SecretsManagerReadWrite。选择下一步。
7. 在命名、检查并创建页面上，在角色名称中输入 **HomogeneousDataMigrationsRole**，然后选择创建角色。
8. 在角色页面上，在角色名称中输入 **HomogeneousDataMigrationsRole**。选择 HomogeneousDataMigrationsRole。
9. 在 HomogeneousDataMigrationsRole 页面上，选择信任关系选项卡。选择编辑信任策略。
10. 在编辑信任策略页面上，将以下 JSON 粘贴到编辑器中，替换现有文本。

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

在以上示例中，请用您的 AWS 区域名称替换 *your\_region*。

上述基于资源的策略向 AWS DMS 服务主体提供了根据 AWS 托管的 SecretsManagerReadWrite 策略及客户托管的 HomogeneousDataMigrationsPolicy 策略执行任务的权限。

## 11. 选择更新策略。

## 为 AWS DMS 中的同构数据迁移设置网络

AWS DMS 会在基于 Amazon VPC 服务的虚拟私有云 (VPC) 中为同构数据迁移创建无服务器环境。在创建实例配置文件时，可指定要使用的 VPC。可以使用账户和 AWS 区域的默认 VPC，也可以创建新的 VPC。

对于每个数据迁移，AWS DMS 会与用于实例配置文件的 VPC 建立 VPC 对等连接。接下来，AWS DMS 在与实例配置文件关联的安全组中添加 CIDR 块。由于 AWS DMS 会将公有 IP 地址附加到实例配置文件，因此使用同一实例配置文件的所有数据迁移都具有相同的公有 IP 地址。如果数据迁移停止或失败，AWS DMS 会删除 VPC 对等连接。

为避免 CIDR 块与实例配置文件 VPC 的 VPC 重叠，AWS DMS 使用 /24 前缀以区别于以下 CIDR 块之一：10.0.0.0/8、172.16.0.0/12 和 192.168.0.0/16。例如，如果您并行运行三个数据迁移，则 AWS DMS 使用以下 CIDR 块建立 VPC 对等连接。

- 192.168.0.0/24 – 用于第一个数据迁移
- 192.168.1.0/24 – 用于第二个数据迁移
- 192.168.2.0/24 – 用于第三个数据迁移

您可以使用不同的网络配置，设置源数据库与目标数据库之间通过 AWS DMS 进行的交互。此外，要进行持续的数据复制，必须设置源数据库与目标数据库之间的交互。这些配置取决于源数据提供程序的位置以及网络设置。以下各节描述了常见的网络配置。

### 主题

- [为源和目标数据提供程序使用单个 VPC](#)
- [为源和目标数据提供程序使用不同 VPC](#)
- [使用本地源数据提供程序](#)
- [配置持续的数据复制](#)

## 为源和目标数据提供程序使用单个 VPC

在此配置中，AWS DMS 在专用网络内连接源和目标数据提供程序。

配置当源和目标数据提供程序位于同一 VPC 中的网络

1. 使用源和目标数据提供程序所用的 VPC 和子网，在 AWS DMS 控制台中创建子网组。有关更多信息，请参阅[创建子网组](#)。
2. 使用 VPC 和刚创建的子网组，在 AWS DMS 控制台中创建实例配置文件。此外，选择源和目标数据提供程序所用的 VPC 安全组。有关更多信息，请参阅[创建实例配置文件](#)。

此配置不需要使用公有 IP 地址进行数据迁移。

## 为源和目标数据提供程序使用不同 VPC

在此配置中，AWS DMS 使用专用网络连接源和目标数据提供程序。对于其他数据提供程序，AWS DMS 使用公共网络。根据与实例配置文件位于同一 VPC 中的数据提供程序，选择以下配置之一。

为源数据提供程序配置专用网络，为目标数据提供程序配置公共网络

1. 使用源数据提供程序所用的 VPC 和子网，在 AWS DMS 控制台中创建子网组。有关更多信息，请参阅[创建子网组](#)。
2. 使用 VPC 和刚创建的子网组，在 AWS DMS 控制台中创建实例配置文件。此外，选择源数据提供程序所用的 VPC 安全组。有关更多信息，请参阅[创建实例配置文件](#)。
3. 打开迁移项目。在数据迁移选项卡上，选择数据迁移。记下详细信息选项卡上连接和安全下的公共 IP 地址。
4. 允许从目标数据库安全组中的数据迁移的公共 IP 地址进行访问。有关更多信息，请参阅《Amazon Relational Database Service 用户指南》中的[使用安全组控制访问](#)。

为源数据提供程序配置公共网络，为目标数据提供程序配置专用网络

1. 使用目标数据提供程序所用的 VPC 和子网，在 AWS DMS 控制台中创建子网组。有关更多信息，请参阅[创建子网组](#)。
2. 使用 VPC 和刚创建的子网组，在 AWS DMS 控制台中创建实例配置文件。此外，选择目标数据提供程序所用的 VPC 安全组。有关更多信息，请参阅[创建实例配置文件](#)。
3. 打开迁移项目。在数据迁移选项卡上，选择数据迁移。记下详细信息选项卡上连接和安全下的公共 IP 地址。

4. 允许从源数据库安全组中的数据迁移的公共 IP 地址进行访问。有关更多信息，请参阅《Amazon Relational Database Service 用户指南》中的[使用安全组控制访问](#)。

## 使用本地源数据提供程序

在此配置中，AWS DMS 在公共网络中连接源数据提供程序。AWS DMS 使用专用网络连接目标数据提供程序。

### 为源本地数据提供程序配置网络

1. 使用目标数据提供程序所用的 VPC 和子网，在 AWS DMS 控制台中创建子网组。有关更多信息，请参阅[创建子网组](#)。
2. 使用 VPC 和刚创建的子网组，在 AWS DMS 控制台中创建实例配置文件。此外，选择目标数据提供程序所用的 VPC 安全组。有关更多信息，请参阅[创建实例配置文件](#)。
3. 打开迁移项目。在数据迁移选项卡上，选择数据迁移。记下详细信息选项卡上连接和安全下的公共 IP 地址。
4. 允许从 AWS DMS 中数据迁移的公共 IP 地址访问源数据库。

AWS DMS 会在 VPC 安全组中创建入站或出站规则。请勿删除这些规则，因为此操作可能会导致数据迁移失败。您可以在 VPC 安全组中配置自己的规则。我们建议您为规则添加描述，以便管理规则。

## 配置持续的数据复制

要运行完全加载和更改数据捕获 (CDC) 或更改数据捕获 (CDC) 类型的数据迁移，必须允许源数据库与目标数据库之间的连接。

### 配置可公开访问的源数据库与目标数据库之间的连接

1. 记下源数据库和目标数据库的公共 IP 地址。
2. 允许从目标数据库的公共 IP 地址访问源数据库。
3. 允许从源数据库的公共 IP 地址访问目标数据库。

### 配置位于单个 VPC 中可私密访问的源数据库与目标数据库之间的连接

1. 记下源数据库和目标数据库的专用 IP 地址。

**⚠ Important**

如果源数据库和目标数据库位于不同 VPC 或不同网络中，那么源数据库和目标数据库只能使用公共 IP 地址。您只能在数据提供程序中使用公共主机名或 IP 地址。

2. 允许从目标数据库的专用 IP 地址访问源数据库。
3. 允许从源数据库的专用 IP 地址访问目标数据库。

## 为同构数据迁移创建源数据提供器 AWS DMS

您可以使用兼容 MySQL、PostgreSQL 和 MongoDB 兼容的数据库作为输入的源数据提供器。[同构数据迁移 AWS DMS](#)

有关支持的数据库版本，请参见[DMS 同构数据迁移的源数据提供程序](#)。

源数据提供程序可以是本地数据库、Amazon EC2 数据库或 Amazon RDS 数据库。

### 主题

- [使用与 MySQL 兼容的数据库作为同构数据迁移的来源 AWS DMS](#)
- [使用 PostgreSQL 数据库作为同构数据迁移的来源 AWS DMS](#)
- [使用与 MongoDB 兼容的数据库作为同构数据迁移的来源 AWS DMS](#)

## 使用与 MySQL 兼容的数据库作为同构数据迁移的来源 AWS DMS

可以使用与 MySQL 兼容的数据库（MySQL 或 MariaDB），作为 AWS DMS 中[同构数据迁移](#)的源。在此情况下，源数据提供程序可以是本地数据库、Amazon EC2 数据库、RDS for MySQL 数据库或 MariaDB 数据库。

要运行同构数据迁移，必须使用对要复制的所有源表和二级对象具有 SELECT 权限的数据库用户。对于更改数据捕获（CDC）任务，此用户还必须具备（REPLICATION CLIENT 对于高于 10.5.2 的 MariaDB 版本为 BINLOG MONITOR）和 REPLICATION SLAVE 权限。对于完全加载数据迁移，不需要这两个权限。

可使用以下脚本，在 MySQL 数据库中创建具所需权限的数据库用户。对您迁移到的所有数据库运行 GRANT 查询 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'@'%';
```

在以上示例中，请用您自己的信息替换每个#####。如果源 MySQL 数据库版本低于 8.0，则可以跳过 GRANT BACKUP\_ADMIN 命令。

可使用以下脚本，在 MariaDB 数据库中创建具所需权限的数据库用户。对您迁移到的所有数据库运行 GRANT 查询 AWS。

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT, RELOAD, LOCK TABLES, REPLICATION SLAVE, BINLOG MONITOR, SHOW VIEW ON *.*
TO 'your_user'@'%';
```

在以上示例中，请用您自己的信息替换每个#####。

以下各节描述了自管理 MySQL 数据库和 AWS 托管 MySQL 数据库的特定配置先决条件。

## 主题

- [使用自管理的 MySQL 兼容数据库作为同构数据迁移的源](#)
- [使用 AWS 托管的 MySQL 兼容数据库作为同构数据迁移的来源 AWS DMS](#)
- [使用与 MySQL 兼容的数据库作为同构数据迁移的源的限制](#)

## 使用自管理的 MySQL 兼容数据库作为同构数据迁移的源

本节介绍如何配置本地托管的或在 Amazon EC2 实例上托管的与 MySQL 兼容的数据库。

选择源 MySQL 或 MariaDB 数据库的版本。请确保 AWS DMS 它支持您的源 MySQL 或 MariaDB 数据库版本，如中所述。[DMS 同构数据迁移的源](#)

要使用 CDC，请确保启用二进制日志记录。要启用二进制日志记录，请在 MySQL 或 MariaDB 数据库的 my.ini ( Windows ) 或 my.cnf ( UNIX ) 文件中配置以下参数。

参数	值
server-id	将该参数设置为 1 或更大的值。

参数	值
log-bin	设置二进制日志文件的路径，例如 log-bin=E:\MySQL_Logs\BinLog。请勿包含文件扩展名。
binlog_format	将该参数设置为 ROW。我们建议在复制期间使用此设置，因为在某些情况下，如果 binlog_format 设置为 STATEMENT，那么在将数据复制到目标时可能会导致不一致。如果 binlog_format 设置为 MIXED，数据库引擎还会向目标写入类似的非一致数据，因为数据库引擎会自动切换到基于 STATEMENT 的日志记录。
expire_logs_days	将该参数设置为 1 或更大的值。为防止过度使用磁盘空间，我们建议您不要使用默认值 0。
binlog_checksum	将该参数设置为 NONE。
binlog_row_image	将该参数设置为 FULL。
log_slave_updates	如果您使用 MySQL 或 MariaDB 副本作为源，请将此参数设置为 TRUE。

## 使用 AWS 托管的 MySQL 兼容数据库作为同构数据迁移的来源 AWS DMS

本节介绍如何配置 Amazon RDS for MySQL 和 Amazon RDS for MariaDB 数据库实例。

当您使用 AWS 托管的 MySQL 或 MariaDB 数据库作为 AWS DMS 中同构数据迁移的源时，请确保您满足以下条件才能实现 CDC：

- 要为 RDS for MySQL 和 RDS for MariaDB 启用二进制日志，请在实例级别启用自动备份。要为 Aurora MySQL 集群启用二进制日志，请更改参数组中的变量 binlog\_format。无需为 Aurora MySQL 集群启用自动备份。

接下来，将 binlog\_format 参数设置为 ROW。

有关设置自动备份的更多信息，请参阅《Amazon RDS 用户指南》中的[启用自动备份](#)。

有关为 Amazon RDS for MySQL 或 MariaDB 数据库设置二进制日志记录的更多信息，请参阅《Amazon RDS 用户指南》中的[设置二进制日志记录格式](#)。



有关为 Aurora MySQL 集群设置二进制日志记录的更多信息，请参阅[如何为 Amazon Aurora MySQL 集群开启二进制日志记录？](#)。

- 确保二进制日志可供使用 AWS DMS。由于 AWS 由托管的 MySQL 和 MariaDB 数据库会尽快清除二进制日志，因此您应该延长日志的可用时间。例如，要将日志保留时间增加到 24 小时，您应运行以下命令。

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- 将 `binlog_row_image` 参数设置为 Full。
- 将 `binlog_checksum` 参数设置为 NONE。
- 如果使用 Amazon RDS MySQL 或 MariaDB 副本作为源，请在只读副本上启用备份，并确保 `log_slave_updates` 参数设置为 TRUE。

## 使用与 MySQL 兼容的数据库作为同构数据迁移的源的限制

使用与 MySQL 兼容的数据库作为同构数据迁移的源时，存在以下限制：

- 同构迁移任务不支持诸如序列之类的 MariaDB 对象。
- 由于对象差异不兼容，从 MariaDB 迁移到 Amazon RDS MySQL/Aurora MySQL 可能会失败。
- 用于连接到数据源的用户名具有以下限制：
  - 长度为 2 到 64 个字符。
  - 不能包含空格。
  - 可以包含以下字符：a-z、A-Z、0-9、下划线 (`_`)。
  - 必须以 a-z 或 A-Z 开头。
- 用于连接到数据源的密码具有以下限制：
  - 长度为 1 到 128 个字符。
  - 不能包含以下任何字符：单引号 (`'`)、双引号 (`"`)、分号 (`;`) 或空格。

## 使用 PostgreSQL 数据库作为同构数据迁移的来源 AWS DMS

可以使用 PostgreSQL 数据库作为 AWS DMS 中[同构数据迁移](#)的源。在此情况下，源数据提供程序可以是本地数据库、Amazon EC2 数据库或 RDS for PostgreSQL 数据库。

要运行同构数据迁移，请为您在 PostgreSQL 源数据库中 AWS DMS 指定的数据库用户授予超级用户权限。该数据库用户需要超级用户权限，才能访问源中特定于复制的函数。对于完全加载数据迁移，数据库用户需要对表具有 SELECT 权限才能迁移表。

可使用以下脚本，在 PostgreSQL 源数据库中创建具所需权限的数据库用户。对您迁移到的所有数据库运行 GRANT 查询 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;
```

在以上示例中，请用您自己的信息替换每个#####。

以下各节描述了自我管理 PostgreSQL 数据库和 AWS 托管 PostgreSQL 数据库的特定配置先决条件。

## 主题

- [使用自我管理的 PostgreSQL 数据库作为同构数据迁移的来源 AWS DMS](#)
- [使用 AWS 托管的 PostgreSQL 数据库作为同构数据迁移的来源 AWS DMS](#)
- [使用与 PostgreSQL 兼容的数据库作为同构数据迁移的源的限制](#)

## 使用自我管理的 PostgreSQL 数据库作为同构数据迁移的来源 AWS DMS

本节介绍如何配置本地托管的或在 Amazon EC2 实例上托管的 PostgreSQL 数据库。

检查源 PostgreSQL 数据库的版本。请确保它 AWS DMS 支持您的源 PostgreSQL 数据库版本，如中所述。[DMS 同构数据迁移的源](#)

同构数据迁移支持使用逻辑复制进行更改数据捕获 ( CDC )。要开启自我管理 PostgreSQL 源数据库的逻辑复制，请在 `postgresql.conf` 配置文件中设置以下参数和值：

- 将 `wal_level` 设置为 `logical`。
- 将 `max_replication_slots` 设置为一个大于 1 的值。

请根据您需要运行的任务数设置 `max_replication_slots` 值。例如，要运行五个任务，您需要设置至少五个时隙。在任务启动时时隙立即打开，即使任务不再运行仍保持打开。请务必手动删除空闲时隙。

- 将 `max_wal_senders` 设置为一个大于 1 的值。

`max_wal_senders` 参数用于设置可以运行的并发任务数。

- `wal_sender_timeout` 参数会终止处于不活动状态的时间超过指定毫秒数的复制连接。默认值为 60000 毫秒 (60 秒)。将此值设置为 0 (零) 会禁用超时机制，这是 DMS 的有效设置。

有些参数是静态的，只能在服务器启动时设置。在服务器重新启动之前，对配置文件条目所做的任何更改都将被忽略。有关更多信息，请参阅 [PostgreSQL 文档](#)。

## 使用 AWS 托管的 PostgreSQL 数据库作为同构数据迁移的来源 AWS DMS

本节介绍如何配置 Amazon RDS for PostgreSQL 数据库实例。

使用 PostgreSQL 数据库实例 AWS 的主用户账户作为 PostgreSQL 源数据提供程序的用户账户，以便在中进行同构数据迁移。AWS DMS 主用户账户具有允许它设置 CDC 所需的角色。如果使用主用户账户之外的账户，该账户必须具有 `rds_superuser` 角色和 `rds_replication` 角色。`rds_replication` 角色可授予管理逻辑时隙和使用逻辑时隙流式处理数据的权限。

可使用以下代码示例授予 `rds_superuser` 和 `rds_replication` 角色。

```
GRANT rds_superuser to your_user;  
GRANT rds_replication to your_user;
```

在以上示例中，请用您的数据库用户名称替换 `your_user`。

要开启逻辑复制，请将数据库参数组中的 `rds.logical_replication` 参数设置为 1。此静态参数要求重启数据库实例才能生效。

## 使用与 PostgreSQL 兼容的数据库作为同构数据迁移的源的限制

使用与 PostgreSQL 兼容的数据库作为同构数据迁移的源时，存在以下限制：

- 用于连接到数据源的用户名具有以下限制：
  - 长度为 2 到 64 个字符。
  - 不能包含空格。
  - 可以包含以下字符：a-z、A-Z、0-9、下划线 (`_`)。
  - 必须以 a-z 或 A-Z 开头。
- 用于连接到数据源的密码具有以下限制：
  - 长度为 1 到 128 个字符。

- 不能包含以下任何字符：单引号 (')、双引号 (")、分号 (;) 或空格。

## 使用与 MongoDB 兼容的数据库作为同构数据迁移的来源 AWS DMS

您可以使用与 MongoDB 兼容的数据库作为同构数据迁移的来源。AWS DMS 在这种情况下，您的源数据提供者可以是本地、适用于 MongoDB 的 Amazon EC2 数据库或亚马逊 DocumentDB (兼容 MongoDB) 数据库。

有关支持的数据库版本，请参见[DMS 同构数据迁移的源数据提供程序](#)。

以下各节描述了自行管理 MongoDB 数据库和托管的 Amazon DocumentDB 数据库的特定配置先决条件。AWS

### 主题

- [使用自我管理的 MongoDB 数据库作为同构数据迁移的来源 AWS DMS](#)
- [使用 Amazon DocumentDB 数据库作为同构数据迁移的来源 AWS DMS](#)
- [使用兼容 MongoDB 的数据库作为同构数据迁移源的功能](#)
- [使用兼容 MongoDB 的数据库作为同构数据迁移源的限制](#)
- [使用兼容 MongoDB 的数据库作为同构数据迁移源的最佳实践](#)

## 使用自我管理的 MongoDB 数据库作为同构数据迁移的来源 AWS DMS

本节介绍如何配置托管在本地或 Amazon EC2 实例上的 MongoDB 数据库。

检查您的源 MongoDB 数据库的版本。请确保它 AWS DMS 支持您的源 MongoDB 数据库版本，如中所述。[DMS 同构数据迁移的源数据提供程序](#)

要使用 MongoDB 源运行同构数据迁移，您可以创建一个具有 root 权限的用户帐户，也可以创建一个仅具有数据库迁移权限的用户。有关用户创建的更多信息，请参阅[将 MongoDB 作为 AWS DMS 的源时所需的权限](#)。

要将持续复制或 CDC 与 MongoDB 配合使用 AWS DMS，需要访问 MongoDB 操作日志 (oplog)。有关更多信息，请参阅[为 CDC 配置 MongoDB 副本集](#)。

有关 MongoDB 身份验证方法的信息，请参阅[将 MongoDB 作为 AWS DMS 源时的安全要求](#)

对于 MongoDB 作为数据源，同构数据迁移支持 Amazon DocumentDB 支持的所有数据类型。

要将 MongoDB 作为来源，要将用户凭据存储在 Secrets Manager 中，您需要使用其他类型的密钥类型以纯文本形式提供这些凭据。有关更多信息，请参阅[使用密钥访问 AWS Database Migration Service 端点](#)。

以下代码示例演示了如何使用纯文本存储数据库密钥。

```
{
  "username": "dbuser",
  "password": "dbpassword"
}
```

## 使用 Amazon DocumentDB 数据库作为同构数据迁移的来源 AWS DMS

本节介绍如何配置您的 Amazon DocumentDB 数据库实例以用作同构数据迁移的来源。

使用 Amazon DocumentDB 实例的主用户名作为兼容 MongoDB 的源数据提供程序的用户账户，以便在中进行同构数据迁移。AWS DMS 主用户账户具有允许它设置 CDC 所需的角色。如果您使用主用户账户以外的账户，则该账户必须具有根角色。有关以 root 账户创建用户的更多信息，请参阅[设置权限以使用 Amazon DocumentDB 作为源](#)。

要开启逻辑复制，请将数据库 `change_stream_log_retention_duration` 参数组中的参数设置为适合您的事务工作负载的设置。更改此静态参数需要您重启数据库实例才能生效。在开始迁移包括仅限满载在内的所有任务类型的数据之前，请为给定数据库中的所有馆藏启用 Amazon DocumentDB 更改流，或者仅为选定的馆藏启用 Amazon DocumentDB 更改流。有关为 Amazon DocumentDB 启用更改流的更多信息，请参阅亚马逊 Document DB 开发者指南中的[启用更改流](#)。

### Note

AWS DMS 使用 Amazon DocumentDB 更改流捕获正在进行的复制期间的更改。如果 Amazon DocumentDB 在 DMS 读取记录之前将其从更改流中清空，则您的任务将失败。我们建议将 `change_stream_log_retention_duration` 参数设置为将更改保留至少 24 小时。

要使用 Amazon DocumentDB 进行同构数据迁移，请将用户凭证存储在 Secrets Manager 中的 Amazon DocumentDB 数据库凭证下。

## 使用兼容 MongoDB 的数据库作为同构数据迁移源的功能

- 在满载阶段，您可以迁移 Amazon DocumentDB 支持的所有二级索引。

- AWS DMS 并行迁移集合。同构数据迁移根据集合中每个文档的平均大小在运行时计算数据段，以实现最佳性能。
- DMS 可以复制您在 CDC 阶段创建的二级索引。在 MongoDB 6.0 版本中，DMS 支持此功能。
- DMS 支持嵌套级别大于 97 的文档。

## 使用兼容 MongoDB 的数据库作为同构数据迁移源的限制

- 文档不能有带\$前缀的字段名。
- AWS DMS 不支持时间序列集合迁移。
- AWS DMS 在 CDC 阶段不支持createdrop、或 rename collection DDL 事件。
- AWS DMS 不支持字段集合中不一致的数据类型。\_id例如，以下不支持的集合具有该\_id字段的多种数据类型。

```
rs0 [direct: primary] test> db.collection1.aggregate([
...   {
...     $group: {
...       _id: { $type: "$_id" },
...       count: { $sum: 1 }
...     }
...   }
... ])
[ { _id: 'string', count: 6136 }, { _id: 'objectId', count: 848033 } ]
```

- 对于仅限 CDC 的任务，AWS DMS 仅支持immediate启动模式。
- AWS DMS 不支持包含无效 UTF8 字符的文档。
- AWS DMS 不支持分片集合。

## 使用兼容 MongoDB 的数据库作为同构数据迁移源的最佳实践

- 对于托管在同一 MongoDB 实例上的多个大型数据库和集合，我们建议您使用每个数据库和集合的选择规则在多个数据迁移任务和项目之间分配任务。您可以调整数据库和集合部门以获得最佳性能。

## 为同构数据迁移创建目标数据提供者 AWS DMS

您可以使用兼容 MySQL、PostgreSQL 和 Amazon DocumentDB 的数据库作为目标数据提供者，在中进行同构数据迁移。AWS DMS

有关支持的数据库版本，请参见[DMS 同构数据迁移的目标数据提供程序](#)。

目标数据提供程序可以是 Amazon RDS 数据库实例或 Amazon Aurora 数据库集群。请注意，目标数据提供程序的数据库版本必须等于或高于源数据提供程序的数据库版本。

## 主题

- [使用与 MySQL 兼容的数据库作为同构数据迁移的目标 AWS DMS](#)
- [使用 PostgreSQL 数据库作为同构数据迁移的目标 AWS DMS](#)
- [使用 Amazon DocumentDB 数据库作为同构数据迁移的目标 AWS DMS](#)

## 使用与 MySQL 兼容的数据库作为同构数据迁移的目标 AWS DMS

可以使用与 MySQL 兼容的数据库作为 AWS DMS 中同构数据迁移的迁移目标。

AWS DMS 需要一定的权限才能将数据迁移到你的目标 Amazon RDS for MySQL、MariaDB 或 Amazon Aurora MySQL 数据库。可使用以下脚本，在 MySQL 目标数据库中创建具所需权限的数据库用户。

```
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'@'%';
```

在以上示例中，请用您自己的信息替换每个#####。

可使用以下脚本，在 MariaDB 数据库中创建具所需权限的数据库用户。对您迁移到的所有数据库运行 GRANT 查询 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'@'%';
```

在以上示例中，请用您自己的信息替换每个#####。

### Note

在 Amazon RDS 中，当您为 MySQL/Maria 数据库实例开启自动备份时，也会开启二进制日志记录。启用这些设置后，在目标数据库上创建二级对象（如函数、过程和触发器）时，数据迁

移任务可能会失败，并出现以下错误。如果目标数据库启用了二进制日志记录，那么请在开始任务之前，在数据库参数组中将 `log_bin_trust_function_creators` 设置为 `true`。

```
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)
```

## 使用与 MySQL 兼容的数据库作为同构数据迁移的目标的限制

使用与 MySQL 兼容的数据库作为同构数据迁移的目标时，存在以下限制：

- 用于连接到数据源的用户名具有以下限制：
  - 长度为 2 到 64 个字符。
  - 不能包含空格。
  - 可以包含以下字符：a-z、A-Z、0-9、下划线 (`_`)。
  - 不能包含连字符 (`-`)。
  - 必须以 a-z 或 A-Z 开头。
- 用于连接到数据源的密码具有以下限制：
  - 长度为 1 到 128 个字符。
  - 不能包含以下任何字符：单引号 (`'`)、双引号 (`"`)、分号 (`;`) 或空格。

## 使用 PostgreSQL 数据库作为同构数据迁移的目标 AWS DMS

可以使用 PostgreSQL 数据库作为 AWS DMS 中同构数据迁移的迁移目标。

AWS DMS 需要一定的权限才能将数据迁移到您的目标 Amazon RDS for PostgreSQL 数据库或亚马逊 Aurora PostgreSQL 数据库。可使用以下脚本，在 PostgreSQL 目标数据库中创建具所需权限的数据库用户。

```
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;
```

在以上示例中，请用您自己的信息替换每个#####。

要为 RDS for PostgreSQL 目标开启逻辑复制，请将数据库参数组中的 `rds.logical_replication` 参数设置为 1。此静态参数要求重启数据库实例或数据库集群才能生效。有些参数是静态的，只能在服务器启动时进行设置。AWS DMS 在重新启动服务器之前，会忽略其在数据库参数组中的条目的更改。

PostgreSQL 使用触发器实现外键约束。在满载阶段，每次 AWS DMS 加载一个表。我们建议您在完全加载期间，关闭目标数据库上的外键约束。为此，请使用以下方法之一：

- 临时关闭实例的所有触发器，并完成完全加载。
- 在 PostgreSQL 中更改 `session_replication_role` 参数的值。

在任何给定时间，触发器可能处于以下状态之一：`origin`、`replica`、`always` 或 `disabled`。如果将 `session_replication_role` 参数设置为 `replica`，那么只有处于 `replica` 状态的触发器处于活动状态。否则，触发器保持非活动状态。

## 使用与 PostgreSQL 兼容的数据库作为同构数据迁移的目标的限制

使用与 PostgreSQL 兼容的数据库作为同构数据迁移的目标时，存在以下限制：

- 用于连接到数据源的用户名具有以下限制：
  - 长度为 2 到 64 个字符。
  - 不能包含空格。
  - 可以包含以下字符：`a-z`、`A-Z`、`0-9`、下划线 (`_`)。
  - 必须以 `a-z` 或 `A-Z` 开头。
- 用于连接到数据源的密码具有以下限制：
  - 长度为 1 到 128 个字符。
  - 不能包含以下任何字符：单引号 (`'`)、双引号 (`"`)、分号 (`;`) 或空格。

## 使用 Amazon DocumentDB 数据库作为同构数据迁移的目标 AWS DMS

您可以使用亚马逊 DocumentDB (兼容 MongoDB) 数据库和 DocumentDB 弹性集群作为同构数据迁移的迁移目标。AWS DMS

要对 Amazon DocumentDB 目标运行同构数据迁移，您可以创建一个具有管理员权限的用户账户，也可以创建一个仅对要迁移的数据库具有读/写权限的用户。

同构数据迁移支持 Amazon DocumentDB 支持的所有 BSON 数据类型。有关这些数据类型的列表，请参阅 Amazon DocumentDB 开发者指南中的[数据类型](#)。

要使用 DocumentDB Elastic 集群的分片功能从源迁移非分片集合，请在开始数据迁移任务之前创建一个要迁移的分片集合。有关亚马逊 DocumentDB 弹性集群中分片集合的更多信息，[请参阅《亚马逊 DocumentDB 开发者指南》中的步骤 5：对您的馆藏进行分片](#)。

对于亚马逊文档数据库目标，AWS DMS 支持none或 require SSL 模式。

## 在中运行同构数据迁移 AWS DMS

您可以使用[同构数据迁移](#)在将数据从源数据库迁移 AWS DMS 到亚马逊关系数据库服务 (Amazon RDS)、亚马逊 Aurora 或 Amazon DocumentDB 上的等效引擎。AWS DMS 通过在源数据库和目标数据库中使用本机数据库工具自动执行数据迁移过程。

为同构数据迁移创建实例配置文件和兼容的数据提供程序后，创建一个迁移项目。有关更多信息，请参阅[创建迁移项目](#)。

以下各节介绍了如何创建、配置和运行同构数据迁移。

### 主题

- [在中创建数据迁移 AWS DMS](#)
- [同构数据迁移的选择规则](#)
- [管理中的数据迁移 AWS DMS](#)
- [监控中的数据迁移 AWS DMS](#)
- [中同构数据迁移的状态 AWS DMS](#)
- [通过同构数据迁移从 MySQL 数据库迁移数据 AWS DMS](#)
- [通过同构数据迁移从 PostgreSQL 数据库迁移数据 AWS DMS](#)
- [通过同构数据迁移从 MongoDB 数据库迁移数据 AWS DMS](#)

## 在中创建数据迁移 AWS DMS

使用相同类型的兼容数据提供程序创建迁移项目后，您可以使用此项目进行同构数据迁移。有关更多信息，请参阅 [创建迁移项目](#)。

要开始使用同构数据迁移，请创建新的数据迁移。可以在单个迁移项目中创建多个不同类型的同构数据迁移。

AWS DMS 具有您可以为自己创建的最大同构数据迁移数量。AWS 账户有关 AWS DMS 服务配额的信息，请参阅以下部分 [AWS Database Migration Service 的限额](#)。

在创建数据迁移之前，请务必设置所需的资源，例如，源数据库和目标数据库、IAM policy 和角色、实例配置文件和数据提供程序。有关更多信息，请参阅 [创建 IAM 资源](#)、[创建实例配置文件](#) 和 [创建实例提供程序](#)。

此外，我们建议您不要使用同构数据迁移，将数据从较高的数据库版本迁移到较低的数据库版本。请检查用于源和目标数据提供程序的数据库版本，并在需要时升级目标数据库版本。

### 创建数据迁移

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后在数据迁移选项卡上，选择创建数据迁移。
4. 对于名称，输入数据迁移的名称。确保为数据迁移使用唯一的名称，以便轻松识别。
5. 对于复制类型，选择要配置的数据迁移类型。可以选择以下选项之一：
  - 完全加载 – 迁移现有的源数据。
  - 完全加载和更改数据捕获 ( CDC ) – 迁移现有的源数据并复制正在进行的更改。
  - 更改数据捕获 ( CDC ) – 复制正在进行的更改。
6. 选中“打开日 CloudWatch 志”复选框将数据迁移日志存储在 Amazon 中 CloudWatch。如果未选中此选项，那么当数据迁移失败时，将无法查看日志文件。
7. ( 可选 ) 展开高级设置。在任务数中，输入 AWS DMS 可用于将源数据迁移到目标的并行线程数。
8. 对于 IAM 服务角色，选择之前创建 IAM 角色。有关更多信息，请参阅 [为 AWS DMS 中的同构数据迁移创建 IAM 角色](#)。
9. 为更改数据捕获 ( CDC ) 类型的数据迁移配置启动模式。可以选择以下选项之一：

- 立即 – 在开始数据迁移的同时启动正在进行的复制。
- 使用本机开始点 – 从指定点启动正在进行的复制。

对于 PostgreSQL 数据库，在时隙名称中输入逻辑复制时隙的名称，在本机开始点中输入事务日志序列号。

对于 MySQL 数据库，在日志序列号 (LSN) 中输入事务日志序列号。

10. 为更改数据捕获 (CDC) 或完全加载和更改数据捕获 (CDC) 类型的数据迁移配置停止模式。可以选择以下选项之一：

- 不要停止 CDC — AWS DMS 继续进行正在进行的复制，直到您停止数据迁移。
- 使用服务器时间点-在指定时间 AWS DMS 停止正在进行的复制。

如果选择此选项，那么可以在停止日期和时间中，输入要自动停止正在进行的复制的日期和时间。

11. 选择创建数据迁移。

AWS DMS 创建您的数据迁移，并将其添加到迁移项目中“数据迁移”选项卡上的列表中。您可以在此处查看数据迁移的状态。有关更多信息，请参阅 [迁移状态](#)。

#### Important

对于“满载”和“满载和更改数据捕获 (CDC)”类型的数据迁移，请 AWS DMS 删除目标数据库上的所有数据、表和其他数据库对象。确保您有目标数据库的备份。

AWS DMS 创建数据迁移后，此数据迁移的状态将设置为“就绪”。要迁移数据，必须手动开始数据迁移。为此，请从列表中选择数据迁移。接下来，在操作中选择启动。有关更多信息，请参阅 [管理数据迁移](#)。

首次启动同构数据迁移需要进行一些设置。AWS DMS 为您的数据迁移创建无服务器环境。此过程耗时最多 15 分钟。停止并重新启动数据迁移后，AWS DMS 不会再次创建环境，您可以更快地访问数据迁移。

## 同构数据迁移的选择规则

您可以使用选择规则来选择要包含在复制中的架构、表或两者。

**Note**

AWS DMS 仅在使用兼容 MongoDB 的数据库作为源时，支持同构数据迁移的选择规则。

创建数据迁移任务时，选择添加选择规则。

对于规则设置，请提供以下值：

- 架构：选择输入架构。
- 架构名称：提供要复制的架构的名称，或%用作通配符。
- 表名:: 提供要复制的表的名称，或%用作通配符。

默认情况下，DMS 支持的唯一规则操作是Include，而 DMS 支持的唯一通配符是。%

Example 迁移架构中的所有表

以下示例将所有表从源中名为 dmsst 的架构中迁移到目标端点。

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-action": "include",
      "object-locator": {
        "schema-name": "dmsst",
        "table-name": "%"
      },
      "filters": [],
      "rule-id": "1",
      "rule-name": "1"
    }
  ]
}
```

Example 迁移架构中的一些表

以下示例将名称以开头的表从源代码dmsst中命名的架构迁移到目标终端节点。collectionTest

```
{
```

```
"rules": [  
  {  
    "rule-type": "selection",  
    "rule-action": "include",  
    "object-locator": {  
      "schema-name": "dmsst",  
      "table-name": "collectionTest%"  
    },  
    "filters": [],  
    "rule-id": "1",  
    "rule-name": "1"  
  }  
]  
}
```

### Example 从多个架构中迁移特定表

以下示例将源中名为dmsst和的多个架构Test中的一些表迁移到目标终端节点。

```
{  
  "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"  
    }  
  ]  
}
```

```
}
```

## 管理中的数据迁移 AWS DMS

创建数据迁移后，AWS DMS 不会自动开始迁移数据。您可以在需要时手动开始数据迁移。

开始数据迁移之前，您可以修改数据迁移的所有设置。开始数据迁移之后，便无法更改复制类型。要使用其他复制类型，请创建新的数据迁移。

### 开始数据迁移

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目。在数据迁移选项卡上，选择数据迁移。数据迁移的摘要页面随即打开。
4. 对于操作，选择启动。

之后，为您的数据迁移 AWS DMS 创建一个无服务器环境。此过程耗时最多 15 分钟。

开始数据迁移后，将其状态 AWS DMS 设置为“正在启动”。用于数据迁移的下一个状态取决于您在数据迁移设置中选择的复制类型。AWS DMS 有关更多信息，请参阅 [迁移状态](#)。

### 创建数据迁移

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目。在数据迁移选项卡上，选择数据迁移。数据迁移的摘要页面随即打开。
4. 选择修改。
5. 配置数据迁移设置。

#### Important

如果已开始数据迁移，那么无法更改复制类型。

6. 要在 Amazon 中查看您的数据迁移日志 CloudWatch，请选中“开启日 CloudWatch 志”复选框。
7. 选择保存更改。

AWS DMS 开始数据迁移后，您可以将其停止。为此，请在数据迁移选项卡上，选择数据迁移。接下来，在操作中选择停止。

停止数据迁移后，将其状态 AWS DMS 设置为“正在停止”。接下来，AWS DMS 将此数据迁移的状态设置为已停止。AWS DMS 停止数据迁移后，您可以修改、恢复、重新启动或删除数据迁移。

要继续数据复制，请在数据迁移选项卡上选择已停止的数据迁移。接下来，在操作中选择恢复处理。

要重新启动数据加载，请在数据迁移选项卡上选择已停止的数据迁移。接下来，在“操作”中选择“重新启动”。AWS DMS 从目标数据库中删除所有数据并从头开始数据迁移。

您可以删除已停止或尚未开始的数据迁移。要删除数据迁移，请在数据迁移选项卡上选择此数据迁移。接下来，在操作中选择删除。要删除迁移项目，请停止并删除所有数据迁移。

## 监控中的数据迁移 AWS DMS

开始同构数据迁移之后，您可以监控其状态和进度。大型数据集（例如数百 GB）的数据迁移需要数小时才能完成。为了保持数据迁移的可靠性、可用性和高性能，请定期监控迁移进度。

### 检查数据迁移的状态和进度

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后导航至数据迁移选项卡。
4. 对于数据迁移，请查看状态列。有关此列中值的更多信息，请参阅[迁移状态](#)。
5. 对于正在运行的数据迁移，迁移进度列显示了已迁移数据的百分比。

### 查看数据迁移的详细信息

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目。在数据迁移选项卡上，选择数据迁移。
4. 在详细信息选项卡上，可以看到迁移进度。特别是，可以查看以下指标。
  - 公有 IP 地址 – 数据迁移的公有 IP 地址。您需要此值才能配置网络。有关更多信息，请参阅[设置网络](#)。



- 已加载的表 – 成功加载的表的数量。
  - 正在加载的表 – 当前正在加载的表的数量。
  - 排队的表 – 当前正等待加载的表的数量。
  - 出错的表 – 加载失败的表的数量。
  - 已用时间 – 数据迁移开始后经过的时间。
  - CDC 延迟 — 从源表发生更改到 AWS DMS 将此更改应用于目标表之间的平均时间。
  - 迁移开始 – 开始此数据迁移的时间。
  - 迁移停止 – 停止此数据迁移的时间。
5. 要查看数据迁移的日志文件，请在同构数据迁移设置下选择查看 CloudWatch 日志。您可以在创建或修改数据迁移时开启日 CloudWatch 志。有关更多信息，请参阅 [创建数据迁移](#) 和 [管理数据迁移](#)。

您可以使用 Amazon CloudWatch 警报或事件来密切跟踪您的数据迁移。有关更多信息，请参阅[什么是亚马逊 CloudWatch](#)、[亚马逊 CloudWatch 事件](#)和[亚马逊 CloudWatch 日志？](#) 在《亚马逊 CloudWatch 用户指南》中。请注意，使用Amazon需要付费 CloudWatch。

对于同构数据迁移，请在 Amazon 中 AWS DMS 包括以下指标。 CloudWatch

指标	描述
OverallCDCLatency	<p>CDC 阶段的总体延迟。</p> <p>对于 MySQL 数据库，此指标显示从源二进制日志发生更改到复制此更改之间经过的秒数。</p> <p>对于 PostgreSQL 数据库，此指标显示 pg_stat_subscription 视图中从 last_msg_receipt_time 到 last_msg_send_time 之间经过的秒数。</p> <p>单位：秒</p>
StorageConsumption	<p>数据迁移消耗的存储空间。</p> <p>单位：字节</p>

## 中同构数据迁移的状态 AWS DMS

对于您运行的每次数据迁移，都会在 AWS DMS 控制台中 AWS DMS 显示状态。以下列出了可用状态。

- **Creating**— AWS DMS 正在创建数据迁移。
- **Ready** – 数据迁移已准备好开始。
- **Starting**— AWS DMS 正在为您的数据迁移创建无服务器环境。此过程耗时最多 15 分钟。
- **Load running**— AWS DMS 正在执行满负荷迁移。
- **Load complete, replication ongoing**— AWS DMS 已完成满载，现在可以复制正在进行的更改。AWS DMS 此状态仅用于满载和更改数据捕获 (CDC) 类型的数据迁移。
- **Replication ongoing**— AWS DMS 正在复制正在进行的更改。AWS DMS 此状态仅用于变更数据捕获 (CDC) 类型的迁移。
- **Reloading target**— AWS DMS 正在重新启动数据迁移并执行指定的迁移类型。
- **Stopping**— AWS DMS 正在停止数据迁移。AWS DMS 在“操作”菜单上选择停止数据迁移后设置此状态。
- **Stopped**— AWS DMS 已停止数据迁移。
- **Failed** – 数据迁移失败。有关更多信息，请参阅日志文件。

要查看日志文件，请在数据迁移选项卡上，选择数据迁移。接下来，在“同构数据迁移设置”下选择“查看 CloudWatch 日志”。

### Important

如果在创建数据迁移时选中“打开日 CloudWatch 志”复选框，则可以查看日志文件。

- **Deleting**— AWS DMS 正在删除数据迁移。AWS DMS 在“操作”菜单上选择删除数据迁移后设置此状态。

## 通过同构数据迁移从 MySQL 数据库迁移数据 AWS DMS

您可以使用[同构数据迁移](#)，将自管理 MySQL 数据库迁移到 RDS for MySQL 或 Aurora MySQL。AWS DMS 会为数据迁移创建无服务器环境。对于不同类型的数据迁移，AWS DMS 使用不同的原生 MySQL 数据库工具。

对于满负荷类型的同构数据迁移，AWS DMS 使用 `mydumper` 从源数据库读取数据并将其存储在连接到无服务器环境的磁盘上。AWS DMS 读取所有源数据后，它使用目标数据库中的 `myloader` 来恢复您的数据。

对于满载和更改数据捕获 (CDC) 类型的同构数据迁移，请 AWS DMS 使用 `mydumper` 从源数据库读取数据并将其存储在连接到无服务器环境的磁盘上。AWS DMS 读取所有源数据后，它使用目标数据库中的 `myloader` 来恢复您的数据。AWS DMS 完成满载后，它会设置二进制日志复制，并将二进制日志位置设置为满载的开始。为避免数据不一致，请将任务数设置为 1，以捕获现有数据的一致状态。有关更多信息，请参阅 [创建数据迁移](#)。

对于更改数据捕获 (CDC) 类型的同构数据迁移，AWS DMS 需要本机 CDC 开始点才能开始复制。如果您提供本地 CDC 起点，则会 AWS DMS 捕获从该点开始的更改。或者，在数据迁移设置中选择立即，以便在实际数据迁移开始时自动捕获复制的开始点。

#### Note

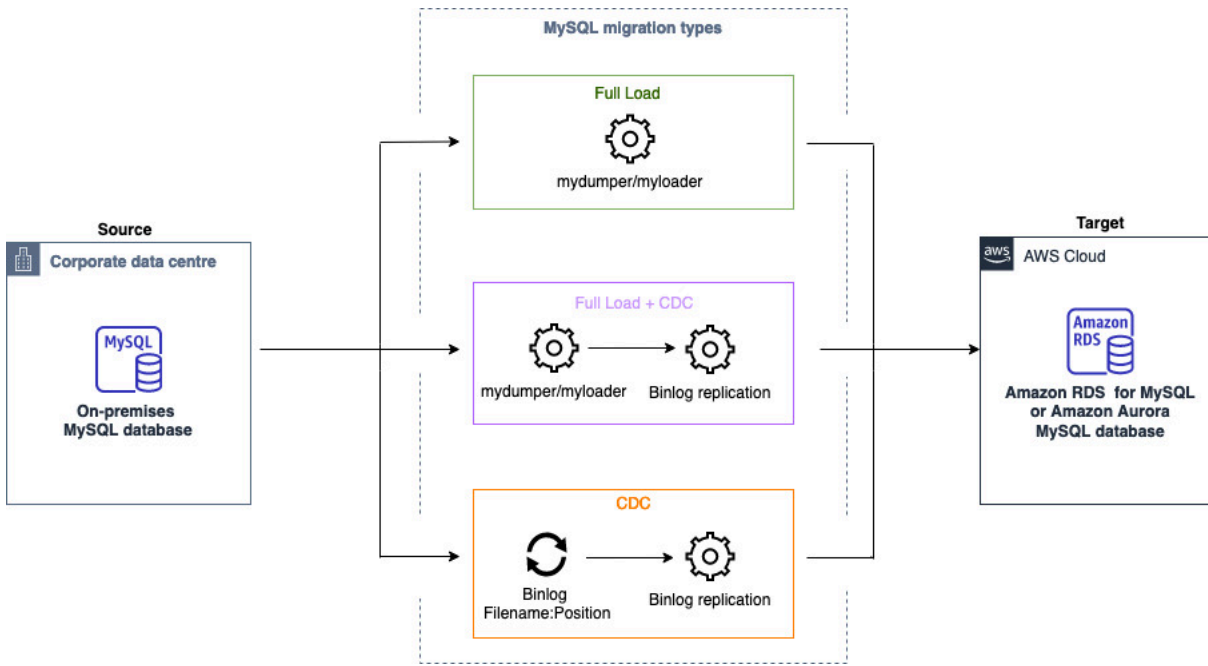
要使仅 CDC 的迁移正常运行，所有源数据库架构和对象都必须已存在于目标数据库中。不过，目标可能具有源上不存在的对象。

您可以使用以下代码示例获取 MySQL 数据库中当前的日志序列号 (LSN)。

```
show master status
```

此查询返回二进制日志文件名和位置。对于本机开始点，可使用二进制日志文件名和位置的组合。例如，`mysql-bin-changelog.000024:373`。在此示例中，`mysql-bin-changelog.000024` 是二进制日志文件名，`373` 也是 AWS DMS 开始捕获更改的位置。

下图显示了使用同构数据迁移将 MySQL 数据库迁移 AWS DMS 到适用于 MySQL 的 RDS 或 Aurora MySQL 的过程。



## 通过同构数据迁移从 PostgreSQL 数据库迁移数据 AWS DMS

您可以使用[同构数据迁移](#)，将自管理 PostgreSQL 数据库迁移到 RDS for PostgreSQL 或 Aurora PostgreSQL。AWS DMS 会为数据迁移创建无服务器环境。对于不同类型的数据迁移，AWS DMS 会使用不同的本机 PostgreSQL 数据库工具。

对于满负荷类型的同构数据迁移，AWS DMS 使用 `pg_dump` 从源数据库读取数据并将其存储在连接到无服务器环境的磁盘上。AWS DMS 读取所有源数据后，它使用目标数据库中的 `pg_restore` 来恢复您的数据。

对于完全加载和更改数据捕获 (CDC) 类型的同构数据迁移，AWS DMS 使用 `pg_dump` 从源数据库中读取没有表数据的架构对象，并将它们存储在连接到无服务器环境的磁盘上。然后，它 `pg_restore` 在目标数据库中使用来恢复您的架构对象。AWS DMS 完成该 `pg_restore` 过程后，它会自动切换到发布者和订阅者模式进行逻辑复制，Initial Data Synchronization 可以选择将初始表数据直接从源数据库复制到目标数据库，然后启动正在进行的复制。在此模式中，一个或多个订阅用户会订阅发布者节点上的一个或多个发布。

对于变更数据捕获 (CDC) 类型的同构数据迁移，AWS DMS 需要本机起点才能开始复制。如果您提供了本机起点，则会 AWS DMS 捕获从该点开始的更改。或者，在数据迁移设置中选择立即，以便在实际数据迁移开始时自动捕获复制的开始点。

**Note**

要使仅 CDC 的迁移正常运行，所有源数据库架构和对象都必须已存在于目标数据库中。不过，目标可能具有源上不存在的对象。

您可以使用以下代码示例获取 PostgreSQL 数据库中的本机开始点。

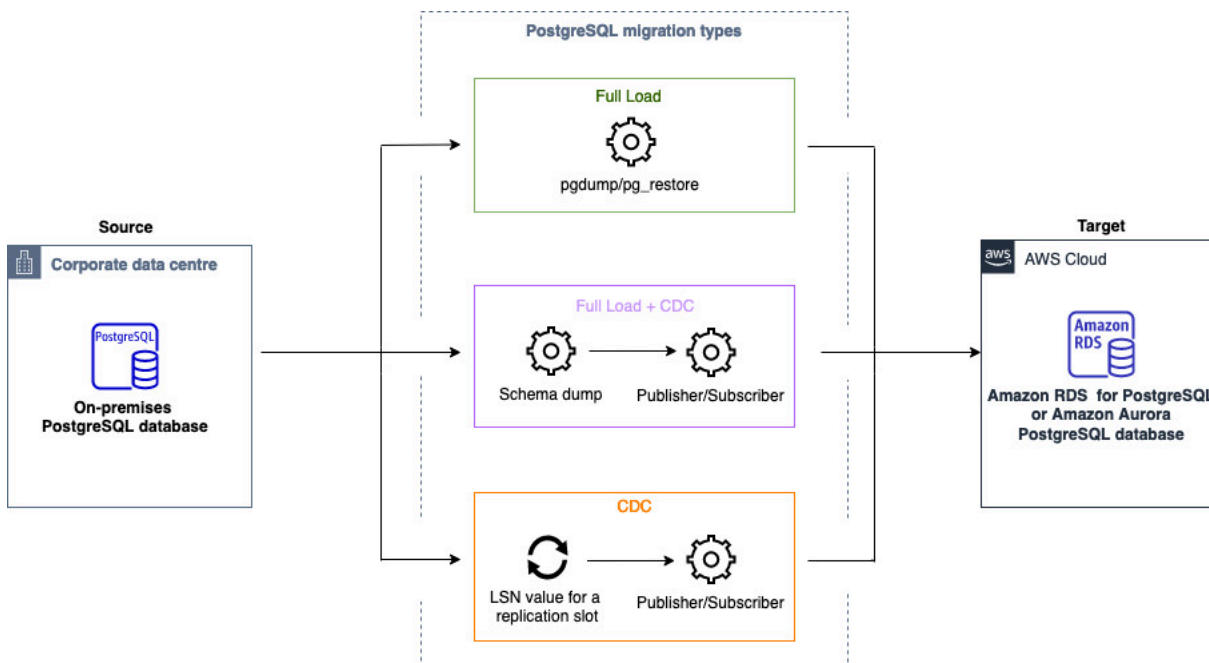
```
select confirmed_flush_lsn from pg_replication_slots where
slot_name='migrate_to_target';
```

此查询使用 PostgreSQL 数据库中的 `pg_replication_slots` 视图，捕获日志序列号 (LSN) 值。

将 PostgreSQL 同构数据迁移的状态 AWS DMS 设置为“已停止”、“失败”或“已删除”后，不会删除发布者和复制。如果您不希望恢复迁移，可使用以下命令删除复制时隙和发布者。

```
SELECT pg_drop_replication_slot('migration_subscriber_{ARN}');
DROP PUBLICATION publication_{ARN};
```

下图显示了使用同构数据迁移将 PostgreSQL 数据库迁移 AWS DMS 到适用于 PostgreSQL 的 RDS 或 Aurora PostgreSQL 的过程。



## 通过同构数据迁移从 MongoDB 数据库迁移数据 AWS DMS

您可以使用将自行管理的 MongoDB 数据库迁移[同构数据迁移](#)到亚马逊 DocumentDB。AWS DMS 为您的数据迁移创建无服务器环境。对于不同类型的数据迁移，AWS DMS 使用不同的原生 MongoDB 数据库工具。

对于满负荷类型的同构数据迁移，AWS DMS 使用 `mongodump` 从源数据库读取数据并将其存储在连接到无服务器环境的磁盘上。AWS DMS 读取所有源数据后，它将在目标数据库 `mongorestore` 中使用来恢复您的数据。

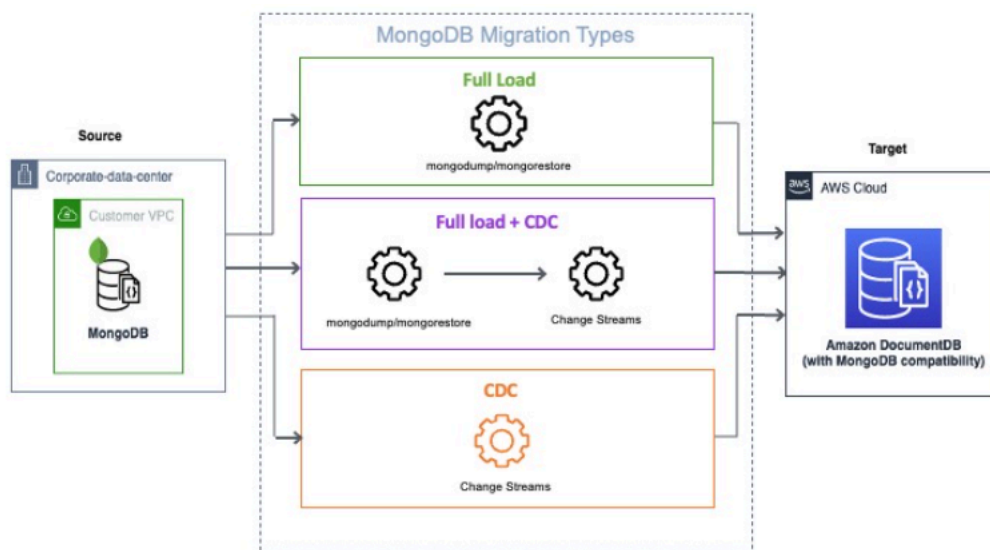
对于完全加载和更改数据捕获 (CDC) 类型的同构数据迁移，AWS DMS 使用 `mongodump` 从源数据库读取数据并将其存储在连接到无服务器环境的磁盘上。AWS DMS 读取所有源数据后，它将在目标数据库 `mongorestore` 中使用来恢复您的数据。AWS DMS 完成满载后，它会自动切换到发布者和订阅者模式进行逻辑复制。在此模型中，我们建议调整操作日志的大小，以将更改保留至少 24 小时。

对于更改数据捕获 (CDC) 类型的同构数据迁移，请在数据迁移设置 `immediately` 中进行选择，以便在实际数据迁移开始时自动捕获复制的起点。

### Note

对于任何新的或重命名的集合，你需要为这些集合创建一个新的数据迁移任务，作为同构数据迁移。对于与 MongoDB 兼容的源，AWS DMS 不支持 `create` 和操作。 `rename drop collection`

下图显示了使用同构数据迁移将 MongoDB 数据库迁移到 Amazon AWS DMS on DocumentDB 的过程。



# 对 AWS DMS 中的同构数据迁移进行故障排除

在下面的列表中，您可以找到在 AWS DMS 中遇到同构数据迁移问题时应采取的措施。

## 主题

- [我无法在 AWS DMS 中创建同构数据迁移](#)
- [我无法在 AWS DMS 中启动同构数据迁移](#)
- [在 AWS DMS 中运行数据迁移时我无法连接到目标数据库](#)
- [AWS DMS 在 PostgreSQL 中以表的形式迁移视图](#)

## 我无法在 AWS DMS 中创建同构数据迁移

如果您在选择了创建数据迁移后收到一条错误消息，提示 AWS DMS 无法连接到数据提供程序，请确保您已配置所需的 IAM 角色。有关更多信息，请参阅[创建 IAM 角色](#)。

如果您已配置 IAM 角色但仍收到此错误消息，请在 AWS KMS 密钥配置中将此 IAM 角色添加到密钥用户。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的[允许密钥用户使用 KMS 密钥](#)。

## 我无法在 AWS DMS 中启动同构数据迁移

如果您在迁移项目中开始数据迁移时获得 Failed 状态，请检查源和目标数据提供程序的版本。为此，请在您的 MySQL 或 PostgreSQL 数据库中运行 `SELECT VERSION();` 查询。请确保使用支持的数据库版本。

有关支持的源数据库的列表，请参阅[DMS 同构数据迁移的源](#)。

有关支持的目标数据库列表，请参阅[DMS 同构数据迁移的目标](#)。

如果您使用的数据库版本不受支持，请升级源数据库或目标数据库，然后重试。

在 AWS DMS 控制台中查看有关数据迁移的错误消息。为此，请打开迁移项目，然后选择数据迁移。在详细信息选项卡上，在常规下选中最后一条失败消息。

最后，分析 CloudWatch 日志。为此，请打开迁移项目，然后选择数据迁移。在详细信息选项卡上，选择查看 CloudWatch 日志。

## 在 AWS DMS 中运行数据迁移时我无法连接到目标数据库

如果您收到无法连接目标的错误消息，请执行以下操作。

1. 确保与源数据库和目标数据库相连的安全组包含针对任何入站和出站流量的规则。有关更多信息，请参阅[配置持续的数据复制](#)。
2. 验证网络访问控制列表 (ACL) 和路由表规则。
3. 数据库必须可从您创建的 VPC 进行访问。在 VPC 安全组中添加公共 IP 地址，并在防火墙中允许输入连接。
4. 在迁移项目的数据迁移选项卡上，选择数据迁移。记下详细信息选项卡上连接和安全下的公共 IP 地址。接下来，在源数据库和目标数据库中允许从数据迁移的公共 IP 地址进行访问。
5. 要进行持续的数据复制，请确保源数据库和目标数据库可以相互通信。

有关更多信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的[使用安全组控制到资源的流量](#)。

## AWS DMS 在 PostgreSQL 中以表的形式迁移视图

同构数据迁移不支持在 PostgreSQL 中以视图的形式迁移视图。对于 PostgreSQL，AWS DMS 以表的形式迁移视图。



# 与数据提供者、实例配置文件和迁移项目合作 AWS DMS

当您在中使用 DMS 架构转换和同构数据迁移时 AWS Database Migration Service，您就是在处理迁移项目。反过来，AWS DMS 迁移项目会使用子网组、实例配置文件和数据提供程序。

子网是您的 VPC 内的 IP 地址范围。复制子网组包括来自不同可用区的子网，您的实例配置文件可以使用这些子网。请注意，复制子网组是一种 DMS 资源，与 Amazon VPC 和 Amazon RDS 使用的子网组不同。

实例配置文件为运行迁移项目的无服务器环境指定网络和安全设置。

数据提供程序存储数据库的数据存储类型和位置信息。将数据提供者添加到迁移项目后，您需要提供来自的数据库凭据 AWS Secrets Manager。AWS DMS 使用此信息连接到您的数据库。

创建数据提供程序、实例配置文件和其他 AWS 资源后，您可以创建迁移项目。迁移项目描述了您的实例配置文件、源和目标数据提供者以及来自的密钥 AWS Secrets Manager。您可以为不同的源和目标数据提供程序创建多个迁移项目。

您的大部分工作都在迁移项目中完成。对于 DMS 架构转换，您可以使用迁移项目来评测源数据提供程序的对象，并将它们转换为与目标数据库兼容的格式。然后，您可以将转换后的代码应用于目标数据提供程序或将其另存为 SQL 脚本。对于同构数据迁移，您可以使用迁移项目，将数据从源数据库迁移到 AWS Cloud 中相同类型的目标数据库。

中的迁移项目仅 AWS DMS 是无服务器的。AWS DMS 自动为您的迁移项目配置云资源。

AWS DMS 具有您可以为自己创建的最大实例配置文件、数据提供程序和迁移项目数量 AWS 账户。有关 AWS DMS 服务限额的信息，请参阅以下章节：[AWS Database Migration Service 的限额](#)。

## 主题

- [为 AWS DMS 迁移项目创建子网组](#)
- [为创建实例配置文件 AWS Database Migration Service](#)
- [在中创建数据提供者 AWS Database Migration Service](#)
- [在中创建迁移项目 AWS Database Migration Service](#)
- [管理中的迁移项目 AWS Database Migration Service](#)

## 为 AWS DMS 迁移项目创建子网组

在创建实例配置文件之前，请为您的实例配置文件配置子网组。

## 创建子网组

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择子网组，然后选择创建子网组。
3. 对于名称，请为您的子网组输入唯一的名称。
4. 对于描述，请输入子网组的描述。
5. 对于 VPC，请选择至少在两个可用区中分别有一个子网的 VPC。
6. 对于添加子网，请选择要包含在子网组中的子网。您必须选择至少位于两个可用区中的子网。

要连接到 Amazon RDS 数据库，请将公有子网添加到您的子网组中。要连接到本地数据库，请将私有子网添加到您的子网组中。

7. 选择 Create subnet group (创建子网组)。

## 为创建实例配置文件 AWS Database Migration Service

您可以在 AWS DMS 控制台中创建多个实例配置文件。请务必为在 AWS DMS 中创建的每个迁移项目选择一个实例配置文件。

### 要创建实例配置文件

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择实例配置文件。
3. 选择创建实例配置文件。
4. 在创建实例配置文件页面上，为您的实例配置文件的名称输入描述性值。
5. 对于网络类型，选择双堆栈模式以创建支持 IPv4 和 IPv6 寻址的实例配置文件。保留默认选项可创建仅支持 IPv4 寻址的实例配置文件。
6. 接下来，选择虚拟私有云 (VPC) 以运行所选网络类型的实例。然后为您的实例配置文件选择子网组和 VPC 安全组。

要连接到 Amazon RDS 数据库，请使用包含公有子网的子网组。要连接到本地数据库，请使用包含私有子网的子网组。确保您的网络已配置为 AWS DMS 可以使用 NAT 网关的公有 IP 地址访问您的源本地数据库。有关更多信息，请参阅 [基于 Amazon VPC 创建 VPC](#)。

7. (可选) 在您为 DMS 架构转换创建迁移项目时, 对于架构转换设置 – 可选, 选择一个 Amazon S3 存储桶来存储迁移项目的信息。然后选择提供对此 Amazon S3 存储桶的访问权限的 AWS Identity and Access Management (IAM) 角色。有关更多信息, 请参阅 [创建 Amazon S3 存储桶](#)。
8. 选择创建实例配置文件。

创建实例配置文件后, 您可以修改或删除实例配置文件。

### 修改实例配置文件

1. 登录 AWS Management Console 并打开 AWS DMS 控制台, [网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择 Instance profiles (实例配置文件)。实例配置文件页面打开。
3. 选择实例配置文件, 然后选择修改。
4. 更新您的实例配置文件名称, 编辑 VPC 或 Amazon S3 存储桶设置。
5. 选择保存更改。

### 删除实例配置文件

1. 登录 AWS Management Console 并打开 AWS DMS 控制台, [网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择 Instance profiles (实例配置文件)。实例配置文件页面打开。
3. 选择实例配置文件, 然后选择删除。
4. 选择删除以确认您的选择。

## 在中创建数据提供器 AWS Database Migration Service

您可以创建数据提供器并在 AWS DMS 迁移项目中使用它们。您的数据提供程序可以是自管理引擎, 可运行在本地, 也可运行在 Amazon EC2 实例上。此外, 您的数据提供程序可以是完全托管式引擎, 例如 Amazon Relational Database Service ( Amazon RDS ) 或 Amazon Aurora。

对于每个数据库, 您可以创建一个数据提供程序。您可以将单个数据提供程序用于多个迁移项目。

在创建迁移项目之前, 请确保您已经创建了至少两个数据提供程序。其中一个数据提供程序必须位于 AWS 服务上。您不能使用 AWS DMS 来转换架构或将数据迁移到本地数据库。

以下过程说明如何在 AWS DMS 控制台向导中创建数据提供器。

## 创建数据提供程序

1. 登录 AWS Management Console，然后[通过 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/) 打开主 AWS DMS 机。
2. 选择数据提供程序。数据提供程序页面打开。
3. 选择创建数据提供程序。下表描述了设置。

选项	操作
配置	选择是手动输入数据提供程序的信息，还是使用 Amazon RDS 数据库实例。
名称	输入数据提供程序的名称。确保为数据提供程序使用唯一的名称，以便轻松识别。
引擎类型	选择数据提供程序的数据库引擎类型。
服务器名称	输入数据库服务器的域名服务 ( DNS ) 名称或 IP 地址。用于同构复制的数据提供程序的服务器名称必须以字母数字字符开头，并且只能包含字母数字字符、连字符 (-)、句点 (.) 或下划线 ( _ )。
端口	输入用于连接到数据库服务器的端口。
服务 ID ( SID ) 或服务名称	输入 Oracle 系统 ID ( SID )。要查找 Oracle SID，请向 Oracle 数据库提交以下查询： <pre>SELECT sys_context('userenv','instance_name') AS SID FROM dual;</pre>
数据库名称	输入此数据提供程序的数据库的名称。用于同构复制的数据提供程序的数据库名称最多可以包含 63 个字符，并且不能包含空格。
安全套接字层 ( SSL ) 模式	如果要为此数据提供程序开启连接加密，请选择 SSL 模式。根据您选择的模式，您可能需要提供证书和服务器证书信息。有关更多详细信息，请参阅 <a href="#">将 SSL 与一起使用 AWS Database Migration Service</a> 。

选项	操作
身份验证模式	对于 MongoDB 源，用于对端点连接进行身份验证 AWS DMS 的身份验证模式。
身份验证源	对于 MongoDB 源，是用于验证身份验证凭据的 MongoDB 数据库的名称。
身份验证机制	对于 MongoDB 来源，这是 MongoDB 用来加密密码的身份验证方法。

#### 4. 选择创建数据提供程序。

创建数据提供程序后，请务必在 AWS Secrets Manager 中添加数据库连接凭证。

## 在中创建迁移项目 AWS Database Migration Service

在中创建迁移项目之前 AWS DMS，请务必创建以下资源：

- 描述您的源和目标数据库的数据提供程序
- 带有数据库凭据的密钥存储在 AWS Secrets Manager
- 提供 Secrets Manager 访问权限的 AWS Identity and Access Management (IAM) 角色
- 包含网络和安全设置的实例配置文件。

### 创建迁移项目

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择创建迁移项目。下表描述了设置。

选项	操作
名称	输入迁移项目的名称。确保为迁移项目使用唯一的名称，以便轻松识别。
实例配置文件	选择要用于迁移项目的实例配置文件。

选项	操作
源	选择浏览，然后选择您的源数据提供程序。
密钥 ID	在存储您的源数据库凭证的 Secrets Manager 中，选择密钥的 Amazon 资源名称 ( ARN )。
IAM 角色	选择一个 IAM 角色，以提供对 Secrets Manager 中源数据库凭证的访问权限。
目标	选择浏览，然后选择您的源数据提供程序。
密钥 ID	在存储目标数据库凭证的 Secrets Manager 中选择您的密钥的 ARN。
IAM 角色	选择一个 IAM 角色，以提供对 Secrets Manager 中目标数据库凭证的访问权限。
转换规则	( 可选 ) 如果您为 DMS 架构转换创建迁移项目，请选择添加转换规则来设置转换规则。转换规则使您可以根据您指定的规则更改对象名称。有关更多信息，请参阅 <a href="#">设置转换规则</a> 。

#### 4. 选择创建迁移项目。

AWS DMS 创建迁移项目后，您可以在 DMS 架构转换或同构数据迁移中使用此项目。要开始处理您的迁移项目，请在迁移项目页面上，从列表中选择您的项目。

## 管理中的迁移项目 AWS Database Migration Service

创建迁移项目后，您可以修改或删除该项目。例如，要更改源或目标数据提供程序，请修改您的迁移项目。

只有在关闭架构转换或数据迁移操作后，您才能修改或删除迁移项目。为此，请从列表中选择您的迁移项目，然后选择架构转换或数据迁移。接下来，为 DMS 架构转换选择关闭架构转换，然后确认您的选择。对于同构数据迁移，请选择您的数据迁移，然后在操作菜单上选择停止。编辑迁移项目后，您可以启动架构转换或重新启动数据迁移。

## 修改迁移项目

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后选择修改。
4. 更新项目名称、编辑实例配置文件或更改源和目标数据提供程序。（可选）添加或编辑在转换过程中更改对象名称的迁移规则。
5. 选择保存更改。

## 删除迁移项目

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 选择迁移项目。将打开迁移项目页面。
3. 选择您的迁移项目，然后选择删除。
4. 选择删除以确认您的选择。

# AWS Database Migration Service 的最佳做法

要想尽可能高效地使用 AWS Database Migration Service (AWS DMS)，请参阅此部分中有关以最有效方式迁移数据的建议。

## 主题

- [AWS Database Migration Service 的迁移规划](#)
- [转换架构](#)
- [查看 AWS DMS 公共文档](#)
- [运行概念验证](#)
- [改进 AWS DMS 迁移的性能](#)
- [使用您自己的本地名称服务器](#)
- [迁移大型二进制对象 \(LOB\)](#)
- [使用行筛选提高迁移大型表时的性能](#)
- [持续复制](#)
- [减少源数据库上的负载](#)
- [减少目标数据库的瓶颈](#)
- [在迁移期间使用数据验证](#)
- [使用指标监控您的 AWS DMS 任务](#)
- [事件和通知](#)
- [使用任务日志以排除迁移问题](#)
- [使用 Time Travel 对复制任务进行故障排除](#)
- [为 Oracle 目标更改用户和架构](#)
- [更改 Oracle 目标的表和索引表空间](#)
- [升级复制实例的版本](#)
- [了解您的迁移成本](#)

## AWS Database Migration Service 的迁移规划

在规划使用 AWS Database Migration Service 进行数据库迁移时，请考虑以下事项：



- 要将源数据库和目标数据库连接到 AWS DMS 复制实例，您需要对网络进行配置。只需在与您的复制实例相同的虚拟私有云 (VPC) 中连接两个 AWS 资源，即可实现此目的。还可以进行更复杂的配置，例如通过虚拟专用网络 (VPN) 将本地数据库连接到 Amazon RDS 数据库实例。有关更多信息，请参见 [数据库迁移的网络配置](#)。
- 源端点和目标端点 – 确保您知道源数据库中的哪些信息和表需要迁移到目标数据库。AWS DMS 支持基本架构迁移，包括创建表和主键。但是，AWS DMS 不会在目标数据库中自动创建辅助索引、外键、用户账户等。根据源和目标数据库引擎，您可能需要设置补充日志记录或者修改源和目标数据库的其他设置。有关更多信息，请参见 [数据迁移的源](#) 和 [数据迁移的目标](#)：
- 架构和代码迁移 – AWS DMS 不执行架构或代码转换。您可以使用 Oracle SQL Developer、MySQL Workbench 和 pgAdmin III 等工具来转换架构。如果需要将现有架构转换到不同的数据库引擎，可以使用 AWS Schema Conversion Tool (AWS SCT)。它可以创建目标架构，也可以生成和创建整个架构：表、索引、视图等。您还可以使用此工具来将 PL/SQL 或 TSQL 转换为 PostgreSQL 和其他格式。有关 AWS SCT 的更多信息，请参见 [AWS SCT 用户指南](#)。
- 不支持的数据类型 – 确保可将源数据类型转换为目标数据库的等效数据类型。有关支持的数据类型的更多信息，请参见有关数据存储的源或目标部分。
- 诊断支持脚本结果 – 在规划迁移时，我们建议您运行诊断支持脚本。利用这些脚本的结果，您可以找到有关潜在迁移失败的预告信息。

如果您的数据库有可用的支持脚本，可以使用下文所述相应脚本主题中的链接下载脚本。验证并查看脚本后，可以按照脚本主题中描述的步骤在本地环境中运行该脚本。脚本运行完成后，您可以查看结果。我们建议将运行这些脚本作为任何故障排除工作的第一步。在与 AWS Support 团队合作时，这些结果可能很有用。有关更多信息，请参见 [使用 AWS DMS 中的诊断支持脚本](#)。

- 迁移前评估 – 迁移前评估会评估数据库迁移任务的指定组件，这有助于确定可能导致迁移任务无法按预期运行的任何问题。使用此评估，您可以在运行新的或修改的任务之前确定潜在的问题。有关使用迁移前评估的更多信息，请参见 [为任务启用和使用迁移前评估](#)。

## 转换架构

AWS DMS 不执行架构或代码转换。如果要将其现有架构转换为其他数据库引擎，您可以使用 AWS SCT。AWS SCT 将源对象、表、索引、视图、触发器和其他系统对象转换为目标数据定义语言 (DDL) 格式。您还可以使用 AWS SCT 将大部分应用程序代码（例如 PL/SQL 或 TSQL）转换为对等的目标语言。

您可以通过从 AWS 免费下载来获取 AWS SCT。有关 AWS SCT 的更多信息，请参见 [AWS SCT 用户指南](#)。

如果源端点和目标端点位于同一个数据库引擎上，则可以使用 Oracle SQL Developer、MySQL Workbench 或 PgAdmin 4 等工具来移动架构。

## 查看 AWS DMS 公共文档

我们强烈建议您在首次迁移之前浏览源端点和目标端点的 AWS DMS 公共文档页面。本文档可以协助您在开始迁移之前确定迁移的先决条件并了解当前的限制。有关更多信息，请参见 [使用 AWS DMS 端点](#)。

在迁移期间，公共文档可以帮您解决与 AWS DMS 相关的任何问题。文档中的故障排除页面可以帮您解决使用 AWS DMS 和所选端点数据库的常见问题。有关更多信息，请参见 [排除 AWS Database Migration Service 中的迁移任务问题](#)。

## 运行概念验证

为了协助在数据库迁移的早期阶段发现环境问题，我们建议您运行一次小型测试迁移。这样做还有助于设置更符合实际情况的迁移时间表。此外，您可能需要运行全面的测试迁移，以衡量 AWS DMS 能否通过网络处理数据库的吞吐量。在此期间，我们建议对您的初始完全加载和持续复制进行基准测试和优化。这样做有助于了解网络延迟和衡量整体性能。

此时，您还有机会了解自己的数据配置文件以及数据库的大小，包括以下内容：

- 大、中、小型表的数量。
- AWS DMS 如何处理数据类型和字符集转换。
- 包含大对象 (LOB) 列的表的数量。
- 运行测试迁移需要多长时间。

## 改进 AWS DMS 迁移的性能

AWS DMS 迁移的性能受多种因素影响：

- 源上的资源可用性。
- 可用的网络吞吐量。
- 复制服务器的资源容量。
- 目标接收更改的能力。

- 源数据的类型和分布。
- 要迁移的对象数量。

可以使用下面所述的部分或全部最佳实践来改进性能。是否使用这些实践之一取决于您的特定使用案例。您可以在下面找到一些限制：

### 预置合适的复制服务器

AWS DMS 是一项在 Amazon EC2 实例上运行的托管服务。此服务可连接到源数据库，读取源数据、格式化数据以供目标数据库使用，并将数据加载到目标数据库中。

大部分这种处理发生在内存中。但是，大型事务可能需要部分缓冲到磁盘上。缓存事务和日志文件也会写入磁盘。在以下章节中，您可以了解在选择复制服务器时应考虑的事项。

### CPU

AWS DMS 专为异构迁移而设计，但它也支持同构迁移。要执行同构迁移，请先将每个源数据类型转换为其等效 AWS DMS 的数据类型。然后将每个 AWS DMS 类型的数据转换为目标数据类型。您可以在《AWS DMS 用户指南》中找到适用于每个数据库引擎的这些转换参考资料。

要让 AWS DMS 以最佳性能执行这些转换，进行转换时必须有 CPU 可用。如果由于 CPU 过载而没有足够的 CPU 资源，可能会导致迁移速度缓慢，这也可能导致其他副作用。

### 复制实例类

一些较小的实例类足够用于测试服务或小型迁移。如果您的迁移涉及大量表，或者打算同时运行多个复制任务，应考虑使用较大的实例之一。最好使用较大的实例，因为该服务会消耗大量的内存和 CPU。

T2 类型实例设计用于提供适度的基准性能，并能够根据您的工作负载的需要实现性能的显著突增。它们可用于不经常或不持续使用完整 CPU、但偶尔需要突增性能的工作负载。T2 实例非常适合通用型工作负载，例如 Web 服务器、开发人员环境和小型数据库。如果您要排除迁移缓慢问题，并使用 T2 实例类型，请查看 CPU 利用率主机指标。它可以显示您是否突破了该实例类型的基准。

C4 实例类旨在为计算机密集型工作负载交付最高级别的处理器性能。它们显著提高了每秒数据包 (PPS) 的性能、降低了网络抖动和网络延迟。AWS DMS 可能会占用大量 CPU，尤其是在执行异构迁移和复制（例如从 Oracle 迁移到 PostgreSQL）时。针对这些情况，C4 实例是理想选择。

R4 实例类针对内存密集型工作负载优化了内存。使用 AWS DMS 持续迁移或复制高吞吐量事务系统有时会占用大量 CPU 和内存。R4 实例的每个 vCPU 包含更多内存。

## AWS DMS 支持 R5 和 C5 实例类

R5 实例类是内存优化型实例，旨在让处理内存中的大型数据集的工作负载实现快速性能。使用 AWS DMS 持续迁移或复制高吞吐量事务系统有时会占用大量 CPU 和内存。R5 实例每个 vCPU 提供的内存比 R4 高 5%，最大容量可提供 768 GiB 的内存。此外，与 R4 相比，R5 实例每 GiB 的价格提高了 10%，CPU 性能提高了大约 20%。

C5 实例类针对计算密集型工作负载进行了优化，能够以较低的价格提供经济实惠的高性能。它们显著提高了网络性能。弹性网络适配器 (ENA) 为 C5 实例提供高达 25 Gbps 的网络带宽和高达 14 Gbps 的 Amazon EBS 专用带宽。AWS DMS 可能会占用大量 CPU，尤其是在执行异构迁移和复制时，例如从 Oracle 迁移至 PostgreSQL。针对这些情况，C5 实例是理想选择。

### 存储

根据具体的实例类，您的复制实例附带 50 GB 或 100 GB 的数据存储。此存储用于存储加载期间收集的日志文件以及所有缓存更改。如果源系统繁忙或需要处理大量事务，则可能需要增加存储空间。如果您在复制服务器上运行多个任务，可能还需要增加存储空间。不过，通常默认容量已经足够。

AWS DMS 中的所有存储卷均为 GP2 或通用型固态硬盘 (SSD)。GP2 卷的基本性能为每秒三次 I/O 操作 (IOPS)，并且能够基于积分突增至 3000 IOPS。根据经验，请检查复制实例的 ReadIOPS 和 WriteIOPS 指标。确保这些值的总和不会超过该卷的基本性能。

### 多可用区

选择多可用区实例可以保护您的迁移免受存储故障的影响。大多数迁移都是短暂的，不会长时间运行。如果您将 AWS DMS 用于持续复制，选择多可用区实例可在出现存储问题时提高可用性。

在完全加载期间使用单可用区或多可用区复制实例并且发生失效转移或主机更换时，预计完全加载任务将失败。对于未完成或处于错误状态的其余表，您可以从失败点重新启动此任务。

### 并行加载多个表

默认情况下，AWS DMS 一次加载八个表。在使用非常大的复制服务器时，例如 `dms.c4.xlarge` 或更大的实例，您可以稍微提升此值来实现一些性能改进。但是，某些时候增加此并行度会降低性能。如果您的复制服务器相对较小，例如 `dms.t2.medium`，则建议您减少并行加载的表数量。

要在 AWS Management Console 中更改此数量，请打开控制台，选择任务，再选择创建或修改任务，然后选择高级设置。在 Tuning Settings (优化设置) 下，更改 Maximum number of tables to load in parallel (并行加载的最大表数) 选项。

要使用 AWS CLI 更改此数，请更改 TaskSettings 下的 MaxFullLoadSubTasks 参数。

## 使用并行完全加载

您可以根据分区和子分区从 Oracle、Microsoft SQL Server、MySQL、Sybase 和 IBM Db2 LUW 源执行并行加载。这样做可以缩短整个完全加载的持续时间。此外，在运行 AWS DMS 迁移任务时，您可以加快大型表或分区表的迁移速度。为此，请将表拆分为多个段，然后在同一个迁移任务中并行加载这些段。

要使用并行加载，您可以使用 `parallel-load` 选项创建类型为 `table-settings` 的表映射规则。在 `table-settings` 规则中，请指定要并行加载的一个或多个表的选择条件。要指定选择条件，请将 `parallel-load` 的 `type` 元素设置为下列设置之一：

- `partitions-auto`
- `subpartitions-auto`
- `partitions-list`
- `ranges`
- `none`

有关这些设置的更多信息，请参阅[表和集合设置规则和操作](#)。

## 使用索引、触发器和引用完整性约束

索引、触发器和引用完整性约束可能会影响您的迁移性能，并导致迁移失败。它们具体如何影响迁移取决于您的复制任务是完全加载任务还是持续复制（更改数据捕获或 CDC）任务。

对于完全加载任务，建议您删除主键索引、二级索引、引用完整性约束和数据操作语言 (DML) 触发器。或者，您也可以将其创建延迟到完全加载任务完成之后。完全加载任务期间不需要索引，如果存在索引，则会导致维护开销。由于完整加载任务一次加载一组表，这会违反引用完整性约束。同样，插入、更新和删除触发器会导致错误，例如，以前批量加载的表会触发行插入。由于增加了处理量，其他类型的触发器也会影响性能。

如果数据量相对较小并且不担心额外增加的迁移时间，您可以在完全加载任务之前生成主键和二级索引。应始终禁用引用完整性约束和触发器。

对于完全加载外加 CDC 任务，建议您在 CDC 阶段之前添加二级索引。由于 AWS DMS 使用逻辑复制，应确保采用支持 DML 操作的二级索引来防止全表扫描。您可以在 CDC 阶段之前暂停复制任务以构建索引、创建引用完整性约束，然后重新启动任务。

您应该在割接之前启用触发器。

## 关闭备份和事务日志记录

迁移到 Amazon RDS 数据库时，在您准备好割接之前，最好关闭目标上的备份和多可用区。类似地，迁移到非 Amazon RDS 系统时，通常最好关闭目标上的任何日志记录，直到完成割接。

## 使用多个任务

有时候，为单个迁移使用多个任务可以提升性能。如果您有不参与到通用事务的表集，也许可将迁移拆分为多个任务。在任务中维护事务一致性，因此，不同任务中的表不参与到通用事务中非常重要。此外，每个任务独立读取事务流，因此请注意，不要对源数据库施加过多压力。

您可以使用多个任务来创建多个单独的复制流。通过这样做，您可以并行处理源数据库上的读取、复制实例上的进程以及对目标数据库的写入。

## 优化更改处理

默认情况下，AWS DMS 在事务模式下处理更改，这可保护事务完整性。如果您可以承受事务完整性的临时失效，可以改为使用批量优化应用 选项。该选项有效分组事务并批量应用，以实现提高效率的目的。使用批量优化的应用选项几乎总是违反引用完整性约束。因此，我们建议您在迁移过程中关闭这些限制，然后在割接过程中再次将其打开。

## 使用您自己的本地名称服务器

通常，AWS DMS 复制实例使用 Amazon EC2 实例中的域名系统 (DNS) 解析程序来解析域端点。但是，如果您使用 Amazon Route 53 Resolver，则可以使用自己的本地名称服务器来解析某些端点。使用此工具，您可以使用入站和出站端点、转发规则和私有连接在本地和 AWS 之间进行查询。使用本地名称服务器的好处包括通过防火墙提高了安全性和易用性。

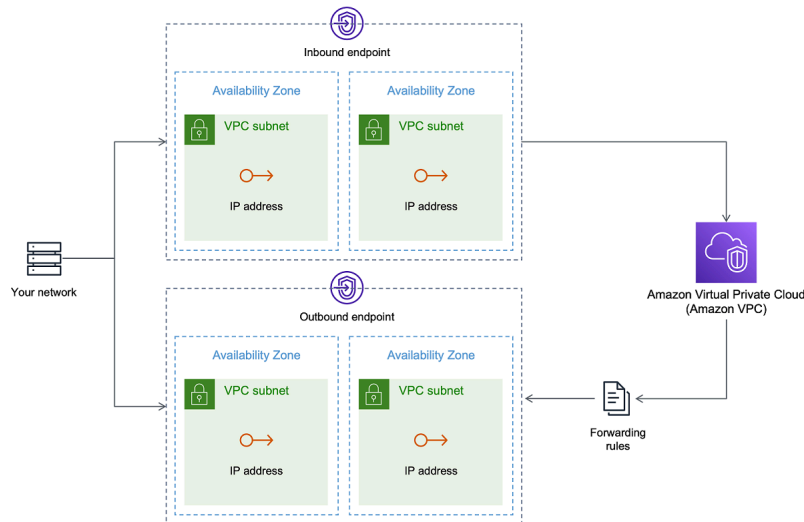
如果您有入站端点，则可以使用源自本地的 DNS 查询来解析 AWS 托管的域。要配置端点，请在您要提供解析程序的每个子网中分配 IP 地址。要在您的本地 DNS 基础设施与 AWS 之间建立连接，请使用 AWS Direct Connect 或虚拟专用网络 (VPN)。

出站端点连接到您的本地名称服务器。域名服务器仅授予对允许列表中包含和出站端点中设置的 IP 地址的访问权限。名称服务器的 IP 地址是目标 IP 地址。为出站端点选择安全组时，请选择复制实例使用的同一安全组。

要将选定域转发到名称服务器，请使用转发规则。出站端点可以处理多个转发规则。转发规则的范围是您的虚拟私有云 (VPC)。通过使用与 VPC 关联的转发规则，您可以预置 AWS 云的逻辑分隔部分。在这个逻辑分隔部分，您可以启动虚拟网络中的 AWS 资源。

您可将本地 DNS 基础设施中托管的域配置为设置出站 DNS 查询的条件转发规则。对其中一个域进行查询时，规则触发尝试将 DNS 请求转发到配置了规则的服务器。同样，需要通过 AWS Direct Connect 或 VPN 建立私有连接。

下图显示了 Route 53 Resolver 的架构。



有关 Route 53 DNS 解析程序的更多信息，请参阅《Amazon Route 53 开发人员指南》中的 [Route 53 Resolver 入门](#)。

## 将 Amazon Route 53 Resolver 与 AWS DMS 配合使用

您可以为 AWS DMS 创建本地名称服务器，从而使用 [Amazon Route 53 Resolver](#) 来解析端点。

为基于 Route 53 的 AWS DMS 创建本地名称服务器

1. 登录到 AWS Management Console，然后通过以下网址打开 Route 53 控制台：<https://console.aws.amazon.com/route53/>。
2. 在 Route 53 控制台上，选择要在其中配置 Route 53 Resolver 的 AWS 区域。Route 53 Resolver 特定于一个区域。
3. 选择查询方向，即入站和/或出站。
4. 提供您的入站查询配置：
  - a. 输入端点名称并选择 VPC。
  - b. 从 VPC 内分配一个或多个子网（例如，选择两个子网以确保可用性）。
  - c. 分配特定的 IP 地址以用作端点，或者让 Route 53 Resolver 自动分配这些地址。
5. 为本地域创建规则，以便 VPC 内的工作负载可以将 DNS 查询路由到您的 DNS 基础设施。

6. 为本地 DNS 服务器输入一个或多个 IP 地址。
7. 提交您的规则。

所有内容都已创建，您的 VPC 与您的入站和出站规则相关联，并且可以开始路由流量。

有关 Route 53 Resolver 的更多信息，请参阅《Amazon Route 53 开发人员指南》中的 [Route 53 Resolver 入门](#)。

## 迁移大型二进制对象 (LOB)

通常，AWS DMS 分两个阶段迁移 LOB 数据：

1. AWS DMS 会在目标表中创建一个新行，并用除相关 LOB 值之外的所有数据来填充该行。
2. AWS DMS 使用 LOB 数据更新目标表中的行。

LOB 的这种迁移过程要求，在迁移期间，目标表上的所有 LOB 列都必须可为空。即使源表上的 LOB 列不可为空，也是如此。如果 AWS DMS 创建目标表，默认情况下会将 LOB 列设置为可为空值。在某些情况下，您可能使用其他机制（例如导入或导出）来创建目标表。在这种情况下，在开始迁移任务之前，请确保 LOB 列可为空值。

此要求有一个例外情况。假设您执行从 Oracle 源到 Oracle 目标的同类迁移，并且您选择 Limited Lob mode (受限 LOB 模式)。在这种情况下，将一次填充整行，包括任何 LOB 值。对于此类情况，如果需要，AWS DMS 可以使用不可为空的约束创建目标表 LOB 列。

### 使用受限 LOB 模式

当迁移包含 LOB 值时，AWS DMS 使用两种方法来平衡性能与简便性。

1. Limited LOB mode (受限 LOB 模式) 迁移所有最大为用户指定的大小限制 (默认为 32 KB) 的 LOB 值。大于此大小限制的 LOB 值必须手动迁移。Limited LOB mode (受限 LOB 模式) 是所有迁移任务的默认值，通常提供最佳性能。不过，请确保最大 LOB 大小参数设置正确。应将该参数设置为所有表的最大 LOB 大小。
2. Full LOB mode (完整 LOB 模式) 迁移表中的所有 LOB 数据，而不受大小限制。Full LOB mode (完整 LOB 模式) 提供了移动表中所有 LOB 数据的便利性，不过该过程对性能可能会有显著影响。

对于某些数据库引擎（如 PostgreSQL），AWS DMS 处理 JSON 数据类型的方式类似于 LOB。确保如果您选择了受限 LOB 模式，则将最大 LOB 大小选项设置为不会导致 JSON 数据被截断的值。



AWS DMS 完全支持使用大对象数据类型 ( BLOB、CLOB 和 NCLOB )。以下源终端节点具有完整的 LOB 支持：

- Oracle
- Microsoft SQL Server
- ODBC

以下目标终端节点具有完整的 LOB 支持：

- Oracle
- Microsoft SQL Server

以下目标终端节点具有有限的 LOB 支持。对于此目标终端节点，您不能使用无限制的 LOB 大小。

- Amazon Redshift
- Amazon S3

对于具有完整 LOB 支持的终端节点，您还可以为 LOB 数据类型设置大小限制。

## 提高了 LOB 性能

迁移 LOB 数据时，您可以指定以下不同的 LOB 优化设置。

### 每个表的 LOB 设置

使用每个表的 LOB 设置，您可以覆盖部分或所有表的表级 LOB 设置。为此，请在 table-settings 规则中定义 lob-settings。以下是包含一些较大的 LOB 值的示例表。

```
SET SERVEROUTPUT ON
CREATE TABLE TEST_CLOB
(
  ID NUMBER,
  C1 CLOB,
  C2 VARCHAR2(4000)
);
DECLARE
bigtextstring CLOB := '123';
iINT;
```

```
BEGIN
WHILE Length(bigtextstring) <= 60000 LOOP
bigtextstring := bigtextstring || '00000000000000000000000000000000';
END LOOP;
INSERT INTO TEST_CLOB (ID, C1, C2) VALUES (0, bigtextstring,'AnyValue');
END;
/
SELECT * FROM TEST_CLOB;
COMMIT
```

接下来，创建迁移任务，并使用新 `lob-settings` 规则修改表的 LOB 处理。`bulk-max-siz` 值决定了最大 LOB 大小 (KB)。如果它大于指定的大小，则会被截断。

```
{
  "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"
    }
  }
]
}
```

即使此 AWS DMS 任务是使用 `FullLobMode : true` 创建的，每个表的 LOB 设置也会指示 AWS DMS 将此特定表中的 LOB 数据截断为 16000。您可以查看任务日志来确认这一点。

```
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
```

## 内联 LOB 设置

创建 AWS DMS 任务时，LOB 模式决定了如何处理 LOB。

对于完整 LOB 模式和受限 LOB 模式，每种模式都有自己的优缺点。内联 LOB 模式结合了完整 LOB 模式和受限 LOB 模式的优点。

当您需要同时复制小型和大型 LOB，并且大多数 LOB 都很小时，可以使用内联 LOB 模式。选择此选项时，在完全加载期间，AWS DMS 任务会以内联方式传输小 LOB，这样更加高效。AWS DMS 任务通过从源表中执行查找来传输大型 LOB。

在更改处理过程中，小型和大型 LOB 都通过从源表中执行查找来复制。

使用内联 LOB 模式时，AWS DMS 任务会检查所有 LOB 大小，以确定内联传输哪些 LOB。大于指定大小的 LOB 将使用完整 LOB 模式进行复制。因此，如果您知道大多数 LOB 的大小都大于指定设置，最好不要使用此选项。而应允许使用无限制的 LOB 大小。

您可以使用任务设置 `InlineLobMaxSize` 中的属性来配置此选项，该属性仅在 `FullLobMode` 设置为 `true` 时才可用。`InlineLobMaxSize` 的默认值是 0，范围为 1–102400 KB (100 MB)。

例如，您可以使用以下 AWS DMS 任务设置。此处，将 `InlineLobMaxSize` 的值设置为 5 会导致所有小于或等于 5 KiB (5120 字节) 的 LOB 都以内联方式传输。

```
{
  "TargetMetadata": {
    "TargetSchema": "",
    "SupportLobs": true,
    "FullLobMode": true,
    "LobChunkSize": 64,
    "LimitedSizeLobMode": false,
    "LobMaxSize": 32,
    "InlineLobMaxSize": 5,
    "LoadMaxFileSize": 0,
    "ParallelLoadThreads": 0,
    "ParallelLoadBufferSize": 0,
    "BatchApplyEnabled": false,
    "TaskRecoveryTableEnabled": false},
  . . .
```

}

## 使用行筛选提高迁移大型表时的性能

要在迁移大型表时改进性能，可将迁移过程分解为多个任务。要通过行筛选将迁移分解为多个任务，您可以使用键或分区键。例如，如果您有一个整数主键 ID，范围从 1 到 8000000，您可以使用行筛选创建 8 个任务，每个任务迁移 100 万条记录。

在控制台中应用行筛选：

1. 打开 AWS Management Console。
2. 选择任务，然后创建新任务。
3. 选择表映射选项卡，然后展开选择规则。
4. 选择添加新选择规则。现在可以使用小于或等于、大于或等于、等于或范围条件（介于两个值之间）添加列筛选。有关列筛选的更多信息，请参阅 [通过控制台指定表选择和转换规则](#)。

如果您有按照日期分区的大型分区表，您可以根据日期来迁移数据。例如，假设您有一个按月分区的表，并且只更新当前月份的数据。在此情况下，您可以为每个静态的每月分区创建一个完全加载任务，并为当前已更新的分区创建一个完全加载外加 CDC 任务。

如果您的表具有单列主键或唯一索引，则可以让 AWS DMS 任务使用范围类型的并行加载来对表进行分段，以并行加载数据。有关更多信息，请参见 [表和集合设置规则和操作](#)。

## 持续复制

AWS DMS 提供数据的持续复制，确保源数据库与目标数据库的同步。它只复制有限数量的数据定义语言 (DDL) 语句。AWS DMS 不传播索引、用户、权限、存储过程和其他不直接关联到表数据的数据库更改等项目。

如果您计划使用持续复制，则应在创建复制实例时启用多可用区选项。通过选择多可用区选项，可为复制实例提供高可用性和失效转移支持。但是，此选项可能会影响性能，并且在目标系统进行更改时可能会减慢复制速度。

在升级源数据库或目标数据库之前，我们建议您停止在这些数据库上运行的所有 AWS DMS 任务。升级完成后再继续执行任务。

在持续复制期间，确定源数据库系统和 AWS DMS 复制实例之间的网络带宽至关重要。确保网络不会在持续复制期间造成任何瓶颈。

确定源数据库系统上每小时的更改率和存档日志生成率也很重要。这样做可以帮您了解在持续复制期间可能获得的吞吐量。

## 减少源数据库上的负载

AWS DMS 使用源数据库上的一些资源。在完整加载任务期间，AWS DMS 对并行处理的每个表执行源表的完整表扫描。此外，您在迁移过程中创建的各个任务会在 CDC 过程中查询源中的更改。要使 AWS DMS 对一些源（例如 Oracle）执行 CDC，您可能需要增加写入到数据库更改日志中的数据量。

如果您发现对源数据库造成的负担过重，可以减少迁移过程中的任务数或每个任务的表数量。每个任务独立获取源更改，因此整合任务可以减少更改捕获工作负载。

## 减少目标数据库的瓶颈

迁移期间，请尽可能消除目标数据库上会争用写入资源的任何进程：

- 关闭不必要的触发器。
- 在初始加载期间关闭二级索引，稍后在持续复制期间将其重新打开。
- 对于 Amazon RDS 数据库，最好在割接之前关闭备份和多可用区。
- 迁移到非 RDS 系统时，最好在割接之前关闭目标上的任何日志记录。

## 在迁移期间使用数据验证

为确保准确地将数据从源迁移到目标，我们强烈建议您使用数据验证。如果为任务启用数据验证，AWS DMS 会在对表执行完全加载后，立即开始比较源和目标数据。

数据验证使用以下数据库，而 AWS DMS 支持它们作为源终端节点和目标终端节点：

- Oracle
- PostgreSQL
- MySQL
- MariaDB
- Microsoft SQL Server
- Amazon Aurora MySQL 兼容版
- Amazon Aurora PostgreSQL 兼容版

- IBM Db2 LUW
- Amazon Redshift

有关更多信息，请参见 [AWS DMS 数据验证](#)。

## 使用指标监控您的 AWS DMS 任务

您可以通过多种方式使用 AWS DMS 控制台监控任务的指标：

### 主机指标

您可以在每个特定复制实例的 CloudWatch 指标选项卡上找到主机指标。在这里，您可以监控您的复制实例的大小是否合适。

### 复制任务指标

可以在每个特定任务的指标选项卡上找到复制任务的 CloudWatch 指标，包括传入和提交的更改，以及复制主机和源/目标数据库之间的延迟。

### 表指标

您可以在每项任务的表统计选项卡上找到各个表指标。这些指标包括以下数字：

- 在完全加载期间加载的行。
- 自任务启动以来的插入、更新和删除操作。
- 自任务启动以来的 DDL 操作。

有关监视指标的更多信息，请参阅 [监控 AWS DMS 任务](#)。

## 事件和通知

AWS DMS 使用 Amazon SNS 在发生 AWS DMS 事件（例如创建或删除复制实例）时提供通知。您可以按照 Amazon SNS 在 AWS 区域中支持的任何形式使用这些通知。这些通知可能包括电子邮件、短信或对 HTTP 端点的调用。

有关更多信息，请参见 [在 AWS Database Migration Service 中使用 Amazon SNS 事件和通知](#)。

## 使用任务日志以排除迁移问题

在某些情况下，AWS DMS 会遇到一些仅在任务日志中显示警告或错误消息的问题。具体而言，数据截断问题或者外键违规造成的行拒绝仅会写入任务日志。因此，在迁移数据库时务必要查看任务日志。要查看任务日志，请在任务创建过程中配置 CloudWatch Amazon。

有关更多信息，请参阅[使用 Amazon 监控复制任务 CloudWatch](#)。

## 使用 Time Travel 对复制任务进行故障排除

要排除 AWS DMS 迁移问题，您可以使用 Time Travel。有关 Time Travel 的更多信息，请参阅[Time Travel 任务设置](#)。

在使用 Time Travel 时，请注意以下事项：

- 为避免 DMS 复制实例的开销，请仅对需要调试的任务启用 Time Travel。
- 当您使用 Time Travel 对可能持续数天的复制任务进行故障排除时，请监控复制实例指标以了解资源开销。这种方法尤其适用于在源数据库上长时间运行高事务负载的情况。有关更多详细信息，请参阅[监控 AWS DMS 任务](#)。
- 如果将 Time Travel 任务设置 EnableRawData 设置为 true，DMS 复制期间的任务内存使用量可能会高于未启用 Time Travel 时的内存使用量。如果您长时间启用 Time Travel，请监视您的任务。
- 目前，你只能在任务级别启用 Time Travel。对所有表的更改都记录在 Time Travel 日志中。如果您要对事务量大的数据库中的特定表进行故障排除，请创建单独的任务。

## 为 Oracle 目标更改用户和架构

在使用 Oracle 作为目标时，AWS DMS 将数据迁移到目标端点的用户所拥有的架构。

例如，假设您将名为 PERFDATA 的架构迁移到 Oracle 目标端点，并且目标端点用户名为 MASTER。AWS DMS 将作为 MASTER 连接到 Oracle 目标，并使用 PERFDATA 中的数据库对象填充 MASTER 架构。

要覆盖此行为，请提供架构转换。例如，要将 PERFDATA 架构对象迁移到目标端点上的 PERFDATA 架构，可使用以下转换。

```
{
  "rule-type": "transformation",
  "rule-id": "2",
```

```
"rule-name": "2",
"object-locator": {
  "schema-name": "PERFDATA"
},
"rule-target": "schema",
"rule-action": "rename",
"value": "PERFDATA"
}
```

有关转换的更多信息，请参阅 [使用 JSON 指定表选择和转换规则](#)。

## 更改 Oracle 目标的表和索引表空间

使用 Oracle 作为目标时，AWS DMS 会将所有表和索引迁移到目标中的默认表空间。例如，假设您的源是 Oracle 之外的数据库引擎。所有目标表和索引都迁移到相同的默认表空间。

要覆盖此行为，请提供相应的表空间转换。例如，假设您要将表和索引迁移到 Oracle 目标中的表和索引表空间，这些表空间以源中的架构命名。在此情况下，您可以使用与下面类似的转换。在这里，源中的架构名为 INVENTORY，目标中的相应表和索引表空间分别名为 INVENTORYTBL 和 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"
}
```

有关转换的更多信息，请参阅 [使用 JSON 指定表选择和转换规则](#)。

如果 Oracle 同时作为源和目标，则可以通过设置 Oracle 源额外连接属性 `enableHomogenousTablespace=true` 来保留现有的表或索引表空间分配。有关更多信息，请参见 [使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。

## 升级复制实例的版本

AWS 定期发布 AWS DMS 复制引擎软件的新版本以提供新功能和增强性能。每个版本的复制引擎软件都有自己的版本号。在将 AWS DMS 复制实例升级到更高版本之前，必须测试运行生产工作负载的复制实例的现有版本。有关可用版本升级的更多信息，请参阅 [AWS DMS 发行说明](#)。

## 了解您的迁移成本

AWS Database Migration Service 可协助您以低成本轻松安全地将数据库迁移到 AWS。您只需为复制实例和任何额外的日志存储空间付费。每个数据库迁移实例都包括足以容纳交换空间的存储空间、复制日志和用于大多数复制的数据缓存，并且入站数据传输是免费的。

在初始加载期间或加载高峰期，您可能需要更多资源。您可以使用 CloudWatch 指标密切监控复制实例的资源利用率。然后，您可以根据使用情况纵向扩展和缩减复制实例的大小。

有关估算迁移成本的更多信息，请参阅：

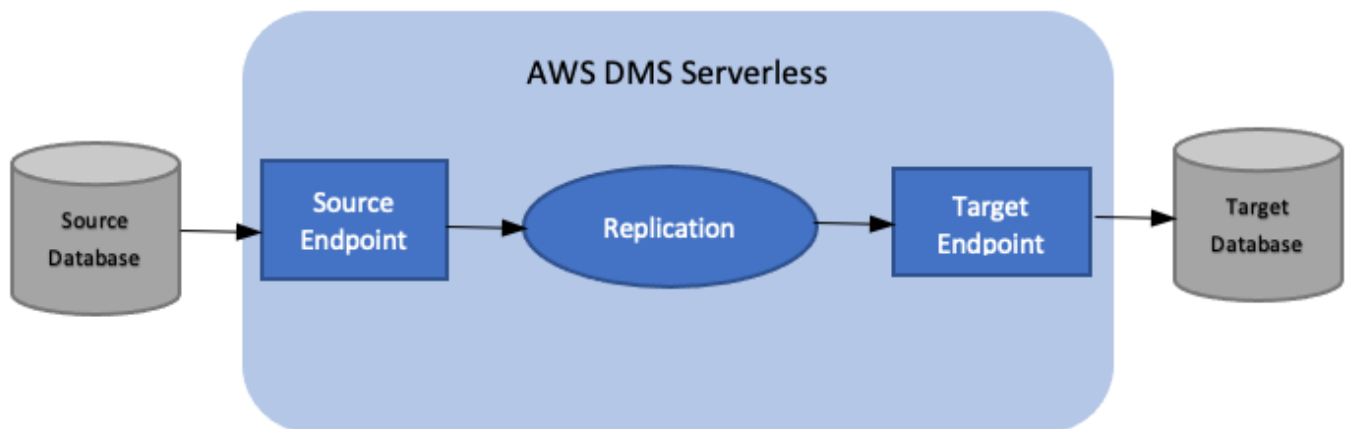
- [AWS Database Migration Service 定价](#)
- [AWS 定价计算器](#)

# 使用无 AWS DMS 服务器

AWS DMS Serverless 是一项提供自动配置、扩展、内置高可用性和 pay-for-use 计费模式的功能，可提高运营灵活性并优化成本。无服务器功能消除了复制实例管理任务，例如容量估计、预置、成本优化以及管理复制引擎版本和修补。

使用 AWS DMS Serverless，与当前的功能 AWS DMS（在本文档中称为“AWS DMS 标准”）类似，您可以使用端点创建源连接和目标连接。创建源端点和目标端点后，您可以创建复制配置，其中包括给定复制的配置设置。您可以通过启动、停止、修改或删除复制来管理复制。每个复制都具有设置，您可以根据数据库迁移需求进行配置。您可以使用 JSON 文件或 AWS DMS 部分来指定这些设置。有关复制设置的更多信息，请参阅[使用 AWS DMS 终端节点](#)。开始复制后，AWS DMS Serverless 将连接到源数据库，然后收集数据库元数据以分析复制工作负载。使用此元数据 AWS DMS 计算和配置所需的容量，然后开始数据复制。

下图显示了 AWS DMS 无服务器复制过程。



## Note

AWS DMS 无服务器使用默认引擎版本。有关默认引擎版本的更多信息，请参阅[发布说明](#)。

查看以下主题，了解有关 AWS DMS Serverless 的更多详细信息。

## 主题

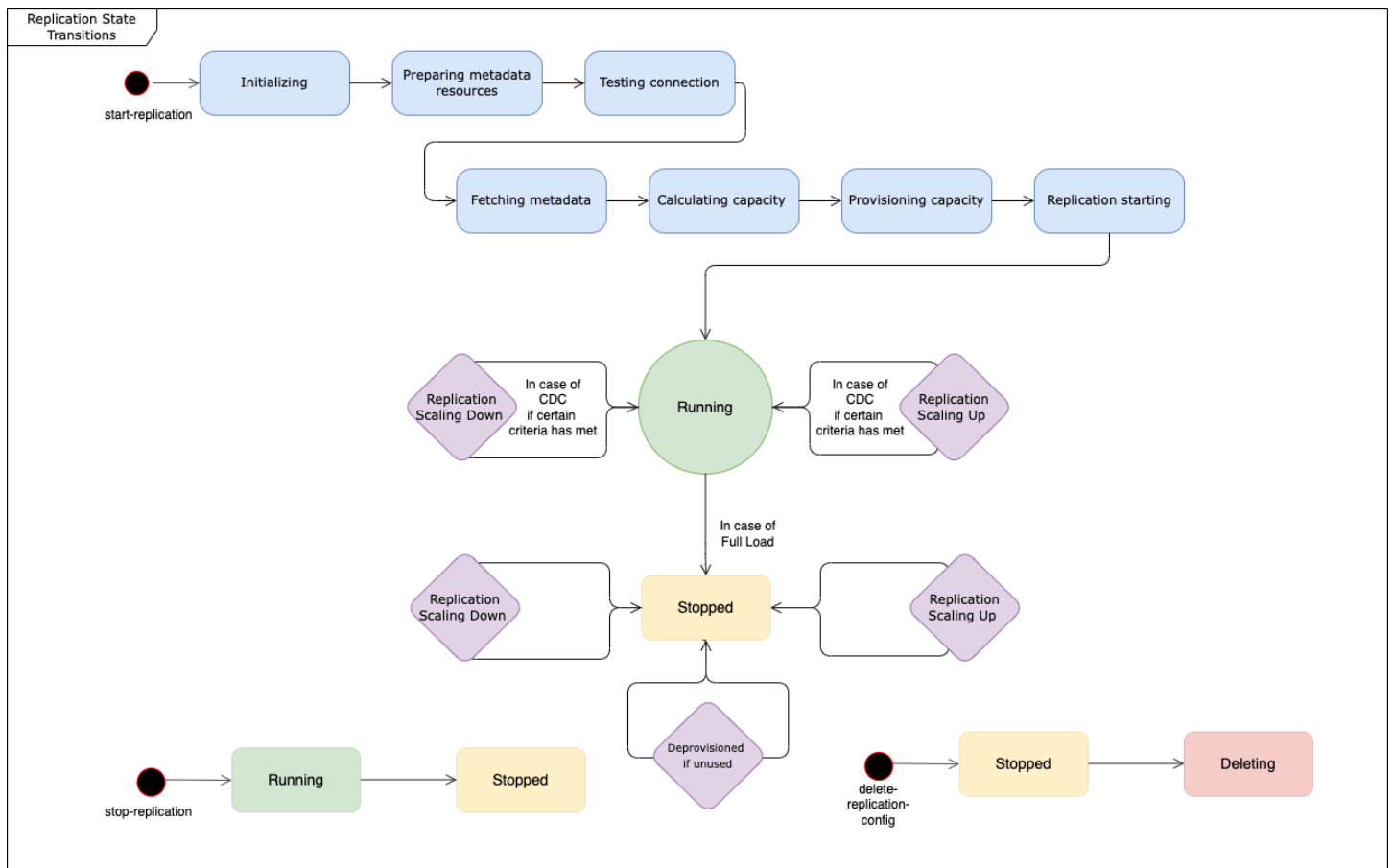
- [AWS DMS 无服务器组件](#)

- [AWS DMS 无服务器限制](#)

## AWS DMS 无服务器组件

为了管理执行复制所需的资源，AWS DMS Serverless 具有精细的状态，可以显示服务采取的不同内部操作。启动复制时，AWS DMS Serverless 会根据以下复制状态计算容量负载，预置计算出的容量，然后开始数据复制。

下图显示了 AWS DMS 无服务器复制的状态转换。



- 在您开始复制后，第一个状态是正在初始化。在此状态下将初始化所有必需的参数。
- 紧随其后的状态包括正在准备元数据资源、正在测试连接和正在获取元数据。在这些状态下，AWS DMS Serverless 会连接到您的源数据库以获取预测所需容量所需的信息。
  - 当复制状态为“测试连接”时，AWS DMS Serverless 会验证与源数据库和目标数据库的连接是否已成功建立。
  - 正在测试连接之后的复制状态为正在获取元数据。在这里，AWS DMS 检索计算容量所需的信息。

- AWS DMS 检索到必要信息后，下一个状态为“计算容量”。此时，系统计算执行复制所需的底层资源的大小。
- 在正在计算容量之后，状态转换为正在预置容量。当复制处于这种状态时，AWS DMS Serverless 会初始化底层计算资源。
- 成功预置所有资源后的复制状态为复制开始。在这种状态下，AWS DMS Serverless 开始复制数据。复制的阶段包括以下几个阶段：
  - 满载：在此阶段，DMS 将复制源数据存储库，就像复制开始时一样。
  - CDC ( 初始 )：在此阶段，DMS 会复制在满载阶段对源数据存储所做的更改。仅当 `StopTaskCachedChangesNotApplied` 任务设置为 `false` 时，DMS 才会运行此阶段。
  - CDC ( 进行中 )：在最初的 CDC 阶段之后，DMS 会在源数据库上复制所发生的更改。只有在 `StopTaskCachedChangesApplied` 任务设置为 CDC 初始阶段之后，DMS 才会继续运行复制。`false`
- 最终状态为正在运行。在正在运行状态下，数据复制正在进行中。
- 您停止的复制进入已停止状态。在以下情况下，您可以重新启动已停止的复制：
  - 您无法重新启动 DMS 已取消配置的复制。
  - 您可以使用操作重新启动已停止的仅限 CDC 或满载复制以及 CDC 复制。[StartReplication](#) 您无法使用控制台重新启动已停止的复制。
  - 您无法重启使用 PostgreSQL 作为引擎的已停止复制。

本主题包含下列部分：

- [支持的引擎版本](#)
- [创建无服务器复制](#)
- [修改 AWS DMS 无服务器复制](#)
- [Compute Config](#)
- [了解无服务器中的 AWS DMS 自动扩展](#)
- [监控 AWS DMS 无服务器复制](#)
- [提高了 Oracle 到 Amazon Redshift 的满负荷迁移的吞吐量](#)

对于 AWS DMS 无服务器，AWS DMS 控制台的左侧导航面板有一个新选项，即无服务器复制。对于无服务器复制，您可以指定复制，而不是指定复制实例类型或任务来定义复制。此外，您可以指定希望 DMS 为复制预置的最大和最小 DMS 容量单位 ( DCU )。一个 DCU 是 2GB 的内存。AWS

DMS 为您的复制当前使用的每个 DCU 向您的账户收费。有关 AWS DMS 定价的信息，请参阅 [AWS Database Migration Service 定价](#)。

AWS DMS 然后根据您的表映射和预测的工作负载大小自动配置复制资源。此容量单位是在您指定的最小和最大容量单位值范围内的值。

## 支持的引擎版本

使用 AWS DMS Serverless，您无需选择和管理引擎版本，因为该服务会处理该设置。AWS DMS 无服务器支持以下来源：

- Microsoft SQL Server
- PostgreSQL 兼容数据库
- MySQL 兼容数据库
- MariaDB
- Oracle
- IBM Db2

AWS DMS 无服务器支持以下目标：

- Microsoft SQL Server
- PostgreSQL
- MySQL 兼容数据库
- Oracle
- Amazon S3
- Amazon Redshift
- Amazon DynamoDB
- Amazon Kinesis Data Streams
- Amazon Managed Streaming for Apache Kafka
- 亚马逊 OpenSearch 服务
- Amazon DocumentDB ( 与 MongoDB 兼容 )
- Amazon Neptune

作为 AWS DMS Serverless 的一部分，您可以访问控制台命令，这些命令允许您创建、配置、启动和管理 AWS DMS 无服务器复制。要使用控制台的无服务器复制部分运行这些命令，您需要执行以下操作之一：

- 设置新的 AWS Identity and Access Management (IAM) 策略和 IAM 角色以附加该策略。
- 使用 AWS CloudFormation 模板提供所需的访问权限。

AWS DMS Serverless 需要您的账户中存在服务关联角色 (SLR)。AWS DMS 管理此角色的创建和使用。有关确保您拥有所需 SLR 的更多信息，请参阅[AWS DMS Serverless 的服务相关角色](#)。

## 创建无服务器复制

要在两个现有 AWS DMS 端点之间创建无服务器复制，请执行以下操作。有关创建 AWS DMS 终端节点的信息，请参阅[创建源和目标终端节点](#)。

### 创建无服务器复制

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格上，选择无服务器复制，然后选择创建复制。
3. 在创建复制页面上，指定您的无服务器复制配置：

选项	操作
名称	输入用于标识复制的名称，例如 <b>DMS-replication</b> 。
描述性 Amazon 资源名称 ( ARN ) – 可选	您可以使用此可选参数来提供复制的描述。
源数据库端点	选择账户中的现有端点。请注意，AWS DMS Serverless 仅支持 AWS DMS 标准版支持的端点类型的子集。
目标数据库端点	选择账户中的现有端点。请注意，AWS DMS Serverless 仅支持 AWS DMS 标准版支持的端点类型的子集。

选项	操作
复制类型	<p>根据您的要求选择复制类型：</p> <ul style="list-style-type: none"> <li>• 满载：仅 AWS DMS 迁移现有数据。</li> <li>• 满载和更改数据捕获 (CDC)：AWS DMS 迁移现有数据和复制期间发生的更改。</li> <li>• 更改数据捕获 (CDC)：AWS DMS 仅迁移开始复制后发生的更改。</li> </ul>

在设置部分中，设置复制所需的设置。

在表映射部分中，设置表映射以定义选择和筛选所要复制的数据的规则。在指定映射之前，请确保查看有关源数据库和目标数据库之间的数据类型映射的文档部分。有关源数据库和目标数据库的数据类型映射的信息，请参阅[使用 AWS DMS 端点](#)主题中源和目标端点类型的“数据类型”部分。

在计算设置部分中，设置以下设置。有关 Compute Config 设置的信息，请参阅[Compute Config](#)。

选项	操作
VPC	选择现有 VPC。
子网组	选择现有子网组。
VPC 安全组	选择默认值 ( 如果尚未选择 )。
AWS KMS 密钥	选择适当的 KMS 密钥。有关 KMS 密钥的信息，请参阅 AWS Key Management Service API 参考中的 <a href="#">创建密钥</a> 。
部署	保留原样。
可用区	保留原样。
最小 DMS 容量单位 ( DCU ) – ( 可选 )	留空可使用默认值 1 DCU。

选项	操作
最大 DMS 容量单位 ( DCU )	选择 16 个 DCU。

保留维护设置不变。

#### 4. 选择创建复制。

AWS DMS 创建无服务器复制来执行迁移。

## 修改 AWS DMS 无服务器复制

要修改您的复制配置，请使用 `modify-replication-config` 操作。您只能修改处于 `CREATEDSTOPPED`、或 `FAILED` 状态的 AWS DMS 复制配置。有关该 `modify-replication-config` 操作的信息，请参阅 AWS Database Migration Service API 参考中的 [ModifyReplicationConfig](#)。

要修改无服务器复制配置，请使用 AWS Management Console

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择无服务器复制。
3. 选择您要修改的复制。下表描述了根据复制的当前状态，您可以进行的修改。

设置	描述	允许的状态
名称	您可以更改复制的名称。输入复制的名称，其中包含 8 到 16 个可打印 ASCII 字符（不含 /、" 和 @）。该名称在所选择 AWS 区域中对于您的账户应唯一。您可以选择在名称中添加一些细节，例如包括您正在执行的 AWS 区域和任务，例如： <b>west2-mysql1mysql-config1</b> 。	ReplicationState 为 <code>CREATED</code> 、 <code>STOPPED</code> 或 <code>FAILED</code> 。
源数据库端点	选择一个新的现有源端点作为复制的源。	ReplicationState 为 <code>CREATED</code> ，当



设置	描述	允许的状态
目标数据库端点	选择一个新的现有目标端点作为复制的目标。	ProvisionState 是 null 时为 FAILED。 ReplicationState 为 CREATED，当 ProvisionState 是 null 时为 FAILED。
复制类型	您可以修改无服务器复制的类型。	ReplicationState 为 CREATED，当 ProvisionState 是 null 时为 FAILED。
复制设置	您可以修改复制设置，包括目标表准备模式、复制中是否包含 LOB 列、最大 LOB 大小、验证和日志记录。有关更多信息，请参阅 <a href="#">Task settings (任务设置)</a> 。	ReplicationState 为 CREATED、STOPPED 或 FAILED。
表映射	您可以修改无服务器复制的表映射设置，包括选择规则和转换规则。有关更多信息，请参阅 <a href="#">表映射</a> 。	ReplicationState 为 CREATED、STOPPED 或 FAILED。

设置	描述	允许的状态
Compute Config	您可以修改无服务器复制的 Compute Config 设置，包括网络设置、扩展设置和维护设置。有关 Compute Config 设置的信息，请参阅 <a href="#">Compute Config</a> 。	<ul style="list-style-type: none"> <li>• 当 ReplicationState 为 CREATED、STOPPED 或 FAILED 时，您可以修改以下扩展、维护和网络设置： <ul style="list-style-type: none"> <li>• MinCapacityUnits</li> <li>• MaxCapacityUnits</li> <li>• MultiAZ</li> <li>• PreferredMaintenanceWindow</li> <li>• VpcSecurityGroupIds</li> </ul> </li> <li>• 在 ReplicationState 为 CREATED 或者为 FAILED ( 当 ProvisionState 是 null 时 ) 时，您可以修改以下网络和安全设置： <ul style="list-style-type: none"> <li>• AvailabilityZone</li> <li>• DnsNameServers</li> <li>• KmsKeyId</li> <li>• ReplicationSubnetGroupId</li> </ul> </li> </ul>

## Compute Config

您可以使用 Compute Config 参数或控制台部分配置复制预置。Compute Config 对象中的字段包括以下内容：

选项	描述
MinCapacity单位	这是要配置的 DMS 容量单位 (DCU) 的最小数量。AWS DMS 这也是自动缩放可以缩减到的最小 DCU。
MaxCapacity单位	这是 AWS DMS 可以预置的最大 DMS 容量单位 (DCU) ，具体取决于对您复制的容量预测。这也是自动缩放可以纵向扩展到的最大 DCU。
KmsKey我是	选择要用于加密复制存储和连接信息的加密密钥。如果您选择 (默认) aws/dms ，则 AWS DMS 使用与您的账户关联的默认 KMS 密钥和。AWS 区域将显示说明和您的账号以及密钥的 ARN。有关使用加密密钥的更多信息，请参阅 <a href="#">设置加密密钥并指定 AWS KMS 权限</a> 。在本教程中，选择 (默认) aws/dms。
ReplicationSubnetGroupId	在选定 VPC 中要在其中创建复制的复制子网组。如果源数据库位于 VPC 中，请选择包含源数据库的子网组作为复制的位置。有关复制子网组的更多信息，请参阅 <a href="#">创建复制子组</a> 。
VpcSecurityGroupIds	在 VPC 中创建复制实例。如果源数据库位于 VPC 中，请选择一个 VPC 安全组，该安全组提供对数据库所在的数据库实例的访问权限。
PreferredMaintenance窗口	此参数定义每周可以进行系统维护的时间范围，采用通用协调时间 (UTC)。默认值是从每个 8 小时的时间段中随机选择一个 30 分钟的窗口 AWS 区域，发生在一周中的随机某一天。
MultiAZ	设置此可选参数可在另一个可用区中创建复制的备用副本以支持失效转移。如果您打算使用更改数据捕获 (CDC) 或持续复制，建议您启用该选项。

## 了解无服务器中的 AWS DMS 自动扩展

在您配置复制并使其处于RUNNING状态后，该 AWS DMS 服务将管理底层资源的容量以适应不断变化的工作负载。此管理功能根据以下复制设置扩展复制资源：

- MinCapacityUnits
- MaxCapacityUnits

在超过利用率阈值上限一段时间后，复制会纵向扩展；当容量利用率在较长一段时间内低于容量利用率阈值下限时，复制会纵向缩减。

### Note

在满载过程中，无服务器复制无法自动缩小规模。

## 在无服务器中 AWS DMS 调整自动缩放

要调整您的复制自动缩放参数，我们建议您将设置MaxCapacityUnits为最大值，然后让 AWS DMS 管理资源的配置。建议您选择最大的 DCU 最大容量设置来适应事务量的峰值，以便从自动扩缩中获得最大的好处。如果您的复制持续使用最大 DCU，则定价计算器会显示每月的最大成本。最大 DCU 并不代表实际成本，因为您只需为使用的容量付费。

如果您的复制未在满负荷状态下使用其资源，则 AWS DMS 会逐渐取消资源配置以节省成本。但是，由于预置和取消预置资源需要时间，因此我们建议您将 MinCapacityUnits 设置为能够应对预计复制工作负载中会出现的任何突然峰值的值。这样可以防止您的复制资源不足，同时 AWS DMS 为更高的工作负载级别配置资源。

如果复制资源的预置不足，最大容量设置太低，无法满足数据需求，或者最小容量太低，无法处理复制工作负载突然激增，则您的 CapacityUtilization 指标可能会始终保持在最大值。这可能会导致您的复制失败。如果由于资源配置不足而导致复制失败，则会在复制日志中 AWS DMS 创建一个 out-of-memory 事件。如果 out-of-memory 这种情况是由于复制工作负载突然激增而发生的，则复制将自动扩展并重新启动。

## 监控 AWS DMS 无服务器复制

AWS 提供了多种用于监控您的 AWS DMS 无服务器复制和响应潜在事件的工具：

- [AWS DMS 无服务器复制指标](#)

- [AWS DMS 无服务器复制日志](#)

## AWS DMS 无服务器复制指标

无服务器复制监控包括以下统计数据的 Amazon CloudWatch 指标。这些统计数据按每个无服务器复制进行分组。

指标	单位	描述
CapacityUtilization	百分比	无服务器复制使用的内存百分比
CDC IncomingChanges	百分比	在上等待应用于目标 point-in-time 的更改事件总数。请注意，这与源终端节点的事务更改速率指标不同。此指标的数字过大 AWS DMS 通常表示无法及时应用捕获的更改，从而导致目标延迟过高。
CDC LatencySource	秒	<p>从源端点中捕获的最后一个事件与 AWS DMS 实例的当前系统时间戳之间的间隔（秒）。CDC LatencySource 表示源实例和复制实例之间的延迟。高 CDC LatencySource 意味着从源头捕获变化的过程会延迟。要确定正在进行的复制过程中的延迟，您可以与 CDC 一起查看此指标 LatencyTarget。如果 CDC LatencySource 和 CDC LatencyTarget 处于高位，请 LatencySource 先调查 CDC。</p> <p>当源和复制之间没有复制延迟时，CDC LatencySource 可以为 0。当复制尝试读取源事务日志中的下一个事件并且与上次从源读取时相比没有新事件时，CDC LatencySource 可能变为零。发生这种情况时，复制会将 CDC LatencySource 重置为 0。</p>
CDC LatencyTarget	秒	<p>在目标上等待提交的第一个事件时间戳与 AWS DMS 实例的当前系统时间戳之间的间隔（秒）。目标延迟是复制实例服务器时间，与转发到目标组件的最早未经确认的事件 ID 时间之间的差异。换句话说，目标延迟是复制实例的时间戳，与已应用但未经 TRG 端点确认的最早事件之间的时间戳之差（99%）。当 CDC LatencyTarget 为高时，它表示将变更事件应用于目标的过程已延迟。</p>

指标	单位	描述
		<p>要确定正在进行的复制过程中的延迟，您可以与 CDC 一起查看此指标LatencySource。如果 CDC 高但疾病预防控制中心LatencySource 不高，请调查LatencyTarget 是否：</p> <ul style="list-style-type: none"> <li>• 目标中没有主键或索引</li> <li>• 目标实例或复制实例中出现资源瓶颈</li> <li>• 复制和目标之间存在网络问题</li> </ul>
疾病预防控制中心Throughput Bandwidth目标	KB/秒	目标的传出数据传输速率，以每秒 KB 数为单位。CDC ThroughputBandwidth 记录采样点传输的传出数据。如果未找到网络流量，则该值为零。由于 CDC 不发布长时间运行的事务，可能不会记录网络流量。
疾病预防控制中心ThroughputRows来源	行/秒	源的传入更改速率，以每秒行数为单位。
疾病预防控制中心ThroughputRows目标	行/秒	目标的传出更改速率，以每秒行数为单位。
FullLoadThroughput Bandwidth目标	KB/秒	目标在满负载情况下的传出数据传输速率，以每秒 KB 数为单位。
FullLoadThroughputRows目标	行/秒	目标上的完全加载的传出更改，以行/秒为单位。

## AWS DMS 无服务器复制日志

在 AWS DMS 迁移过程中 CloudWatch ，您可以使用 Amazon 记录复制信息。您可以在选择复制设置时启用日志记录。

无服务器复制会将状态日志上传到您的 CloudWatch 帐户，以提高对复制进度的可见性并帮助进行故障排除。

AWS DMS 将无服务器链接的日志上传到带有前缀的专用日志组。dms-serverless-replication-*<your replication config resource ID>*在此日志组中，有一个名为的 dms-serverless-replication-orchestrator-*<your replication config resource ID>* 日志流。此日志流报告您的复制的复制状态，并显示一条相关消息，提供有关其在此阶段所做工作的更多详细信息。有关日志条目的示例，请参阅下文的 [无服务器复制日志示例](#)。

#### Note

AWS DMS 在运行复制之前，不会创建日志组或流。AWS DMS 如果您只创建复制，则不会创建日志组或流。

要查看所运行复制的日志，请按照下列步骤操作：

1. 打开 AWS DMS 控制台，然后从导航窗格中选择无服务器复制。将出现无服务器复制对话框。
2. 转到配置部分，然后在“常规”列中选择查看无服务器日志。CloudWatch 日志组打开。
3. 找到“迁移任务日志”部分，然后选择“查看 CloudWatch 日志”。

如果复制失败，则 AWS DMS 会创建一个复制状态为的日志条目 failed，并显示一条描述失败原因的消息。作为排除复制失败故障的第一步，您应该检查 CloudWatch 日志。

#### Note

与 AWS DMS Classic 一样，您可以选择对数据迁移本身的进度（即底层复制任务发出的日志）启用更精细的日志记录。您可以在复制设置中，通过将 Logging 字段中的 EnableLogging 设置为 true 来启用这些日志，如以下 JSON 示例所示：

```
{
  "Logging": {
    "EnableLogging": true
  }
}
```

如果您启用这些日志，它们将仅在无服务器复制的 `running` 阶段开始出现。它们将与之前的日志流出现在同一个日志组中，但会出现在新的日志流 `dms-serverless-serv-res-id-{unique identifier}` 下。有关如何解释无服务器复制日志的信息，请参阅以下章节。

## 无服务器复制日志示例

本节包括用于无服务器复制的日志条目示例。

### 示例：复制开始

运行无服务器复制时，AWS DMS 会创建一个类似于以下内容的日志条目：

```
{'replication_state':'initializing', 'message': 'Initializing the replication workflow.'}
```

### 示例：复制失败

如果复制的其中一个端点配置不正确，则 AWS DMS 会创建一个类似于以下内容的日志条目：

```
{'replication_state':'failed', 'message': 'Test connection failed for endpoint X.', 'failure_message': 'X'}
```

如果在出现故障后，您在日志中看到此消息，请确保指定的端点运行正常且配置正确。

## 提高了 Oracle 到 Amazon Redshift 的满负荷迁移的吞吐量

AWS DMS 显著提高了从 Oracle 到 Amazon Redshift 的满负荷迁移的吞吐量性能。DMS 会自动为表映射中没有该 `custom parallel-load` 选项的表启用此功能。对于具有自定义并行加载选项的表，DMS serverless 会根据给定的表映射配置来分配表负载。要使用增强的吞吐量，请执行以下操作：

- 提供不引用分区或边界的选择规则。例如，如果表映射中的表设置包含 `parallel-load`，则 DMS Serverless 将不会使用增强吞吐量功能。有关更多信息，请参阅 [选择规则和操作](#)。
- 将 `MaxFileSize` 和设置 `WriteBufferSize` 为 64 MB。有关更多信息，请参阅 [使用 Amazon Redshift 作为 AWS DMS 的目标时的端点设置](#)。
- 对于包含稀疏数据的数据存储，`true` 对于包含密集数据的数据存储，我们建议将设置 `CompressCsvFiles` 为 `false`。



- 将以下任务设置设置为 0：
  - ParallelLoadThreads
  - ParallelLoadQueuesPerThread
  - ParallelApplyThreads
  - ParallelApplyQueuesPerThread
  - ParallelLoadBufferSize
- 设置为 MaxFullLoadSubTasks49 以支持并行数据迁移。
- 将 LOB mode 设置为 inline。有关更多信息，请参阅 [在任务中为源数据库设置 LOB 支持 AWS DMS](#)。

AWS DMS 不为以下复制提供增强的吞吐量性能：

- 使用并行加载对表进行复制。有关更多信息，请参阅 [对所选表、视图和集合使用并行加载](#)。
- 使用数据转换规则进行复制。
- 带有筛选规则的复制。
- 使用 change-data-type 转换规则进行复制。

## AWS DMS 无服务器限制

AWS DMS 无服务器有以下限制：

- 您只能修改处于 CREATEDSTOPPED、或 FAILED 状态的 AWS DMS 复制配置。有关在哪些条件下可以更改哪些设置的详细信息，请参阅 [修改 AWS DMS 无服务器复制](#)。
- 您只能删除处于 STOPPED、或 FAILED 状态的 AWS DMS 复制配置。
- 每个复制分配了 100 GB 的静态存储空间供使用。如果由于长时间运行的事务或缓存等要求，您的复制消耗的内存超过此值，我们建议您将工作负载分区为单独的无服务器复制。您可以按表或要求对工作负载进行分区，例如将涉及 LOB 的所有复制放入单独的无服务器复制中。
- 与复制实例不同，AWS DMS 无服务器复制没有用于管理任务的公有 IP 地址。您可以使用控制台管理无服务器复制。
- 此版本的 AWS DMS 无服务器不支持 AWS DMS 标准版支持的所有源端点和目标端点类型。有关受支持的引擎类型的列表，请参阅 [AWS DMS 无服务器组件](#)。
- 无服务器复制需要使用 VPC 端点访问依赖关系。您必须使用 VPC 端点访问以下端点类型：
  - Amazon S3

- Amazon Kinesis
- AWS Secrets Manager
- Amazon DynamoDB
- Amazon Redshift
- 亚马逊 OpenSearch 服务

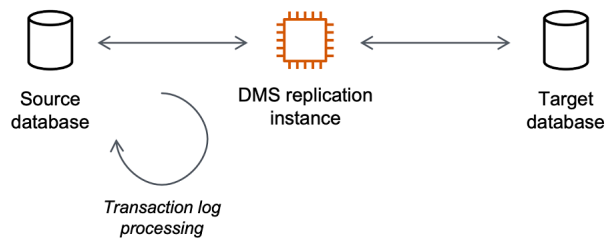
有关 VPC 端点的更多信息，请参阅[将 VPC 端点配置为 AWS DMS 源端点和目标端点](#)。

- AWS DMS serverless 不支持带有选择和转换规则的视图。
- AWS DMS 无服务器不支持使用 AWS 客户托管密钥。AWS DMS 无服务器仅支持使用默认 DMS 密钥。有关更多信息，请参阅[中的数据保护 AWS Database Migration Service](#)。
- DMS 无服务器不支持 DB2 端点的 SSL 连接。

## 使用 AWS DMS 复制实例

当您创建 AWS DMS 复制实例时，在基于亚马逊 VPC 服务的虚拟私有云 (VPC) 中的 Amazon EC2 实例上 AWS DMS 创建该实例。您可以使用此复制实例执行数据库迁移。在选择多可用区选项时，复制实例可通过多可用区部署提供高可用性和失效转移支持。

在多可用区部署中，AWS DMS 自动在不同的可用区域中预置和维护复制实例的同步备用副本。主复制实例可以跨可用区同步复制到备用副本。此方法提供数据冗余，消除 I/O 冻结，并将延迟峰值降至最小。



AWS DMS 使用复制实例连接到您的源数据存储、读取源数据并格式化数据以供目标数据存储使用。复制实例还会将数据加载到目标数据存储中。大部分这种处理发生在内存中。但是，大型事务可能需要部分缓冲到磁盘上。缓存事务和日志文件也会写入磁盘。

您可以在以下 AWS 区域创建 AWS DMS 复制实例。

区域名称	区域	端点	协议
美国东部 ( 俄亥俄州 )	us-east-2	dms.us-east-2.amazonaws.com	HTTPS
		dms-fips.us-east-2.amazonaws.com	HTTPS
美国东部 ( 弗吉尼亚州北部 )	us-east-1	dms.us-east-1.amazonaws.com	HTTPS
		dms-fips.us-east-1.amazonaws.com	HTTPS
美国西部 ( 加利福尼亚北部 )	us-west-1	dms.us-west-1.amazonaws.com	HTTPS
		dms-fips.us-west-1.amazonaws.com	HTTPS

区域名称	区域	端点	协议
美国西部 ( 俄勒冈州 )	us-west-2	dms.us-west-2.amazonaws.com	HTTPS
		dms-fips.us-west-2.amazonaws.com	HTTPS
非洲 ( 开普敦 )	af-south-1	dms.af-south-1.amazonaws.com	HTTPS
亚太地区 ( 香港 )	ap-east-1	dms.ap-east-1.amazonaws.com	HTTPS
亚太地区 ( 海得拉巴 )	ap-south-2	dms.ap-south-2.amazonaws.com	HTTPS
亚太地区 ( 雅加达 )	ap-southeast-3	dms.ap-southeast-3.amazonaws.com	HTTPS
亚太地区 ( 墨尔本 )	ap-southeast-4	dms.ap-southeast-4.amazonaws.com	HTTPS
亚太地区 ( 孟买 )	ap-south-1	dms.ap-south-1.amazonaws.com	HTTPS
亚太地区 ( 大阪 )	ap-northeast-3	dms.ap-northeast-3.amazonaws.com	HTTPS
亚太地区 ( 首尔 )	ap-northeast-2	dms.ap-northeast-2.amazonaws.com	HTTPS
亚太地区 ( 新加坡 )	ap-southeast-1	dms.ap-southeast-1.amazonaws.com	HTTPS

区域名称	区域	端点	协议
亚太地区 (悉尼)	ap-southeast-2	dms.ap-southeast-2.amazonaws.com	HTTPS
亚太地区 (东京)	ap-northeast-1	dms.ap-northeast-1.amazonaws.com	HTTPS
加拿大 (中部)	ca-central-1	dms.ca-central-1.amazonaws.com	HTTPS
加拿大西部 (卡尔加里)	ca-west-1	dms.ca-west-1.amazonaws.com	HTTPS
欧洲地区 (法兰克福)	eu-central-1	dms.eu-central-1.amazonaws.com	HTTPS
欧洲地区 (爱尔兰)	eu-west-1	dms.eu-west-1.amazonaws.com	HTTPS
欧洲地区 (伦敦)	eu-west-2	dms.eu-west-2.amazonaws.com	HTTPS
欧洲地区 (米兰)	eu-south-1	dms.eu-south-1.amazonaws.com	HTTPS
欧洲地区 (巴黎)	eu-west-3	dms.eu-west-3.amazonaws.com	HTTPS
欧洲(西班牙)	eu-south-2	dms.eu-south-2.amazonaws.com	HTTPS
欧洲地区 (斯德哥尔摩)	eu-north-1	dms.eu-north-1.amazonaws.com	HTTPS

区域名称	区域	端点	协议
欧洲 ( 苏黎世 )	eu-centra l-2	dms.eu-central-2.amazonaws.com	HTTPS
以色列 ( 特拉维夫 )	il-centra l-1	dms.il-central-1.amazonaws.com	HTTPS
中东 ( 巴林 )	me- south-1	dms.me-south-1.amazonaws.com	HTTPS
中东 ( 阿联酋 )	me- central-1	dms.me-central-1.amazonaws.com	HTTPS
南美洲 ( 圣保罗 )	sa-east-1	dms.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud ( 美国东部 )	us-gov- east-1	dms.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud ( 美国西部 )	us-gov- west-1	dms.us-gov-west-1.amazonaws.com	HTTPS

AWS DMS 支持名为的特殊 AWS 区域 AWS GovCloud (US) ，该区域旨在允许美国政府机构和客户将敏感工作负载迁移到云中。AWS GovCloud (US) 满足了美国政府的具体监管和合规要求。有关的更多信息 AWS GovCloud (US) ，请参阅[什么是 AWS GovCloud \( 美国 \) ?](#)

接下来，您可以了解有关复制实例的更多详细信息。

## 主题

- [为您的迁移选择正确的 AWS DMS 复制实例](#)
- [选择复制实例的最佳大小](#)

- [使用复制引擎版本](#)
- [公有和私有复制实例](#)
- [IP 地址和网络类型](#)
- [为复制实例设置网络](#)
- [为复制实例设置加密密钥](#)
- [创建复制实例](#)
- [修改复制实例](#)
- [重启复制实例](#)
- [删除复制实例](#)
- [使用 AWS DMS 维护时段](#)

## 为您的迁移选择正确的 AWS DMS 复制实例

AWS DMS 在 Amazon EC2 实例上创建复制实例。AWS DMS 目前支持复制实例的 T2、T3、C4、C5、C6i、R4、R5 和 R6i Amazon EC2 实例类别：

- T2 实例是具爆发能力的通用实例，此类实例具有基准水平的 CPU 性能并能够提升到基准水平之上。基准性能和突增能力由 CPU 积分控制。T2 实例以设定的速率持续获得 CPU 积分，具体取决于实例大小。它们在空闲时累积 CPU 积分，在处于活动状态时消耗 CPU 积分。

T2 实例非常适合各种通用工作负载。这些工作负载包括微服务、低延迟交互式应用程序、中小型数据库、虚拟桌面、开发、构建和暂存环境、代码存储库以及产品原型。

- T3 实例是下一代可突增性能通用实例类型。此类型提供基准级别的 CPU 性能，并能够根据需要随时突增 CPU 使用率。T3 实例兼顾了计算、内存和网络资源，非常适合 CPU 使用中等、偶尔遇到峰值的应用程序。当工作负载的运行低于基准阈值时，T3 实例会累积 CPU 积分。每获得一个 CPU 积分，T3 实例就有机会在需要时为完整 CPU 核心突增一分钟内的性能。

T3 实例可在 unlimited 模式下突增到所需的时间。有关 unlimited 模式的更多信息，请参阅[对可突增性能实例使用无限模式](#)。

- C4 实例针对计算密集型工作负载进行了优化，能够以较低的单位计算价格提供非常经济高效的高性能。它们显著提高了每秒数据包数 (PPS) 性能、更低的网络抖动和更低的网络延迟。AWS DMS 也可能占用 CPU 密集型，尤其是在执行异构迁移和复制（例如从 Oracle 迁移到 PostgreSQL）时。针对这些情况，C4 实例是理想选择。
- C5 实例是下一代实例类型，可以较低的单位计算价格，为运行高级计算密集型工作负载提供经济高效的高性能。这包括高性能 Web 服务器、高性能计算 (HPC)、批处理、广告服务、高度可扩展的

多人游戏和视频编码等工作负载。适合 C5 实例的其他工作负载包括科学建模、分布式分析以及机器学习和深度学习推理。C5 实例可选择 Intel 和 AMD 的处理器。

- 与同类第 5 代实例相比，C6i 实例为各种工作负载提供的计算性价比高出 15%，而且内存加密始终处于开启状态。C6i 实例非常适合计算密集型工作负载，例如批处理、分布式分析、高性能计算（HPC）、广告服务、高度可扩展的多人游戏和视频编码。
- R4 实例针对内存密集型工作负载优化了内存。使用 AWS DMS 持续迁移或复制高吞吐量事务系统也会占用大量 CPU 和内存。与早代实例类型相比，R4 实例的每个 vCPU 包含更多内存。
- R5 实例是适用于 Amazon EC2 的下一代内存优化型实例类型。R5 实例非常适合内存密集型应用程序，例如高性能数据库、分布式 Web 级内存缓存、中型内存数据库、实时大数据分析和其他企业应用程序。使用高吞吐量事务系统的持续迁移或复制 AWS DMS 也会消耗大量 CPU 和内存。
- 与同类第 5 代实例相比，R6i 实例为各种工作负载提供的计算性价比高出 15%，而且内存加密始终处于开启状态。R6i 实例已通过 SAP 认证，非常适合 SQL 和 NoSQL 数据库等工作负载、Memcached 和 Redis 等分布式网络规模内存缓存、SAP HANA 等内存数据库以及 Hadoop 和 Spark 集群等实时大数据分析。

每个复制实例都有特定的内存和 vCPU 配置。下表显示了各复制实例类型的配置。有关定价的信息，请参阅 [AWS Database Migration Service 服务定价页面](#)。

### 通用复制实例类型

类型	vCPU	内存 (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



## 计算优化复制实例类型

类型	vCPU	内存 (GiB)
dms.c4.large	2	3.75
dms.c4.xlarge	4	7.5
dms.c4.2xlarge	8	15
dms.c4.4xlarge	16	30
dms.c5.large	2	4
dms.c5.xlarge	4	8
dms.c5.2xlarge	8	16
dms.c5.4xlarge	16	32
dms.c5.9xlarge	36	72
dms.c5.12xlarge	48	96
dms.c5.18xlarge	72	144
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

类型	vCPU	内存 (GiB)
dms.c6i.24xlarge	96	192
dms.c6i.32xlarge	128	256

### 内存优化复制实例类型

类型	vCPU	内存 (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

类型	vCPU	内存 (GiB)
dms.r6i.2xlarge	8	64
dms.r6i.4xlarge	16	128
dms.r6i.8xlarge	32	256
dms.r6i.12xlarge	48	384
dms.r6i.16xlarge	64	512
dms.r6i.24xlarge	96	768
dms.r6i.32xlarge	128	1024

上表列出了所有 AWS DMS 复制实例类型，但您所在地区的可用类型可能会有所不同。要查看您所在区域的可用复制实例类型，可以运行以下 [AWS CLI](#) 命令：

```
aws dms describe-orderable-replication-instances --region your_region_name
```

## 主题

- [决定使用哪个实例类](#)
- [对可突增性能实例使用无限模式](#)

## 决定使用哪个实例类

为了帮助确定哪个复制实例类最适合您，让我们来看看 AWS DMS 使用的变更数据捕获 (CDC) 流程。

假设您运行的是完全加载加 CDC 任务（批量加载加持续复制）。在这种情况下，任务有自己的 SQLite 存储库来存储元数据和其他信息。在 AWS DMS 开始满负荷之前，需要执行以下步骤：

- AWS DMS 开始捕获它正在从源引擎的事务日志中迁移的表的更改（我们称之为缓存的更改）。完全加载完成后，将收集这些缓存的更改并将其应用于目标中。根据缓存更改的量，可以直接从内存中应用这些更改，最初在内存中收集更改，并一直收集到设定的阈值。或者，也可以从内存中无法保存更改时将更改写入到的磁盘中应用更改。
- 应用缓存的更改后，默认情况下 AWS DMS 会在目标实例上启动事务性应用进程。

在应用缓存更改阶段和正在进行的复制阶段，AWS DMS 使用两个流缓冲区，每个缓冲区用于传入和传出数据。AWS DMS 还使用了一个名为分拣器的重要组件，它是另一个内存缓冲区。下面是排序器组件的两个重要用法 (还有其他用法)：

- 它跟踪所有事务并确保只将相关事务转发至传出缓冲区。
- 它确保按照与源中相同的提交顺序转发事务。

如您所见，此架构中有三个重要的内存缓冲区用于 AWS DMS 中的 CDC。如果任何缓冲区遇到内存压力，迁移可能会出现导致故障的性能问题。

在将每秒事务数 (TPS) 很高的重型工作负载插入此架构时，您会发现 R5 和 R6i 实例提供的额外内存很有用。您可以使用 R5 和 R6i 实例在内存中保存大量事务，并可避免在持续复制期间出现内存压力问题。

## 对可突增性能实例使用无限模式

配置为 `unlimited` 的可突增性能实例 (例如 T3 实例) 可以承受所需的任何时段的高 CPU 利用率。每小时实例价格可以自动涵盖所有 CPU 使用峰值。如果在滚动 24 小时或实例生命周期 (以较短者为准) 内实例的平均 CPU 使用率等于或低于基准，则会这样处理。

对于绝大多数通用型工作负载，配置为 `unlimited` 的实例可提供足够高的性能，而不会收取任何额外的费用。如果实例长时间以较高的 CPU 利用率运行，可能会按每 vCPU 小时的固定费率收取额外的费用。有关 T3 实例定价的信息，请参阅 [AWS Database Migration Service](#) 中的“T3 CPU 积分”。

有关 T3 实例 `unlimited` 模式的更多信息，请参阅 Amazon EC2 用户指南中的 [突发性能实例的无限模式](#)。

### Important

如果您使用的 `dms.t3.micro` 实例享受 [AWS 免费套餐](#) 优惠，并在 `unlimited` 模式下使用该实例，可能会收取相应的费用。特别是，如果连续 24 小时内的平均利用率超过实例的基准利用率，可能需要收取相应费用。有关更多信息，请参阅 Amazon EC2 用户指南中的 [基准利用率](#)。

T3 实例默认情况下作为 `unlimited` 启动。如果滚动 24 小时内的平均 CPU 使用率超过基准，将会产生超额积分费用。在某些情况下，您可以使用 `unlimited` 模式启动 T3 竞价型实例，并计划立即短时间使用它们。如果您在没有空闲时间来累积 CPU 积分的情况下这样做，则会产生超额积分费用。建议您使用标准模式启动 T3 竞价型实例，以避免支付更高的费用。

有关更多信息，请参阅 Amazon EC2 用户指南中的[剩余积分可能会产生费用](#)、[T3 竞价型实例](#)和[突发性能实例的标准模式](#)。

## 选择复制实例的最佳大小

如何选择合适的复制实例取决于与您的使用案例相关的多种因素。为帮助您了解如何使用复制实例资源，请参阅以下讨论。其中涵盖了完全加载 + CDC 任务的常见场景。

在满负荷任务期间，单独 AWS DMS 加载表。默认情况下，一次加载八个表。AWS DMS 捕获满载任务期间对源的持续更改，以便稍后可以在目标端点上应用这些更改。更改缓存在内存中；如果可用内存用完，则将更改缓存到磁盘上。当表的满载任务完成后，会 AWS DMS 立即将缓存的更改应用于目标表。

在应用表的所有未完成的缓存更改后，目标终端节点将处于事务一致状态。此时，目标与源端点在上次缓存的更改方面保持同步。AWS DMS 然后开始在源和目标之间进行持续复制。为此，AWS DMS 需要从源事务日志中提取变更操作，并以事务一致的方式将其应用于目标。（此过程假设未选择批量优化应用）。AWS DMS 如果可能，将正在进行的更改流式传输到复制实例上的内存。否则，AWS DMS 会将更改写入复制实例上的磁盘，直到它们可以应用到目标上。

您对复制实例如何应对更改处理以及在该流程中如何使用内存有一定的控制能力。有关如何优化更改处理的更多信息，请参阅[更改处理优化设置](#)。

## 需要考虑的因素

内存和磁盘空间是为使用案例选择合适的复制实例的关键因素。接下来，您可以找到有关使用案例特征的讨论，用于分析和选择复制实例。

- 数据库和表大小

数据卷有助于确定可优化完全加载性能的任务配置。例如，对于两个 1 TB 的架构，您可以将表分成四个 500 GB 的任务，然后并行运行这些表。可能的并行度取决于复制实例中可用的 CPU 资源。因此，最好了解数据库和表的大小以优化完全加载性能。它有助于确定您可以执行的任务数。

- 大型对象

迁移范围内涵盖的数据类型可能会影响性能。尤其是大型对象（LOB），此类对象会影响性能和内存消耗。要迁移 LOB 值，请 AWS DMS 执行两步过程。首先，在目标中 AWS DMS 插入不带 LOB 值的行。其次，使用 LOB 值 AWS DMS 更新该行。这会对内存产生影响，因此识别源中的 LOB 列并分析其大小非常重要。

- 加载频率和事务大小

加载频率和每秒事务数 ( TPS ) 会影响内存使用情况。大量的 TPS 或数据操作语言 ( DML ) 活动会导致内存使用率高。之所以发生这种情况，是因为 DMS 会缓存更改，直到将它们应用于目标。在 CDC 期间，这会执行交换 ( 由于内存溢出而写入物理磁盘 ) ，从而导致延迟。

- 表键和引用完整性

有关表密钥的信息决定了用于迁移数据的 CDC 模式 ( 批量应用或事务应用 ) 。通常，事务应用比批量应用慢。对于长时间运行的事务，可能需要迁移许多更改。使用事务性应用时，与批处理应用相比，AWS DMS 可能需要更多的内存来存储更改。如果迁移没有主键的表，批量应用将失败，DMS 任务将切换至事务应用模式。如果在 CDC 期间表之间的引用完整性处于活动状态，则默认 AWS DMS 使用事务应用。有关批量应用与事务应用的更多对比信息，请参阅[如何使用 DMS 批量应用功能来提高 CDC 复制性能？](#)。

使用这些指标来确定您需要对复制实例进行计算优化还是内存优化。

## 常见问题

在迁移过程中，您可能会遇到以下导致复制实例资源争用的常见问题。有关复制实例指标的信息，请参阅[复制实例指标](#)。

- 如果复制实例的内存不足，则会导致向磁盘写入数据。从磁盘读取数据可能会导致延迟，您可以通过使用足够的内存调整复制实例的大小来避免延迟。
- 分配给复制实例的磁盘大小可以小于所需的容量。当内存中的数据溢出时，会使用磁盘大小；它也用于存储任务日志。最大 IOPS 也取决于它。
- 运行多个任务或高度并行的任务会影响复制实例的 CPU 消耗。这会减慢任务的处理速度并导致延迟。

## 最佳实践

在调整复制实例规模时，请考虑这两种最常见的最佳做法。有关更多信息，请参阅[AWS Database Migration Service 的最佳做法](#)。

1. 调整工作负载大小，了解工作负载是计算机密集型还是内存密集型。在此基础上，您可以确定复制实例的类和大小：
  - AWS DMS 处理内存中的 LOB。此操作需要占用相当多的内存。

- 任务数和线程数会影响 CPU 消耗。在完全加载运行期间，避免使用超过八个 MaxFullLoadSubTasks。
2. 当您在完全加载期间工作负载较高时，请增加分配给复制实例的磁盘空间。这样做可以让复制实例使用为其分配的最大 IOPS。

前面的准则并未涵盖所有可能的场景。在确定复制实例的大小时，考虑特定使用案例的具体细节至关重要。

前面的测试表明，CPU 和内存因工作负载不同而异。特别是，LOB 会影响内存，而任务计数或并行度会影响 CPU。开始运行迁移之后，请监控 CPU、可释放内存、空闲存储以及复制实例的 IOPS。根据您的数据，您可以视根据增加或减小复制实例的大小。

## 使用复制引擎版本

复制引擎是在您的复制实例上运行并执行您指定的迁移任务的核心 AWS DMS 软件。AWS 定期发布具有新功能和性能改进的 AWS DMS 复制引擎软件的新版本。每个版本的复制引擎软件具有自己的版本号以区别于其他版本。

除非您另行指定，否则当您启动新的复制实例时，它将运行最新的 AWS DMS 引擎版本。有关更多信息，请参阅 [使用 AWS DMS 复制实例](#)。

如果您的复制实例当前正在运行，则可以将其升级到更新的引擎版本。（AWS DMS 不支持引擎版本降级。）有关复制引擎版本的更多信息，请参阅 [AWS DMS 发行说明](#)。

## 使用控制台升级引擎版本

您可以使用升级 AWS DMS 复制实例 AWS Management Console。

### 使用控制台升级复制实例

1. 打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择复制实例。
3. 选择您的复制引擎，然后选择修改。
4. 对于引擎版本，请选择所需的版本号，然后选择修改。

**Note**

我们建议您在升级复制实例之前停止所有任务。如果您不停止任务，AWS DMS 将在升级之前自动停止任务。如果您手动停止任务，则需在升级完成后手动启动任务。复制实例升级需要几分钟的时间。在实例准备就绪后，其状态将变为可用。

## 使用升级引擎版本 AWS CLI

您可以使用升级 AWS DMS 复制实例 AWS CLI，如下所示。

要升级复制实例，请使用 AWS CLI

1. 使用以下命令确定您的复制实例的 Amazon 资源名称 (ARN)。

```
aws dms describe-replication-instances \  
--query "ReplicationInstances[*].\  
[ReplicationInstanceIdentifier,ReplicationInstanceArn,ReplicationInstanceClass]"
```

在输出中，记下要升级的复制实例的 ARN，例如：`arn:aws:dms:us-east-1:123456789012:rep:6EFQQ06U6EDPRCPKLNPL2SCEEY`

2. 使用以下命令确定可用的复制实例版本。

```
aws dms describe-orderable-replication-instances \  
--query "OrderableReplicationInstances[*].[ReplicationInstanceClass,EngineVersion]"
```

在输出中，记下适用于您的复制实例类的引擎版本号。您应该会在步骤 1 的输出中看到此信息。

3. 使用以下命令升级复制实例。

```
aws dms modify-replication-instance \  
--replication-instance-arn arn \  
--engine-version n.n.n
```

将前面的 *arn* 替换为上一步中的实际复制实例 ARN。

将 *n.n.n* 替换为所需的引擎版本号，例如：`3.4.5`。



**Note**

复制实例升级需要几分钟的时间。您可以使用以下命令查看复制实例状态。

```
aws dms describe-replication-instances \  
--query "ReplicationInstances[*].\  
[ReplicationInstanceIdentifier,ReplicationInstanceStatus]"
```

在复制实例准备就绪后，其状态将变为可用。

## 公有和私有复制实例

您可以指定复制实例在连接到源和目标数据库时使用公有还是私有 IP 地址。

私有复制实例具有无法从复制网络之外访问的私有 IP 地址。当源数据库和目标数据库位于连接到复制实例的虚拟私有云 (VPC) 的同一网络中时，您可以使用私有实例。可以使用虚拟专用网络 (VPN) 或 VPC 对等连接将网络连接到 VPC。AWS Direct Connect

VPC 对等连接是两个 VPC 之间的网络连接。此连接允许使用每个 VPC 的私有 IP 地址进行路由，就像位于同一个网络中一样。有关 VPC 对等连接的更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 对等连接](#)。

公有复制实例可以使用复制实例的 VPC 安全组，以及复制实例的公有 IP 地址或 NAT 网关的公有 IP 地址。这些连接构成了您用于数据迁移的网络。

## IP 地址和网络类型

AWS DMS 始终在亚马逊虚拟私有云 (VPC) 中创建您的复制实例。创建 VPC 时，您可以确定要使用的 IP 地址：IPv4 或 IPv6，或者两者兼而有之。然后，在创建或修改复制实例时，您可以使用双堆栈模式来指定使用 IPv4 地址协议或 IPv6 地址协议。

### IPv4 地址

在创建 VPC 时，您必须以无类别域间路由 (CIDR) 块的形式为 VPC 指定 IPv4 地址范围；例如，10.0.0.0/16。子网组定义此 CIDR 块中的 IP 地址范围。这些 IP 地址可以是私有地址，也可以是公有地址。

私有 IPv4 地址是指无法通过 Internet 访问的 IP 地址。您可以使用私有 IPv4 地址在位于同一 VPC 中的复制实例和其他资源（如 Amazon EC2 实例）之间进行通信。每个复制实例都有一个用于在 VPC 中通信的私有 IP 地址。

公有 IP 地址是指可通过 Internet 访问的 IPv4 地址。您可以使用公用地址在您的复制实例和 Internet 上的资源之间进行通信。您可以控制复制实例是否接收公有 IP 地址。

## 双堆栈模式和 IPv6 地址

如果您有必须通过 IPv6 与复制实例通信的资源，请使用双堆栈模式。要使用双堆栈模式，请确保与复制实例关联的子网组中的每个子网都具有与之关联的 IPv6 CIDR 块。您可以创建新的复制子网组或修改现有复制子网组来满足此要求。每个 IPv6 地址都是全局唯一的。我们将自动从 Amazon 的 IPv6 地址池中为您的 VPC 分配 IPv6 CIDR 块。您不能自行选择范围。

DMS 禁止互联网网关访问私有双堆栈模式复制实例的 IPv6 端点。DMS 这样做是为了确保您的 IPv6 端点是私有的，并且只能从您的 VPC 内部对其进行访问。

您可以使用 AWS DMS 控制台创建或修改复制实例，并在网络类型部分指定双堆栈模式。下图显示了控制台中的 Network type（网络类型）部分。

### 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.

## 参考信息

- 有关 IPv4 和 IPv6 地址的更多信息，请参阅《Amazon VPC 用户指南》中的 [IP 寻址](#)。
- 有关使用双堆栈模式创建复制实例的更多信息，请参阅 [创建复制实例](#)。
- 有关修改复制实例的更多信息，请参阅 [修改复制实例](#)。

## 为复制实例设置网络

AWS DMS 始终在基于 Amazon VPC 的 VPC 中创建复制实例。您可以指定复制实例所在的 VPC。您可以为您的账户和 AWS 地区使用默认 VPC，也可以创建新的 VPC。

确保为复制实例的 VPC 分配的弹性网络接口与安全组关联。此外，请确保此安全组的规则允许所有端口上的所有流量流出（出口）VPC。使用此方法，如果端点上启用了正确的传出规则，就允许从复制实例与您的源和目标数据库端点通信。我们建议您为终端节点使用默认设置，这允许在所有端口上传出到所有地址。

源和目标终端节点连接到 VPC 或者位于 VPC 内部，以此来访问位于 VPC 内部的复制实例。数据库终端节点必须包含网络访问控制列表 (ACL) 安全组规则 (在适用时)，允许来自复制实例的传入访问。如何进行此项设置取决于您使用的网络配置。您可以使用复制实例 VPC 安全组、复制实例的私有或公有 IP 地址或者 NAT 网关的公有 IP 地址。这些连接构成了您用于数据迁移的网络。

### Note

由于 IP 地址可能会因底层基础设施的变化而发生变化，因此我们建议您使用 VPC CIDR 范围，或者通过与 NAT GW 关联的弹性 IP 路由复制实例的出站流量。有关创建 VPC（包括 CIDR 块）的更多信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的[使用 VPC 和子网](#)。有关弹性 IP 地址的更多信息，请参阅《Amazon Elastic Compute Cloud 用户指南》中的[弹性 IP 地址](#)。

## 数据库迁移的网络配置

您可以在 D AWS atabase Migration Service 中使用几种不同的网络配置。以下是用于数据库迁移的常见网络配置。

### 主题

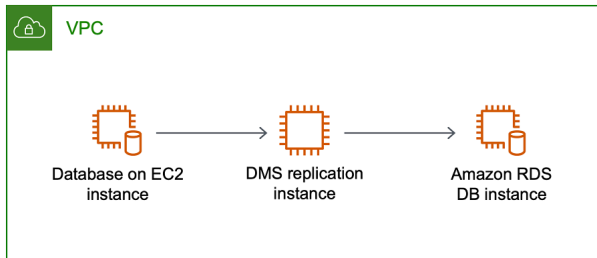
- [一个 VPC 中具有所有数据库迁移组件的配置](#)
- [多 VPC 配置](#)
- [使用共享 VPC 的配置](#)
- [使用 AWS Direct Connect 或 VPN 配置到 VPC 的网络](#)
- [使用 Internet 连接 VPC 的网络配置](#)
- [使用不在 VPC 中的 RDS 数据库实例与 VPC 中的数据库实例进行配置 ClassicLink](#)

如果可行，我们建议您在与目标端点所在的相同区域以及相同 VPC 或子网中创建 DMS 复制实例。

## 一个 VPC 中具有所有数据库迁移组件的配置

对于数据库迁移而言，最简单的网络是源终端节点、复制实例和目标终端节点都位于同一个 VPC 中。当源和目标端点均位于 Amazon RDS 数据库实例或 Amazon EC2 实例上时，此配置非常合适。

下图显示了位于 Amazon EC2 实例上的数据库连接到复制实例并将数据迁移到 Amazon RDS 数据库实例的配置。



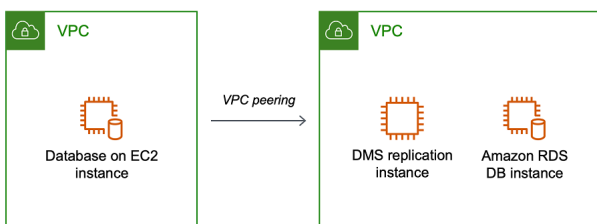
此配置中使用的 VPC 安全组必须允许从复制实例向数据库端口的传入。可以通过多种方式执行此操作。您可以确保复制实例使用的安全组可以传入端点。或者，如果您正在使用复制实例，则可以允许复制实例的 VPC CIDR 范围、NAT GW 弹性 IP 或私有 IP 地址。但是我们不建议您使用复制实例的私有 IP 地址，因为如果复制 IP 地址发生变化，它可能中断您的复制。

## 多 VPC 配置

如果您的源端点和目标端点位于不同 VPC，则可以在其中一个 VPC 中创建复制实例。然后，可以使用 VPC 对等连接将这两个 VPC 链接起来。

VPC 对等连接是两个 VPC 之间的连接，使用各 VPC 的私有 IP 地址进行路由，就像它们位于同一个网络中一样。您可以在自己的 VPC 之间、与其他 AWS 账户中的 VPC 或不同 AWS 区域的 VPC 之间创建 VPC 对等连接。有关 VPC 对等连接的更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 对等连接](#)。

下图显示了一个使用 VPC 对等连接的示例配置。在此处，VPC 中 Amazon EC2 实例上的源数据库通过 VPC 对等连接来连接到 VPC。此 VPC 包含 Amazon RDS 数据库实例上的复制实例和目标数据库。



要实现 VPC 对等连接，请按照《Amazon Virtual Private Cloud VPC 对等连接》文档中的[使用 VPC 对等连接](#)说明进行操作。请确保一个 VPC 的路由表中包含另一个 VPC 的 CIDR 块。例如，如果 VPC A 使用目标 10.0.0.0/16，而 VPC B 使用目标 172.31.0.0，则 VPC A 的路由表应包含 172.31.0.0，VPC B 的路由表必须包含 10.0.0.0/16。有关更多详细信息，请参阅《Amazon Virtual Private Cloud VPC 对等连接》文档中的[更新 VPC 对等连接的路由表](#)。

此配置中使用的 VPC 安全组必须允许从复制实例向数据库端口的传入，还应允许具有对等关系的 VPC 上 CIDR 块的传入。

## 使用共享 VPC 的配置

AWS DMS 将共享给组织中参与客户账户的子网视为同一账户中的普通子网。以下描述了如何 AWS DMS 处理 VPC、子网以及如何使用共享 VPC。

通过创建 `ReplicationSubnetGroup` 对象，您可以将网络配置设置为在自定义子网或 VPC 中运行。创建 `ReplicationSubnetGroup` 时，您可以选择指定账户中特定 VPC 的子网。您指定的子网列表必须包括至少两个位于不同可用区的子网，并且所有子网都必须位于同一 VPC 中。在创建时 `ReplicationSubnetGroup`，客户只能指定子网。AWS DMS 将代表您确定 VPC，因为每个子网仅链接到一个 VPC。

创建 AWS DMS `ReplicationInstance` 或 `AWS DMS ReplicationConfig`，您可以选择指定 `ReplicationSubnetGroup` 或无服务器复制在其中运行的 VPC 安全组。`ReplicationInstance` 如果未指定，则 AWS DMS 选择客户默认值 `ReplicationSubnetGroup`（如果未为默认 VPC 中的所有子网指定，则代表您 AWS DMS 创建）和默认 VPC 安全组。

您可以选择在您指定的可用区中运行迁移，也可以选择在 `ReplicationSubnetGroup` 中的任何可用区中运行迁移。当 AWS DMS 尝试创建复制实例或启动无服务器复制时，它会将您的子网的可用区转换为核心服务账户中的可用区，以确保即使两个账户之间的可用区域映射不相同，我们也能在正确的可用区中启动实例。

如果您使用共享 VPC，则需确保创建映射到您希望从共享 VPC 使用的子网的 `ReplicationSubnetGroup` 对象。创建 `ReplicationInstance` 或 `ReplicationConfig` 时，必须为共享 VPC 指定 `ReplicationSubnetGroup`，并在创建请求中指定您为共享 VPC 创建的 VPC 安全组。

请注意以下有关使用共享 VPC 的注意事项：

- VPC 所有者不能与参与者共享资源，但参与者可以在所有者的子网中创建服务资源。

- VPC 所有者无法访问参与者创建的资源（例如复制实例），因为所有资源都是特定于账户的。但是，只要您在共享 VPC 中创建复制实例，并且复制端点或任务具有正确的权限，它就可以访问 VPC 中的资源，而不管拥有哪个账户。
- 由于资源是特定于账户的，因此其他参与者无法访问其他账户拥有的资源。您无法向其他账户授予任何权限，允许他们使用您的账户访问在共享 VPC 中创建的资源。

## 使用 AWS Direct Connect 或 VPN 配置到 VPC 的网络

远程网络可以使用多种选项连接到 VPC，例如 AWS Direct Connect 或软件或硬件 VPN 连接。这些选项通常将内部网络扩展到 AWS 云来集成现有现场服务，例如监控、身份验证、安全、数据或其他系统。使用此类型的网络扩展，您可以无缝连接到 AWS 托管的资源，例如 VPC。

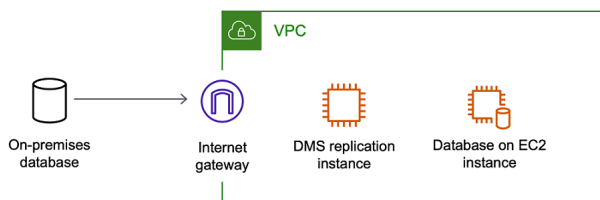
下图显示了源终端节点是公司数据中心内本地数据库的配置。它使用 AWS Direct Connect 或 VPN 连接到 Amazon RDS 数据库实例上包含复制实例和目标数据库的 VPC。



在此配置中，VPC 安全组必须包含一个路由规则，即，将以 VPC CIDR 范围或特定 IP 地址为目标的流量发送到主机。此主机必须能够将流量从 VPC 桥接到本地 VPN 中。在这种情况下，NAT 主机包括自己的安全组设置。这些设置必须允许来自复制实例的 VPC CIDR 范围、私有 IP 地址或安全组的流量进入 NAT 实例。但是我们不建议您使用复制实例的私有 IP 地址，因为如果复制 IP 地址发生变化，它可能中断您的复制。

## 使用 Internet 连接 VPC 的网络配置

如果您不使用 VPN 或 AWS Direct Connect 不连接 AWS 资源，则可以使用互联网迁移数据库。在这种情况下，您可以迁移到 Amazon EC2 实例或 Amazon RDS 数据库实例。此配置涉及到带有 Internet 网关的 VPC 中的公有复制实例，该 VPC 中包含目标终端节点和复制实例。



要将互联网网关添加到 VPC，请参阅《Amazon VPC 用户指南》中的[连接互联网网关](#)。

VPC 路由表必须包含一个路由规则，即，将并非以 VPC 为目标的流量默认发送到互联网网关。在此配置中，与终端节点的连接将显示为来自复制实例的公有 IP 地址，而非私有 IP 地址。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [VPC 路由表](#)。

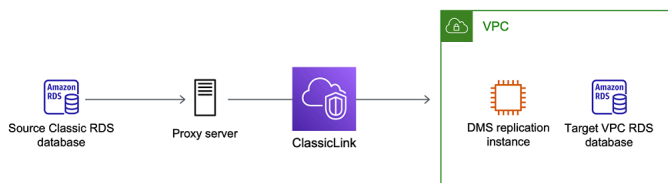
## 使用不在 VPC 中的 RDS 数据库实例与 VPC 中的数据库实例进行配置 ClassicLink

我们将于 2022 年 8 月 15 日停用 EC2-Classic。我们建议您从 EC2-Classic 迁移到 VPC。有关更多信息，请参阅《Amazon EC2 用户指南》中的[从 EC2-Classic 迁移到 VPC](#) 和博客 [EC2-Classic Networking is Retiring – Here's How to Prepare](#)。

要将不在 VPC 中的 Amazon RDS 数据库实例连接到 DMS 复制服务器和 VPC 中的数据库实例 ClassicLink，您可以使用代理服务器。

ClassicLink 使您能够将 EC2-Classic 数据库实例关联到同一 AWS 区域内您账户中的 VPC。在您创建了链接之后，源数据库实例可以与 VPC 中的复制实例通过其私有 IP 地址进行通信。

由于 VPC 中的复制实例无法使用直接访问 EC2-Classic 平台上的源数据库实例 ClassicLink，因此您可以使用代理服务器。代理服务器将源数据库实例连接到包含复制实例和目标数据库实例的 VPC。代理服务器用于连接 ClassicLink 到 VPC。代理服务器上的端口转发允许源数据库实例与 VPC 中的目标数据库实例之间的通信。



## 使用 AWS DMS ClassicLink 和 Migration Service

您可以将不在 VPC 中的 Amazon RDS 数据库实例连接到位于 VPC 中的 AWS DMS 复制服务器和数据库实例。为此，您可以将 Amazon EC2 ClassicLink 与代理服务器一起使用。

以下过程说明如何 ClassicLink 用于此目的。此过程将不在 VPC 中的 Amazon RDS 源数据库实例连接到包含 AWS DMS 复制实例和目标数据库实例的 VPC。

- 在 VPC 中创建 AWS DMS 复制实例。（在 VPC 中创建所有复制实例。）
- 将 VPC 安全组与复制实例和目标数据库实例关联。当两个实例共享一个 VPC 安全组时，默认情况下，它们可以相互通信。
- 在 EC2 Classic 实例上设置代理服务器。

- 在代理服务器和 VPC ClassicLink 之间使用创建连接。
- 为源数据库和目标数据库创建 AWS DMS 端点。
- 创建 AWS DMS 任务。

用于 ClassicLink 将不在 VPC 中的数据库实例上的数据库迁移到 VPC 中数据库实例上的数据库

#### 1. 创建 AWS DMS 复制实例并分配 VPC 安全组：

- a. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。

如果您以 AWS Identity and Access Management (IAM) 用户身份登录，请确保您拥有相应的访问权限 AWS DMS。有关数据库迁移所需权限的更多信息，请参阅 [使用 AWS DMS 所需的 IAM 权限](#)。

- b. 在控制面板页中，选择复制实例。按照[步骤 1：使用 AWS DMS 控制台创建复制实例](#)中的说明操作来创建复制实例。
  - c. 创建 AWS DMS 复制实例后，打开 EC2 服务控制台。从导航窗格中选择网络接口。
  - d. 选择 DMS NetworkInterface，然后从“操作”菜单中选择“更改安全组”。
  - e. 选择要用于复制实例和目标数据库实例的安全组。
- #### 2. 将最后一步中的安全组与目标数据库实例关联：

- a. 打开 Amazon RDS 服务控制台。从导航窗格中选择实例。
- b. 选择目标数据库实例。在实例操作中，选择修改。
- c. 对于安全组参数，请选择在上一步中使用的安全组。
- d. 选择继续，然后选择修改数据库实例。

#### 3. 步骤 3：使用 NGINX 在 EC2 Classic 实例上设置代理服务器。使用您选择的 AMI 启动 EC2 Classic 实例。下面的示例基于 AMI Ubuntu Server 14.04 LTS (HVM)。

在 EC2 Classic 实例上设置代理服务器

- a. 使用以下命令连接到 EC2 Classic 实例并安装 NGINX：

```
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. 使用以下代码编辑 NGINX 守护程序文件 `/etc/init/nginx.conf` :

```
# /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. 在 `/usr/local/nginx/conf/nginx.conf` 中创建 NGINX 配置文件。在配置文件中，添加以下内容：

```
# /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. 从命令行中，使用以下命令启动 NGINX：

```
Prompt> sudo initctl reload-configuration
Prompt> sudo initctl list | grep nginx
Prompt> sudo initctl start nginx
```

4. 在代理服务器和包含目标数据库实例和复制实例的目标 VPC 之间创建 ClassicLink 连接：
  - a. 打开 EC2 控制台，并选择正在运行代理服务器的 EC2 Classic 实例。
  - b. 对于“操作”，选择 ClassicLink，然后选择“链接到 VPC”。
  - c. 选择本过程中前面使用的安全组。
  - d. 选择链接到 VPC。
5. 步骤 5：使用中的步骤创建 AWS DMS 端点。[步骤 2：指定源和目标端点](#)在指定源端点时，请确保将代理的内部 EC2 DNS 主机名作为服务器名称。
6. 使用中的步骤创建 AWS DMS 任务。[步骤 3：创建任务并迁移数据](#)

## 创建复制子组

作为数据库迁移所用网络的一部分，您需要指定计划在虚拟私有云 (VPC) 中使用的子网。此 VPC 必须基于 Amazon VPC 服务。子网是在指定可用区中您的 VPC 内的 IP 地址范围。这些子网可以分布在您的 VPC 所在 AWS 区域的可用区中。

在 AWS DMS 控制台中创建复制实例或实例配置文件时，您可以使用您选择的子网。

您创建复制子网组来定义要使用的子网。您必须指定至少位于两个可用区中的子网。

## 创建复制子网组

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。

如果以 IAM 用户身份登录，请确保具有适当的 AWS DMS 访问权限。有关数据库迁移所需权限的更多信息，请参阅 [使用 AWS DMS 所需的 IAM 权限](#)。

2. 在导航窗格中，选择子网组。
3. 选择 Create subnet group (创建子网组)。
4. 在创建复制子网组页中，指定您的复制子网组信息。下表描述了设置。

选项	操作
名称	为复制子网组输入包含 8 到 16 个可打印 ASCII 字符（不含 /、" 和 @）的名称。对于您所选 AWS 地区的账户，该名称应是唯一的。例如，您可以选择在名称中添加一些情报，例如包括您正在执行的 AWS 区域和任务 <b>DMS-default-VPC</b> 。
描述	输入复制子网组的简短说明。
VPC	选择您要用于数据库迁移的 VPC。请记住，VPC 必须至少有一个位于两个可用区中的子网。
添加子网	选择您要包括在复制子网组中的子网。您必须选择至少位于两个可用区中的子网。

5. 选择 Create subnet group (创建子网组)。

## 使用 DNS 解析域端点

通常，AWS DMS 复制实例使用 Amazon EC2 实例中的域名系统 (DNS) 解析器来解析域终端节点。如果您需要 DNS 解析，可以使用 Amazon Route 53 Resolver。有关使用 Route 53 DNS 解析程序的信息，请参阅 [Route 53 Resolver 入门](#)。

有关如何使用您自己的本地名称服务器通过 Amazon Route 53 Resolver 解析某些端点的信息，请参阅[使用您自己的本地名称服务器](#)。

## 为复制实例设置加密密钥

AWS DMS 对复制实例使用的存储空间和端点连接信息进行加密。为了加密复制实例使用的存储，AWS DMS 使用您的 AWS 账户独有的存储。AWS KMS key 您可以使用 AWS Key Management Service (AWS KMS) 查看和管理此 KMS 密钥。您可以使用您账户 ( aws/dms ) 中的默认 KMS 密钥，也可以自行创建 KMS 密钥。如果您已有 AWS KMS 加密密钥，也可以使用该密钥进行加密。

您可以通过提供 KMS 密钥标识符来加密您的 AWS DMS 资源来指定自己的加密密钥。在您指定自己的加密密钥时，执行数据库迁移所用的用户账户必须具有对该密钥的访问权限。有关创建您自己的加密密钥以及向用户提供对加密密钥访问权限的更多信息，请参阅[AWS KMS 开发人员指南](#)。

如果您未指定 KMS 密钥标识符，则 AWS DMS 将使用您的默认加密密钥。KMS 为您的 AWS 账户创建 AWS DMS 的默认加密密钥。您的 AWS 账户在每个 AWS 区域都有不同的默认加密密钥。

要管理用于加密 AWS DMS 资源的密钥，可以使用。AWS KMS 您可以通过 AWS Management Console 通过 AWS KMS 在导航窗格中搜索 KMS 来查找。

AWS KMS 将安全、高度可用的硬件和软件相结合，提供可扩展到云端的密钥管理系统。使用 AWS KMS，您可以创建加密密钥并定义控制如何使用这些密钥的策略。AWS KMS 支持 AWS CloudTrail，因此您可以审核密钥使用情况，以验证密钥的使用是否正确。您的 AWS KMS 密钥可以与 AWS DMS 和其他支持的 AWS 服务结合使用。支持的 AWS 服务包括 Amazon RDS、Amazon S3、Amazon Elastic Block Store ( Amazon EBS ) 和 Amazon Redshift。

使用特定加密密钥创建 AWS DMS 资源后，您无法更改这些资源的加密密钥。在创建 AWS DMS 资源之前，请务必确定您的加密密钥要求。

## 创建复制实例

迁移数据的第一个任务是创建复制实例。此复制实例具有足够的存储和处理能力来执行您分配的任务，并将数据从源数据库迁移至目标数据库。此实例的所需大小是变化的，具体取决于需迁移的数据量和需要实例执行的任务数。有关复制实例的更多信息，请参阅[使用 AWS DMS 复制实例](#)。

使用 AWS 控制台创建复制实例

1. 在 AWS DMS 控制台的导航窗格中选择“复制实例”，然后选择“创建复制实例”。
2. 在创建复制实例页中，指定您的复制实例信息。下表描述了可以配置的设置。

选项	操作
名称	为复制实例输入包含 8 到 16 个可打印 ASCII 字符 ( 不含 /、" 和 @ ) 的名称。该名称在所选择 AWS 区域中对于您的账户应唯一。您可以选择在名称中添加一些情报，例如包括您正在执行的 AWS 区域和任务 <b>west2-mysql2mysql-instance1</b> 。
描述性 Amazon 资源名称 ( ARN ) – 可选	用于覆盖默认 DMS ARN 的友好名称。您不能在创建后对其进行修改。
描述	输入复制实例的简要描述。
实例类	选择带迁移所需的配置的实例类。请记住，实例必须具有足够的存储、网络和处理能力才能成功完成迁移。有关如何决定最适合迁移的实例类的更多信息，请参阅 <a href="#">使用 AWS DMS 复制实例</a> 。
引擎版本	在 AWS DMS 控制台中，您可以根据需要选择任何支持的引擎版本。从中 AWS CLI，除非您在中指定不同的引擎版本，否则 AWS DMS 复制实例将运行最新的非测试版复制引擎。AWS CLI
高可用性	可使用此可选参数在另一个可用区中创建复制实例的备用副本以支持故障转移。如果您打算使用更改数据捕获 ( CDC ) 或持续复制，您应启用该选项。

选项	操作
分配的存储空间 (GiB)	<p>存储空间主要由日志文件和缓存的事务消耗。对于缓存事务，仅在需要将缓存的事务写入磁盘时消耗存储空间。因此，AWS DMS 不会使用大量存储空间。一些例外情况包括：</p> <ul style="list-style-type: none"> <li>• 产生大量事务负载的超大型表。加载大型表可能需要花费一些时间，因此，在大型表加载期间，很有可能将缓存的事务写入磁盘。</li> <li>• 配置为在加载缓存的事务之前暂停的任务。在此情况下，将缓存所有事务，直到为所有表完成完全加载。在使用此配置时，缓存的事务可能会占用大量存储空间。</li> <li>• 使用将加载到 Amazon Redshift 中的表配置的任务。但在目标为 Amazon Aurora 的情况下，此配置不会成为问题。</li> </ul> <p>在大多数情况下，默认分配的存储空间已足够。不过，关注与存储相关的指标始终是不错的做法。如果您发现使用的存储空间超过了默认分配值，请务必扩展存储。</p>
网络类型	<p>DMS 支持 IPv4 寻址协议网络类型，并且在双栈模式下同时支持 IPv4 和 IPv6 寻址协议网络类型。如果您有必须通过 IPv6 寻址协议网络类型与复制实例通信的资源，请使用双堆栈模式。有关双堆栈模式的限制信息，请参阅 <a href="#">Amazon Relational Database Service</a> 用户指南中的 <a href="#">双堆栈网络数据库实例的限制</a>。</p>
VPC	<p>选择要使用的 VPC。如果源或目标数据库位于 VPC 中，请选择该 VPC。如果您的源数据库和目标数据库位于不同的 VPC 中，请确保它们位于公有子网中且可公开访问。然后选择将放置复制实例的 VPC。复制实例必须能够访问源 VPC 中的数据。如果您的源数据库和目标数据库都不在 VPC 中，请选择将放置该复制实例的 VPC。</p>

选项	操作
复制子网组	在选定 VPC 中选择要在其中创建复制实例的复制子网组。如果源数据库位于 VPC 中，请选择包含源数据库的子网组作为复制实例的位置。有关复制子网组的更多信息，请参阅 <a href="#">创建复制子组</a> 。
公开访问	如果您希望能够从 Internet 访问复制实例，请选择该选项。默认设置为可公开访问，一旦选择此选项，您就无法在创建复制实例后对其进行修改。

3. 选择 Advanced (高级) 选项卡以设置网络和加密设置值 ( 如果需要 )。下表描述了设置。

选项	操作
可用区	选择源数据库所在的可用区。
VPC 安全组	在 VPC 中创建复制实例。如果源数据库位于 VPC 中，请选择一个 VPC 安全组，该安全组提供对数据库所在的数据库实例的访问权限。
KMS 密钥	选择要用于加密复制存储和连接信息的加密密钥。如果您选择 ( 默认 ) aws/dms，则使用与您的账户和 AWS 地区关联的默认 AWS Key Management Service (AWS KMS) 密钥。将显示说明和您的账号以及密钥的 ARN。有关使用加密密钥的更多信息，请参阅 <a href="#">设置加密密钥并指定 AWS KMS 权限</a> 。

4. 指定维护设置。下表描述了设置。有关维护设置的更多信息，请参阅 [使用 AWS DMS 维护时段](#)。

选项	操作
自动升级版本	<p>AWS DMS 不区分主要版本和次要版本。例如，从版本 3.4.x 升级到 3.5.x 不被视为重大升级，因此所有更改都应向后兼容。</p> <p>启用自动升级版本后，如果复制实例的版本已过时，DMS 会在维护时段内自动升级该版本。</p>

选项	操作
	<p>启用后 <code>AutoMinorVersionUpgrade</code>，DMS 将在您创建复制实例时使用当前的默认引擎版本。例如，如果您将引擎版本设置为低于当前默认版本的版本号，DMS 将使用默认版本。</p> <p>如果在创建复制实例时 <code>AutoMinorVersionUpgrade</code> 未启用，则 DMS 将使用引擎版本参数指定的引擎版本。</p>
维护窗口	<p>选择每周可以进行系统维护的时间范围，采用通用协调时间 (UTC)。</p> <p>默认：从每个 AWS 区域的 8 小时时间段中随机选择一个 30 分钟的窗口，发生在一周中的随机某一天。</p>

## 5. 选择创建复制实例。

## 修改复制实例

您可以修改复制实例设置以更改实例类或增加存储 (举例)。

在修改复制实例时，您可以立即应用更改。要立即应用更改，请选中 AWS Management Console 中的立即应用更改选项。或者在调用时使用 `--apply-immediately` 参数 AWS CLI，或者在使用 DMS API `true` 时将 `ApplyImmediately` 参数设置为。

如果您没有选择立即应用更改，更改将被放置在等待修改队列中。队列中的所有等待更改都将在下一维护时段应用。

### Note

如果您选择立即应用更改，也将同时应用等待修改队列中的所有更改。如果任何待处理修改需要停机，选择立即应用更改可能导致意外停机。

### 使用 AWS 控制台修改复制实例

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。



2. 在导航窗格中，选择复制实例。
3. 选择要修改的复制实例。下表描述了可进行的修改。

选项	操作
名称	您可以更改复制实例的名称。为复制实例输入包含 8 到 16 个可打印 ASCII 字符（不含 /、" 和 @）的名称。该名称在所选择 AWS 区域中对于您的账户应唯一。您可以选择在名称中添加一些情报，例如包括您正在执行的 AWS 区域和任务 <code>west2-mysql2mysql-instance1</code> 。
描述	修改或输入复制实例的简要描述。
实例类	<p>您可以更改实例类。选择带迁移所需的配置的实例类。更改实例类会导致复制实例重启。将在下一维护时段内重启；如果选择立即应用更改选项，则会立即进行重启。</p> <p>有关如何决定最适合迁移的实例类的更多信息，请参阅 <a href="#">使用 AWS DMS 复制实例</a>。</p>
引擎版本	您可以升级复制实例使用的引擎版本。升级复制引擎版本将导致在升级时关闭复制实例。
多可用区	您可以更改该选项以在另一个可用区中创建您的复制实例的备用副本以支持故障转移，也可以删除该选项。如果您打算使用更改数据捕获 (CDC) 持续复制，则应该启用该选项。
分配的存储空间 (GiB)	<p>存储空间主要由日志文件和缓存的事务消耗。对于缓存事务，仅在需要将缓存的事务写入磁盘时消耗存储空间。因此，AWS DMS 不会使用大量存储空间。一些例外情况包括：</p> <ul style="list-style-type: none"> <li>• 产生大量事务负载的超大型表。加载大型表可能需要花费一些时间，因此，在大型表加载期间，很有可能将缓存的事务写入磁盘。</li> </ul>

选项	操作
	<ul style="list-style-type: none"> <li>配置为在加载缓存的事务之前暂停的任务。在此情况下，将缓存所有事务，直到为所有表完成完全加载。在使用此配置时，缓存的事务可能会占用大量存储空间。</li> <li>使用将加载到 Amazon Redshift 中的表配置的任务。但在目标为 Amazon Aurora 的情况下，此配置不会成为问题。</li> </ul> <p>在大多数情况下，默认分配的存储空间已足够。不过，关注与存储相关的指标并在您发现使用的存储空间大于默认分配值时纵向扩展始终是个不错的做法。</p>
网络类型	<p>DMS 支持 IPv4 寻址协议网络类型，并且在双栈模式下同时支持 IPv4 和 IPv6 寻址协议网络类型。如果您有必须通过 IPv6 寻址协议网络类型与复制实例通信的资源，请选择双堆栈模式。有关双堆栈模式的限制信息，请参阅 <a href="#">Amazon Relational Database Service 用户指南中的双堆栈网络数据库实例的限制</a>。</p>
VPC 安全组	<p>在 VPC 中创建复制实例。如果源数据库位于 VPC 中，请选择一个 VPC 安全组，该安全组提供对数据库所在的数据库实例的访问权限。</p>
自动升级版本	<p>AWS DMS 不区分主要版本和次要版本。例如，从版本 3.4.x 升级到 3.5.x 不被视为重大升级，因此所有更改都应向后兼容。启用自动升级版本后，如果复制实例的版本已过时，DMS 会在维护时段内自动升级该版本。</p> <p>启用自动升级版本后，DMS 将在您创建复制实例时使用当前的默认引擎版本。例如，如果您将引擎版本设置为低于当前默认版本的版本号，DMS 将使用默认版本。</p> <p>如果在创建复制实例时未启用自动升级版本，DMS 将使用引擎版本参数指定的引擎版本。</p>

选项	操作
维护窗口	<p>选择每周可以进行系统维护的时间范围，采用通用协调时间 (UTC)。</p> <p>默认：从每个 AWS 区域的 8 小时时间段中随机选择一个 30 分钟的窗口，发生在一周中的随机某一天。</p>
立即应用更改	<p>选择该选项以立即应用所做的任何修改。根据您的选择的设置，选择该选项可能会导致立即重启复制实例。</p> <p>如果您选择在 AWS DMS 应用更改时测试连接，则会看到一条错误消息。AWS DMS 将更改应用于您的复制实例后，再次选择测试连接。</p>
在下一个计划的维护时段内应用更改	<p>如果您希望 DMS 等到下一个计划的维护时段才应用您的更改，请选择此选项。</p>

## 重启复制实例

您可以重启 AWS DMS 复制实例以重新启动复制引擎。重启会导致复制实例的短暂中断，在此期间此实例状态会设置为重启中。如果 AWS DMS 实例配置为多可用区，则可以通过故障转移进行重启。重新启动完成后会创建一个 AWS DMS 事件。

如果您的 AWS DMS 实例是多可用区部署，则可以在重启时强制按计划从一个 AWS 可用区转移到另一个可用区。当您强制对 AWS DMS 实例进行计划内故障转移时，会先 AWS DMS 关闭当前实例上的活动连接，然后自动切换到另一个可用区域中的备用实例。使用计划内故障转移重新启动可以帮助您模拟 AWS DMS 实例的计划内故障转移事件，例如扩展复制实例类别时。

### Note

当重新启动强制执行从一个可用区到另一个可用区的失效转移后，可用区更改可能要过几分钟才会反映出来。这种延迟出现在对 AWS Management Console AWS CLI 和 AWS DMS API 的调用中。

如果重启时正在复制实例上运行迁移任务，则不会发生数据丢失，但任务会停止，任务状态将更改为错误状态。

如果迁移任务中的表处于批量加载（完全加载阶段）并且尚未启动，它们就会进入错误状态。但当时已完成的表仍处于完成状态。在完全加载阶段重启时，我们建议您执行以下任一步骤。

- 从任务中移除处于完成状态的表，然后对剩余表重启该任务。
- 为处于错误状态和待处理状态的表创建一个新任务。

如果迁移任务中的表位于持续复制阶段，则任务会在重启完成后立即继续。

如果 AWS DMS 复制实例的状态不是“可用”状态，则无法重启该实例。您的 AWS DMS 实例可能由于多种原因而不可用，例如之前请求的修改或维护窗口操作。重启 AWS DMS 复制实例所需的时间通常很短（不到 5 分钟）。

## 使用控制台重启复制实例 AWS

要重启复制实例，请使用 AWS 控制台。

使用 AWS 控制台重启复制实例

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择复制实例。
3. 选择要重启的复制实例。
4. 选择重启。此时显示重启复制实例对话框。
5. 如果复制实例已配置为多可用区部署，并且需要失效转移到另一个 AWS 可用区，请选中通过计划内失效转移重启？对应的复选框。
6. 选择重启。

## 使用 CLI 重启复制实例

要重启复制实例，请使用带有以下参数的 AWS CLI [reboot-replication-instance](#) 命令：

- `--replication-instance-arn`

Example 示例：简单重启

以下 AWS CLI 示例重启复制实例。

```
aws dms reboot-replication-instance \
```

```
--replication-instance-arn arn of my rep instance
```

Example 示例：简单的通过故障转移重启

以下 AWS CLI 示例通过故障转移重启复制实例。

```
aws dms reboot-replication-instance \  
--replication-instance-arn arn of my rep instance \  
--force-planned-failover
```

## 使用 API 重启复制实例

要重启复制实例，请使用带有以下参数的 AWS DMS API [RebootReplicationInstance](#) 操作：

- ReplicationInstanceArn = *arn of my rep instance*

Example 示例：简单重启

以下代码示例重启复制实例。

```
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 示例：简单的通过故障转移重启

以下代码示例重新启动复制实例并故障转移到另一个 AWS 可用区。

```
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
```

## 删除复制实例

使用完 AWS DMS 复制实例后，您可以将其删除。如果您有迁移任务在使用该复制实例，则必须先停止并删除任务，然后才能删除该复制实例。

如果您关闭 AWS 账户，则与您的账户关联的所有 AWS DMS 资源和配置将在两天后删除。这些资源包括所有复制实例、源和目标终端节点配置、复制任务和 SSL 证书。如果两天后您决定 AWS DMS 再次使用，则可以重新创建所需的资源。

如果您的复制实例符合所有删除条件，并且在较长一段时间内保持 DELETING 状态，请联系支持人员排查问题。

### 使用 AWS 控制台删除复制实例

要删除复制实例，请使用 AWS 控制台。

使用 AWS 控制台删除复制实例

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择复制实例。
3. 选择要删除的复制实例。
4. 选择删除。
5. 在对话框中，选择删除。

### 使用 CLI 删除复制实例

要删除复制实例，请使用带有以下参数的 AWS CLI [delete-replication-instance](#) 命令：

- `--replication-instance-arn`

## Example 删除操作示例

以下 AWS CLI 示例删除了一个复制实例。

```
aws dms delete-replication-instance \  
--replication-instance-arn arn of my rep instance
```

## 使用 API 删除复制实例

要删除复制实例，请使用带有以下参数的 AWS DMS API [DeleteReplicationInstance](#) 操作：

- ReplicationInstanceArn = *arn of my rep instance*

## Example 删除操作示例

以下代码示例将删除复制实例。

```
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
```

## 使用 AWS DMS 维护时段

每个 AWS DMS 复制实例都有一个每周维护窗口，在此期间会应用任何可用的系统更改。您可以通过维护时段控制何时进行修改和软件修补。

如果 AWS DMS 确定需要在给定的一周内进行维护，则维护将在您在创建复制实例时选择的 30 分钟维护时段内进行。AWS DMS 在 30 分钟的维护窗口内完成大部分维护。但是，对于较大的更改可能需要更长时间。

## 维护对现有迁移任务的影响

在实例上运行 AWS DMS 迁移任务时，应用补丁时会发生以下事件：

- 如果迁移任务中的表处于复制持续更改阶段 ( CDC )，则 AWS DMS 会将该任务暂停一段时间，并在应用修补程序后恢复运行。在应用修补程序后，将从中断的位置继续进行迁移。
- 如果 AWS DMS 在迁移现有数据或迁移现有数据并复制正在进行的更改任务中迁移表，则在应用补丁期间，DMS 会停止并重新启动所有处于满负荷阶段的表的迁移。在应用修补程序时，DMS 还会停止所有处于 CDC 阶段的表，并随后恢复。

## 使用维护时段设置

您可以使用 AWS Management Console、或 AWS DMS API 更改维护时段的时间范围。AWS CLI

### 使用控制台更改维护时段设置

您可以使用 AWS Management Console 更改维护时段时间范围。

### 使用控制台更改首选的维护时段

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，选择复制实例。
3. 选择要修改的复制实例，然后选择修改。
4. 展开维护选项卡，然后为您的维护时段选择一个日期和时间。
5. 选择立即应用更改。
6. 选择修改。

### 使用 CLI 更改维护时段设置

要调整首选维护时段，请使用带有以下参数的 AWS CLI [modify-replication-instance](#) 命令。

- `--replication-instance-identifier`
- `--preferred-maintenance-window`

### Example

以下 AWS CLI 示例将维护时段设置为星期二凌晨 4:00 — 4:30。(UTC)。

```
aws dms modify-replication-instance \  
--replication-instance-identifier myrepliance \  

```



```
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

## 使用 API 更改维护时段设置

要调整首选维护时段，请使用带有以下参数的 AWS DMS API [ModifyReplicationInstance](#) 操作。

- `ReplicationInstanceIdentifier` = *myrepinstance*
- `PreferredMaintenanceWindow` = *Tue:04:00-Tue:04:30*

## Example

以下代码示例将维护时段设置为周二的凌晨 4:00–4:30 (UTC)。

```
https://dms.us-west-2.amazonaws.com/  
?Action=ModifyReplicationInstance  
&DBInstanceIdentifier=myrepinstance  
&PreferredMaintenanceWindow=Tue:04:00-Tue:04:30  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-09-01  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20140425/us-east-1/dms/aws4_request  
&X-Amz-Date=20140425T192732Z  
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
&X-Amz-Signature=1dc9dd716f4855e9bdf188c70f1cf9f6251b070b68b81103b59ec70c3e7854b3
```

# 使用 AWS DMS 端点

端点提供有关您的数据存储的连接、数据存储类型和位置信息。AWS Database Migration Service 使用此信息连接到数据存储，并将数据从源端点迁移到目标端点。您可以使用端点设置为端点指定其他连接属性。这些设置可以控制日志记录、文件大小和其他参数；有关端点设置的更多信息，请参阅您的数据存储的文档部分。

接下来，您可以了解有关终端节点的更多详细信息。

## 主题

- [创建源和目标终端节点](#)
- [数据迁移的源](#)
- [数据迁移的目标](#)
- [将 VPC 端点配置为 AWS DMS 源端点和目标端点](#)
- [AWS DMS 支持的 DDL 语句](#)

## 创建源和目标终端节点

您可以在创建复制实例时创建源和目标终端节点，也可以在创建复制实例后创建这些终端节点。源和目标数据存储可能位于 Amazon Elastic Compute Cloud (Amazon EC2) 实例和 Amazon Relational Database Service (Amazon RDS) 数据库实例上，也可能是本地数据库。（请注意，您的其中一个端点必须位于 AWS 服务上。您不能使用 AWS DMS 从本地数据库迁移到另一个本地数据库。）

以下过程假定您已选择 AWS DMS 控制台向导。请注意，您也可以从 AWS 控制台的导航窗格中选择端点，然后选择创建端点以执行该步骤。在使用控制台向导时，您将在同一页面上创建源和目标终端节点。如果未使用控制台向导，您可以单独创建每个终端节点。

### 使用 AWS 控制台指定源或目标数据库端点

1. 在连接源和目标数据库终端节点页中，指定源或目标数据库的连接信息。下表描述了设置。

对于该选项	请执行该操作
端点类型	选择此终端节点是源还是目标终端节点。

对于该选项	请执行该操作
Select RDS DB Instance (选择 RDS 数据库实例)	如果终端节点是 Amazon RDS DB 实例，请选择此选项。
终端节点标识符	键入要用于标识终端节点的名称。您可能需要在名称中包含终端节点的类型，例如 <b>oracle-source</b> 或 <b>PostgreSQL-target</b> 。对于所有复制实例来说，名称必须是唯一的。
源引擎和目标引擎	选择作为终端节点的数据库引擎的类型。
访问端点数据库	选择您要用于指定端点数据库凭证的选项： <ul style="list-style-type: none"> <li>• <a href="#">选择 AWS Secrets Manager</a> – 使用 AWS Secrets Manager 中定义的密钥，保密地提供您的凭证，如下所示。有关创建这些密钥以及允许 AWS DMS 访问这些密钥的密钥访问角色的更多信息，请参阅<a href="#">使用密钥访问 AWS Database Migration Service 端点</a>。</li> <li>• <a href="#">手动提供访问信息</a> – 使用您直接输入的明文凭证，如下所示。</li> </ul>
选择 AWS Secrets Manager	设置以下密钥凭证。
密钥 ID	键入您在 AWS Secrets Manager 中创建的密钥的完整 Amazon 资源名称 (ARN)、部分 ARN 或友好名称，用于访问端点数据库。
IAM 角色	键入您在 IAM 中创建的密钥访问角色的 ARN，以向 AWS DMS 提供访问权限来代表您访问由密钥 ID 标识的密钥。有关创建密钥访问角色的更多信息，请参阅 <a href="#">使用密钥访问 AWS Database Migration Service 端点</a> 。
Oracle Automatic Storage Management (ASM) 的密钥 ID	(仅适用于使用 Oracle ASM 的 Oracle 源端点) 键入您在 AWS Secrets Manager 中创建的密钥的完整 Amazon 资源名称 (ARN)、部分 ARN 或友好名称，以便访问 Oracle ASM。创建此密钥通常是为了在与密钥 ID 所标识密钥相同的服务器上访问 Oracle ASM。

对于该选项	请执行该操作
适用于 Oracle ASM 的 IAM 角色	( 仅适用于使用 Oracle ASM 的 Oracle 源端点 ) 键入您在 IAM 中创建的密钥访问角色的 ARN，以向 AWS DMS 提供访问权限来代表您访问由 Oracle Automatic Storage Management (ASM) 的密钥 ID 标识的密钥。
手动提供访问信息	设置以下明文凭证。
服务器名称	键入服务器名称。对于本地数据库，这可以是 IP 地址或公有主机名。对于 Amazon RDS 数据库实例，这可以是数据库实例的终端节点 ( 也称为 DNS 名称 )，例如 <code>mysqlsrvinst.abcd12345678.us-west-2.rds.amazonaws.com</code> 。
端口	键入数据库使用的端口。
安全套接字层 (SSL) 模式	如果您需要为此终端节点启用连接加密，请选择 SSL 模式。根据您的模式，您可能需要提供证书和服务证书信息。
用户名	使用允许数据迁移所需的权限键入用户名。有关所需权限的信息，请参阅本用户指南中的源和目标数据库引擎的安全性部分。
密码	键入具有所需权限的账户的密码。AWS DMS 源和目标端点的密码有字符限制，具体取决于数据库引擎。有关更多信息，请参阅下表。
数据库名称	对于某些数据库引擎，是您要用作端点数据库的数据库的名称。

下表列出了所列数据库引擎的端点密码和密钥管理器密钥中不支持的字符。如果您想在端点密码中使用逗号 (,)，请使用 AWS DMS 中提供的 Secrets Manager 支持来验证对您的 AWS DMS 实例的访问权限。有关更多信息，请参见 [使用密钥访问 AWS Database Migration Service 端点](#)。

对于此数据库引擎	端点密码和密钥管理器密钥中不支持以下字符
全部	{ }
Microsoft Azure，仅作为源	;
Microsoft SQL Server	, ;
MySQL 兼容，包括 MySQL、MariaDB 和 Amazon Aurora MySQL	;
Oracle	,
PostgreSQL、Amazon Aurora PostgreSQL 兼容版和 Amazon Aurora Serverless，仅作为 Amazon Aurora PostgreSQL 兼容版的目标	; + %
Amazon Redshift，仅作为目标	, ;

2. 如果需要，请选择端点设置和 AWS KMS key。您可以选择运行测试以测试终端节点连接。下表描述了设置。

对于该选项	请执行该操作
端点设置	<p>在此处选择任何其他连接参数。有关端点设置的更多信息，请参阅源引擎或目标引擎（在步骤 1 中指定）的文档部分。</p> <p>对于使用 Oracle ASM 的 Oracle 源端点，如果您在步骤 1 中选择手动提供访问信息，则可能还需要键入端点设置以指定 Oracle ASM 用户身份凭证。有关这些 Oracle ASM 端点设置的更多信息，请参阅<a href="#">在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器</a>。</p>
AWS KMS key	选择要用于加密复制存储和连接信息的加密密钥。如果您选择（默认值）aws/dms，则会使用与您的账户和

对于该选项	请执行该操作
	AWS 区域关联的默认 AWS Key Management Service (AWS KMS) 密钥。有关使用加密密钥的更多信息，请参阅 <a href="#">设置加密密钥并指定 AWS KMS 权限</a> 。
测试终端节点连接 ( 可选 )	添加 VPC 和复制实例名称。要测试连接，请选择 Run test (运行测试)。

## 数据迁移的源

AWS Database Migration Service (AWS DMS) 可以将很多最常用的数据引擎作为数据复制源。数据库源可以是在 Amazon EC2 实例上运行的自我管理引擎，也可以是本地数据库。或者，它也可以是 AWS 服务 ( 如 Amazon RDS 或 Amazon S3 ) 上的数据来源。

有关有效来源的完整列表，请参阅 [AWS DMS 的来源](#)。

### 主题

- [将 Oracle 数据库作为 AWS DMS 的源](#)
- [使用微软 SQL Server 数据库作为来源 AWS DMS](#)
- [将 Microsoft Azure SQL 数据库作为 AWS DMS 源](#)
- [使用 Microsoft Azure SQL 托管实例作为 AWS DMS 的来源](#)
- [将 Microsoft Azure Database for PostgreSQL 灵活服务器作为 AWS DMS 的来源](#)
- [将 Microsoft Azure Database for MySQL 灵活服务器作为 AWS DMS 的来源](#)
- [使用 OCI MySQL Heatwave 作为 AWS DMS 的来源](#)
- [将 Google Cloud for MySQL 作为 AWS DMS 的来源](#)
- [将 Google Cloud for PostgreSQL 作为 AWS DMS 的来源](#)
- [将 PostgreSQL 数据库作为 AWS DMS 源](#)
- [将与 MySQL 兼容的数据库作为 AWS DMS 源](#)
- [将 SAP ASE 数据库作为 AWS DMS 源](#)
- [将 MongoDB 作为 AWS DMS 源](#)
- [使用亚马逊 DocumentDB \( 兼容 MongoDB \) 作为来源 AWS DMS](#)
- [使用亚马逊 S3 作为来源 AWS DMS](#)

- [使用适用于 Linux、Unix、Windows 的 IBM Db2 和亚马逊 RDS 数据库 \(Db2 LUW\) 作为来源 AWS DMS](#)
- [将 IBM Db2 for z/OS 数据库用作 AWS DMS 的源](#)

## 将 Oracle 数据库作为 AWS DMS 的源

您可以使用迁移一个或多个 Oracle 数据库中的数据 AWS DMS。使用 Oracle 数据库作为源，您可以将数据迁移到 AWS DMS 支持的任何目标。

AWS DMS 支持以下 Oracle 数据库版本：

- Oracle Enterprise Edition
- Oracle Standard Edition
- Oracle Express 版本
- Oracle Personal Edition

有关 AWS DMS 支持作为源的 Oracle 数据库版本的信息，请参见[的来源 AWS DMS](#)。

可以使用安全套接字层 (SSL) 加密 Oracle 终端节点与您复制实例之间的连接。有关将 SSL 用于 Oracle 终端节点的更多信息，请参见[Oracle 终端节点的 SSL 支持](#)。

AWS DMS 支持使用 Oracle 透明数据加密 (TDE) 对源数据库中的静态数据进行加密。有关将 Oracle TDE 与 Oracle 源终端节点结合使用的更多信息，请参见[使用 Oracle 作为数据源的支持的加密方法 AWS DMS](#)。

AWS 支持在 Oracle 终端节点（以及所有其他端点类型）中使用 TLS 版本 1.2 及更高版本，并建议使用 TLS 版本 1.3 或更高版本。

按照以下步骤将 Oracle 数据库配置为 AWS DMS 源端点：

1. 创建具有相应权限的 Oracle 用户 AWS DMS 来访问您的 Oracle 源数据库。
2. 创建符合所选 Oracle 数据库配置的 Oracle 源端点。要创建 full-load-only 任务，无需进一步配置。
3. 要创建处理变更数据捕获的任务（仅限 CDC 或满载和 CDC 任务），请选择 Oracle LogMiner 或 AWS DMS Binary Reader 来捕获数据更改。选择 LogMiner 或 Binary Reader 决定了后面的一些权限和配置选项。有关 LogMiner 和二进制阅读器的比较，请参阅以下部分。

**Note**

有关完全加载任务、仅 CDC 的任务以及完全加载和 CDC 任务的更多信息，请参阅[创建任务](#)

有关使用 Oracle 源数据库和的更多详细信息 AWS DMS，请参阅以下各节。

**主题**

- [在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器](#)
- [配置自我管理或 AWS 托管的 Oracle 源数据库的工作流 AWS DMS 配置 Oracle 源数据库](#)
- [使用自行管理的 Oracle 数据库作为数据源 AWS DMS](#)
- [使用 AWS 由托管的 Oracle 数据库作为数据源 AWS DMS](#)
- [使用 Oracle 作为来源的限制 AWS DMS](#)
- [Oracle 终端节点的 SSL 支持](#)
- [使用 Oracle 作为数据源的支持的加密方法 AWS DMS](#)
- [使用 Oracle 作为源时支持的压缩方法 AWS DMS](#)
- [使用 Oracle 作为源来复制嵌套表 AWS DMS](#)
- [使用 Oracle 作为源时在 Oracle ASM 上存储 REDO AWS DMS](#)
- [使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)
- [Oracle 的源数据类型](#)

**在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器**

在中 AWS DMS，有两种方法可以在将 Oracle 作为源进行更改数据捕获 (CDC) 时读取重做日志：Oracle LogMiner 和 AWS DMS Binary Reader。LogMiner 是一个 Oracle API，用于读取在线重做日志和存档的重做日志文件。Binary Reader 是一种直接读取和解析原始重做日志文件的 AWS DMS 方法。这些方法具有以下功能。

功能	LogMiner	Binary Reader
易于配置	是	否
减少对源系统 I/O 和 CPU 的影响	否	是
CDC 性能更好	否	是



功能	LogMiner	Binary Reader
支持 Oracle 表集群	是	不支持
支持所有类型的 Oracle 混合列式压缩 ( HCC )	是	部分支持  Binary Reader 不支持在执行 CDC 的任务中使用 QUERY LOW。完全支持所有其他 HCC 类型。
仅在 Oracle 12c 中支持 LOB 列	否 ( Oracle 12c LogMiner 中不提供 LOB Support。 )	是
支持仅影响 LOB 列的 UPDATE 语句	否	是
支持 Oracle 透明数据加密 ( TDE )	部分支持  使用 Oracle 时 LogMiner , AWS DMS 不支持 Amazon RDS for Oracle 的列级别的 TDE 加密。	部分支持  Binary Reader 仅支持自我管理 Oracle 数据库的 TDE。
支持所有 Oracle 压缩方法	是	不支持
支持 XA 事务	否	是
RAC	是  由于性能原因和一些内部 DMS 限制，不建议这样做。	是  强烈建议

**Note**

默认情况下，AWS DMS 使用 Oracle LogMiner 进行 (CDC)。

AWS DMS 在使用 Oracle 源数据库时支持透明数据加密 (TDE) 方法。如果您指定的 TDE 凭据不正确，则 AWS DMS 迁移任务不会失败，这可能会影响加密表的持续复制。有关指定 TDE 凭证的更多信息，请参阅[使用 Oracle 作为数据源的支持的加密方法 AWS DMS](#)。

LogMiner 与一起使用的主要优点 AWS DMS 包括以下几点：

- LogMiner 支持大多数 Oracle 选项，例如加密选项和压缩选项。Binary Reader 不支持所有 Oracle 选项，特别是压缩选项和大多数加密选项。
- LogMiner 提供了更简单的配置，尤其是与 Binary Reader 直接访问设置或使用 Oracle 自动存储管理 (ASM) 管理重做日志时。
- LogMiner 支持表群集供使用 AWS DMS。Binary Reader 则不支持。

使用二进制阅读器的主要优点 AWS DMS 包括以下几点：

- 对于更改量大的迁移，LogMiner 可能会对托管 Oracle 源数据库的计算机产生一些 I/O 或 CPU 影响。Binary Reader 对 I/O 或 CPU 造成影响的几率更小，因为日志是直接挖掘的，而不是进行多个数据库查询。
- 对于变更量大的迁移，与使用 Oracle LogMiner 相比，使用 Binary Reader 时 CDC 的性能通常要好得多。
- 在 Oracle 版本 12c 中，二进制阅读器支持 LOB 的 CDC。LogMiner 不是。

通常，除非出现以下情况之一，否则使用 Oracle 迁移您的 Oracle 数据库：LogMiner

- 需要在源 Oracle 数据库上运行多个迁移任务。
- 源 Oracle 数据库上的更改量或重做日志量很大，或者您有更改并且还在使用 Oracle ASM。

**Note**

如果您在使用 Oracle LogMiner 和 AWS DMS Binary Reader 之间切换，请务必重新启动 CDC 任务。

## Oracle 源数据库上的 CDC 的配置

要将 Oracle 源端点连接到数据库以执行更改数据捕获 (CDC) 任务，您可能需要指定额外的连接属性。对于完全加载和 CDC 任务，或者仅 CDC 的任务，都可能如此。您指定的额外连接属性取决于您用来访问重做日志的方法：Oracle LogMiner 或 AWS DMS Binary Reader。

您在创建源端点时指定了额外的连接属性。如果您具有多个连接属性设置，请用分号将它们彼此分开，并且不需要额外的空格（例如，`oneSetting;thenAnother`）。

AWS DMS LogMiner 默认使用。您无需指定额外连接属性即可使用它。

要使用 Binary Reader 访问重做日志，请添加以下额外连接属性。

```
useLogMinerReader=N;useBfile=Y;
```

为额外连接属性使用以下格式来访问将 ASM 与 Binary Reader 一起使用的服务器。

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_number
+ASM;
```

如下所示将源终端节点 Password 请求参数设置为 Oracle 用户密码和 ASM 密码，以逗号分隔。

```
oracle_user_password,asm_user_password
```

如果 Oracle 源使用 ASM，您可以使用 Binary Reader 中的高性能选项进行大规模事务处理。这些选项包括额外的连接属性，用于指定并行线程数 (`parallelASMReadThreads`) 和预读缓冲区数 (`readAheadBlocks`)。同时设置这些属性可以显著提高 CDC 任务的性能。以下设置为大多数 ASM 配置提供了良好的结果。

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_number
+ASM;
parallelASMReadThreads=6;readAheadBlocks=150000;
```

有关额外连接属性支持的值的更多信息，请参阅[使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。

此外，对使用 ASM 的 Oracle 源执行 CDC 任务的性能取决于您选择的其他设置。这些设置包括用于配置 Oracle 源的 AWS DMS 额外连接属性和 SQL 设置。有关使用 ASM 的 Oracle 源的额外连接属性的更多信息，请参阅[使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。

您还需要选择适当的 CDC 开始点。通常，在执行此操作时，您需要确定从哪个事务处理点捕获最早的未结事务，以便启动 CDC。否则，CDC 任务可能会错过之前的未结事务。对于 Oracle 源数据库，您可以根据 Oracle 系统更改号 ( SCN ) 选择 CDC 本机开始点，以此来确定最早的未结事务。有关更多信息，请参阅 [从 CDC 开始点开始执行复制](#)。

有关为使用自我管理 Oracle 数据库作为源配置 CDC 的更多信息，请参阅[使用 Oracle LogMiner 访问重做日志时所需的帐户权限](#)、[使用 AWS DMS Binary Reader 访问重做日志时所需的帐户权限](#)和[将 Binary Reader 与 Oracle ASM 配合使用时所需的其他帐户权限](#)。

有关将 CDC 配置为 AWS 托管的 Oracle 数据库作为源的更多信息，请参见 [将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS](#)和 [将 Amazon RDS Oracle Standby \( 只读副本 \) 作为源与 Binary Reader 配合用于在 AWS DMS 中执行 CDC](#)

## 配置自我管理或 AWS 托管的 Oracle 源数据库的工作流 AWS DMS

## 配置自我管理或 AWS 托管的 Oracle 源数据库的工作流 AWS DMS

要配置自我管理的源数据库实例，请使用以下工作流程步骤，具体取决于您执行 CDC 的方式。

在此工作流程步骤中	如果您使用执行 CDC LogMiner，请执行此操作	如果您使用 Binary Reader 执行 CDC，请执行此操作
授予 Oracle 帐户权限。	请参阅 <a href="#">在自行管理的 Oracle 源上需要用户帐户权限 AWS DMS</a> 。	请参阅 <a href="#">在自行管理的 Oracle 源上需要用户帐户权限 AWS DMS</a> 。
准备源数据库以便使用 CDC 进行复制。	请参阅 <a href="#">使用为 CDC 准备 Oracle 自行管理的源数据库 AWS DMS</a> 。	请参阅 <a href="#">使用为 CDC 准备 Oracle 自行管理的源数据库 AWS DMS</a> 。
为 CDC 授予所需的其他 Oracle 用户权限。	请参阅 <a href="#">使用 Oracle LogMiner 访问重做日志时所需的帐户权限</a> 。	请参阅 <a href="#">使用 AWS DMS Binary Reader 访问重做日志时所需的帐户权限</a> 。
对于使用 ASM 的 Oracle 实例，请为 CDC 授予访问 ASM 所需的其他用户帐户权限。	没有其他操作。AWS DMS 无需其他帐户权限即可支持 Oracle ASM。	请参阅 <a href="#">将 Binary Reader 与 Oracle ASM 配合使用时所需的其他帐户权限</a> 。

在此工作流程步骤中	如果您使用执行 CDC LogMiner，请执行此操作	如果您使用 Binary Reader 执行 CDC，请执行此操作
如果您尚未执行此操作，请配置要使用的任务 LogMiner 或用于 CDC 的 Binary Reader。	请参阅 <a href="#">在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器</a> 。	请参阅 <a href="#">在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器</a> 。
将 Oracle Standby 配置为源来执行 CDC。	AWS DMS 不支持 Oracle 备用服务器作为来源。	请参阅 <a href="#">将自管理 Oracle Standby 作为源与 Binary Reader 配合用于在 AWS DMS 中执行 CDC</a> 。

使用以下工作流程步骤配置 AWS 托管的 Oracle 源数据库实例。

在此工作流程步骤中	如果您使用执行 CDC LogMiner，请执行此操作	如果您使用 Binary Reader 执行 CDC，请执行此操作
授予 Oracle 账户权限。	有关更多信息，请参阅 <a href="#">需要在 AWS 托管的 Oracle 源上拥有用户帐户权限 AWS DMS</a> 。	有关更多信息，请参阅 <a href="#">需要在 AWS 托管的 Oracle 源上拥有用户帐户权限 AWS DMS</a> 。
准备源数据库以便使用 CDC 进行复制。	有关更多信息，请参阅 <a href="#">为配置由 AWS 托管的 Oracle 源 AWS DMS</a> 。	有关更多信息，请参阅 <a href="#">为配置由 AWS 托管的 Oracle 源 AWS DMS</a> 。
为 CDC 授予所需的其他 Oracle 用户权限。	不需要额外的账户权限。	有关更多信息，请参阅 <a href="#">将 CDC 任务配置为将二进制阅读器与 RDS for Oracle 源一起使用 AWS DMS</a> 。
如果您尚未执行此操作，请配置要使用的任务 LogMiner 或用于 CDC 的 Binary Reader。	有关更多信息，请参阅 <a href="#">在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器</a> 。	有关更多信息，请参阅 <a href="#">在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器</a> 。
将 Oracle Standby 配置为源来执行 CDC。	AWS DMS 不支持 Oracle 备用服务器作为来源。	有关更多信息，请参阅 <a href="#">将 Amazon RDS Oracle Standby (只读副本) 作为源</a>

在此工作流程步骤中	如果您使用执行 CDC LogMiner，请执行此操作	如果您使用 Binary Reader 执行 CDC，请执行此操作
		<a href="#">与 Binary Reader 配合用于在 AWS DMS 中执行 CDC。</a>

## 使用自行管理的 Oracle 数据库作为数据源 AWS DMS

自我管理数据库是您配置和控制的数据库，它是本地数据库实例或 Amazon EC2 上的数据库。接下来，您可以了解使用自行管理的 Oracle 数据库时所需的权限和配置。AWS DMS

在自行管理的 Oracle 源上需要用户帐户权限 AWS DMS

要使用 Oracle 数据库作为中的源 AWS DMS，请向在 Oracle 端点连接设置中指定的 Oracle 用户授予以下权限。

### Note

当授予权限时，请使用对象的实际名称，而不是每个对象的同义词。例如，请使用带下划线的 `V_$OBJECT`，而不是不带下划线的 `V$OBJECT`。

```
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 ## Oracle #####
##### 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.

```

当您使用特定表列表时，为每个复制表授予以下附加权限。

```
GRANT SELECT on any-replicated-table to db_user;
```

授予以下额外权限，即可使用验证功能验证 LOB 列。

```
GRANT EXECUTE ON SYS.DBMS_CRYPTO TO db_user;
```

如果您使用二进制读取器代替，则授予以下额外权限 LogMiner。

```
GRANT SELECT ON SYS.DBA_DIRECTORIES TO db_user;
```

授予以下额外权限以公开视图。

```
GRANT SELECT on ALL_VIEWS to dms_user;
```

要公开视图，您还必须向源端点添加额外连接属性 `exposeViews=true`。

使用无服务器复制时，请授予以下额外权限。

```
GRANT SELECT on dba_segments to db_user;
```

有关无服务器复制的更多信息，请参阅[使用无 AWS DMS 服务器](#)。

使用 Oracle 特定的迁移前评测时，请授予以下额外权限。

```
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;
```

有关 Oracle 特定迁移前评测的更多信息，请参阅[甲骨文评估](#)。

处理 Oracle Standby 未结事务的先决条件

使用 3.4.6 及更高 AWS DMS 版本时，请执行以下步骤来处理 Oracle 备用版本的未结事务。

1. 在主数据库上创建名为 `AWSDMS_DBLINK` 的数据库链接。`DMS_USER` 将使用此数据库链接连接到主数据库。请注意，此数据库链接是从备用实例执行的，用于查询主数据库上运行的未结事务。请参阅以下示例。

```
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. 验证是否已建立与使用 `DMS_USER` 的数据库链接的连接，如以下示例所示。

```
select 1 from dual@AWS_DMS_DBLINK
```

使用为 CDC 准备 Oracle 自行管理的源数据库 AWS DMS

通过执行以下操作，准备将您的 Oracle 自管理数据库作为运行 CDC 任务的源：

- [正在验证是否 AWS DMS 支持源数据库版本](#).
- [确保已开启 ARCHIVELOG 模式](#).
- [设置补充日志记录](#).

正在验证是否 AWS DMS 支持源数据库版本

运行以下示例查询，验证 AWS DMS 是否支持 Oracle 源数据库的当前版本。

```
SELECT name, value, description FROM v$parameter WHERE name = 'compatible';
```

此处，`name`、`value` 和 `description` 列是数据库中某个位置的列，正根据 `name` 的值查询这些列。如果此查询运行没有错误，则 AWS DMS 支持数据库的当前版本，您可以继续迁移。如果查询引发错误，则 AWS DMS 不支持数据库的当前版本。要继续迁移，请先将 Oracle 数据库转换为支持的版本 AWS DMS。

确保已开启 ARCHIVELOG 模式

您可以在两种不同的模式下运行 Oracle：ARCHIVELOG 模式和 NOARCHIVELOG 模式。要运行 CDC 任务，请在 ARCHIVELOG 模式下运行数据库。要知道数据库是否处于 ARCHIVELOG 模式，请执行以下查询。

```
SQL> SELECT log_mode FROM v$database;
```

如果返回到 NOARCHIVELOG 模式，请按照 Oracle 的指示将数据库设置为 ARCHIVELOG。

设置补充日志记录

要捕获正在进行的更改，AWS DMS 需要在 Oracle 源数据库上启用最低限度的补充日志记录。此外，您需要对数据库中每个已复制的表启用补充日志记录。

默认情况下，在所有复制的表上 AWS DMS 添加 PRIMARY KEY 补充日志记录。AWS DMS 要允许添加 PRIMARY KEY 补充日志记录，请为每个复制的表授予以下权限。

```
ALTER on any-replicated-table;
```

您可以 AWS DMS 使用 extra 连接属性 `addSupplementalLogging` 禁用添加的默认 PRIMARY KEY 补充日志记录。有关更多信息，请参阅 [使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。

如果您的复制任务使用未引用主键列的 WHERE 子句更新表，请务必打开补充日志记录。

## 要手动设置补充日志记录

1. 运行以下查询来验证是否为数据库启用了补充日志记录。

```
SELECT supplemental_log_data_min FROM v$database;
```

如果返回的结果为 YES 或 IMPLICIT，则为数据库启用了补充日志记录。

否则，请通过运行以下命令，为数据库启用补充日志记录。

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

2. 确保为每个复制的表添加了所需的补充日志记录。

请考虑以下事项：

- 如果向表中添加了 ALL COLUMNS 补充日志记录，则无需添加更多日志记录。
- 如果存在主键，请为主键添加补充日志记录。为此，您可以使用格式在主键上添加补充日志记录，也可以在数据库的主键列上添加补充日志记录。

```
ALTER TABLE Tablename ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;  
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
```

- 如果不存在主键并且表具有单个唯一索引，则将所有唯一索引的列添加到补充日志。

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG GROUP LogGroupName  
(UniqueIndexColumn1[, UniqueIndexColumn2] ...) ALWAYS;
```

使用 SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS 不会将唯一索引列添加到日志。

- 如果不存在主键且表有多个唯一索引，则在按字母顺序升序列表 AWS DMS 中选择第一个唯一索引。您需要像上一项那样在所选索引的列上添加补充日志记录。

使用 SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS 不会将唯一索引列添加到日志。

- 如果不存在主键，也没有唯一索引，则在所有列上添加补充日志记录。

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

在某些情况下，目标表主键或唯一索引不同于源表主键或唯一索引。在这些情况下，请在构成目标表主键或唯一索引的源表列上手动添加补充日志记录。

此外，如果您更改目标表主键，请在目标唯一索引的列上添加补充日志记录，而不是在原始主键或唯一索引的列上添加。

如果为表定义了筛选器或转换，则可能需要启用其他日志记录。

请考虑以下事项：

- 如果向表中添加了 ALL COLUMNS 补充日志记录，则无需添加更多日志记录。
- 如果表具有唯一索引或主键，请在筛选器或转换中涉及的每一列上添加补充日志记录。但是，仅在这些列与主键列或唯一索引列不同时才执行此操作。
- 如果转换只包含一列，请不要将此列添加到补充日志记录组。例如，对于转换 A+B，请在 A 和 B 这两列上添加补充日志记录。但是，对于转换 substring(A,10)，请不要在列 A 上添加补充日志记录。
- 要在主键或唯一索引列以及经过筛选或转换的其他特定列上设置补充日志记录，可以设置 USER\_LOG\_GROUP 补充日志记录。在主键或唯一索引列和经过筛选或转换的其他特定列上添加此日志记录。

例如，要复制名为 TEST.LOGGING 的表，该表的主键为 ID 且按列 NAME 进行筛选，您可以运行如下命令来创建日志组补充日志记录。

```
ALTER TABLE TEST.LOGGING ADD SUPPLEMENTAL LOG GROUP TEST_LOG_GROUP (ID, NAME) ALWAYS;
```

## 使用 Oracle LogMiner 访问重做日志时所需的帐户权限

要使用 Oracle 访问重做日志 LogMiner，请向在 Oracle 端点连接设置中指定的 Oracle 用户授予以下权限。

```
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.
```

## 使用 AWS DMS Binary Reader 访问重做日志时所需的帐户权限

要使用 AWS DMS 二进制读取器访问重做日志，请向在 Oracle 端点连接设置中指定的 Oracle 用户授予以下权限。

```
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 与 Oracle 文件功能（包括 Oracle 目录）配合使用。每个 Oracle 目录对象都包含要处理的重做日志文件所在文件夹的名称。这些 Oracle 目录不在文件系统级别表示。相反，它们是在 Oracle 数据库级别创建的逻辑目录。您可以在 Oracle ALL\_DIRECTORIES 视图中查看它们。

如果 AWS DMS 要创建这些 Oracle 目录，请授予前面指定的 CREATE ANY DIRECTORY 权限。AWS DMS 创建带有 DMS\_ 前缀的目录名称。如果您不授予 CREATE ANY DIRECTORY 权限，请手动创建相应的目录。在某些情况下，当您手动创建 Oracle 目录时，Oracle 源终端节点中指定的 Oracle 用户不是创建这些目录的用户。在这些情况下，还要授予 READ on DIRECTORY 特权。

如果 Oracle 源端点处于 Active Dataguard 待机 (ADG) 状态，请参阅数据库博客上的 [“如何将 Binary Reader 与 ADG 一起使用 Binary Reader”](#) 一文。AWS

### Note

AWS DMS CDC 不支持未配置为使用自动重做传输服务的 Active Dataguard 待机。

在某些情况下，您可能会使用 Oracle 管理的文件 (OMF) 来存储日志。或者，您的源端点位于 ADG 中，并且无法授予 CREATE ANY DIRECTORY 权限。在这些情况下，请在开始 AWS DMS 复制任务之前手动创建包含所有可能日志位置的目录。如果 AWS DMS 找不到预先创建的目录，则任务将停止。此外，AWS DMS 不会删除它在 ALL\_DIRECTORIES 视图中创建的条目，因此应手动删除这些条目。

将 Binary Reader 与 Oracle ASM 配合使用时所需的其他账户权限

要使用 Binary Reader 访问 Automatic Storage Management (ASM) 中的重做日志，请向 Oracle 端点连接设置中指定的 Oracle 用户授予以下权限。

```
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.
```

您可以通过打开命令提示符并调用以下语句之一来验证 ASM 账户访问权限，具体取决于前面指定的 Oracle 版本。

如果您需要 SYSDBA 权限，请使用以下语句。

```
sqlplus asmuser/asmpassword@+asmserver as sysdba
```

如果您需要 SYSASM 权限，请使用以下语句。

```
sqlplus asmuser/asmpassword@+asmserver as sysasm
```

将自我管理 Oracle Standby 作为源与 Binary Reader 配合用于在 AWS DMS 中执行 CDC

要在使用 Binary Reader 执行 CDC 时将 Oracle Standby 实例配置为源，请先满足以下先决条件：

- AWS DMS 目前仅支持 Oracle 主动数据防护待机。
- 确保 Oracle Data Guard 配置使用：
  - 重做传输服务以自动传输重做数据。
  - 应用服务以自动对备用数据库应用重做。

要确认是否满足这些要求，请执行以下查询。

```
SQL> select open_mode, database_role from v$database;
```

从该查询的输出中，确认备用数据库是否以只读模式打开，并且正在自动应用重做。例如：

```
OPEN_MODE          DATABASE_ROLE
-----
READ ONLY WITH APPLY  PHYSICAL STANDBY
```

要在使用 Binary Reader 执行 CDC 时将 Oracle Standby 实例配置为源

1. 授予访问待机日志文件所需的其他权限。

```
GRANT SELECT ON v_$standby_log TO db_user;
```

2. 使用 AWS Management Console 或 AWS CLI 为 Oracle Standby 创建源端点。创建端点时，请指定以下额外的连接属性。

```
useLogminerReader=N;useBfile=Y;
```

#### Note

在中 AWS DMS，您可以使用额外的连接属性来指定是否要从存档日志而不是重做日志中迁移。有关更多信息，请参阅 [使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。

3. 配置存档日志目标。

未配置 ASM 的 Oracle 源的 DMS Binary Reader 使用 Oracle 目录来访问存档的重做日志。如果您的数据库配置为使用闪回恢复区 (FRA) 作为存档日志目标，则存档重做文件的位置不固定。每天生成存档的重做日志会导致在 FRA 中创建一个目录名称格式为 YYYY\_MM\_DD 的新目录。例如：

```
DB_RECOVERY_FILE_DEST/SID/archivelog/YYYY_MM_DD
```

当 DMS 需要访问在新创建的 FRA 目录中存档的重做文件，并且使用主读写数据库作为源时，DMS 会创建一个新的 Oracle 目录或替换现有的 Oracle 目录，如下所示。

```
CREATE OR REPLACE DIRECTORY dmsrep_taskid AS 'DB_RECOVERY_FILE_DEST/SID/archivelog/YYYY_MM_DD';
```

如果使用备用数据库作为源，DMS 无法创建或替换 Oracle 目录，因为该数据库处于只读模式。但是，您可以选择执行以下附加步骤之一：

- a. 将 `log_archive_dest_id_1` 改为使用实际路径，而不使用 Oracle 不会创建每日子目录的配置中的 FRA：

```
ALTER SYSTEM SET log_archive_dest_1='LOCATION=full directory path'
```

然后，创建一个供 DMS 使用的 Oracle 目录对象：

```
CREATE OR REPLACE DIRECTORY dms_archived_logs AS 'full directory path';
```

- b. 创建一个额外的存档日志目标和一个指向该目标的 Oracle 目录对象。例如：

```
ALTER SYSTEM SET log_archive_dest_3='LOCATION=full directory path';  
CREATE DIRECTORY dms_archived_log AS 'full directory path';
```

然后向任务源端点添加一个额外的连接属性：

```
archivedLogDestId=3
```

- c. 手动预创建一个供 DMS 使用的 Oracle 目录对象。

```
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. 创建每天运行的 Oracle 调度程序任务，并创建所需的目录。

在 Oracle Cloud Infrastructure (OCI) 上使用用户管理的数据库作为在 AWS DMS 中执行 CDC 的源

用户管理的数据库是您配置和控制的数据库，例如在虚拟机 (VM)、裸机或 Exadata 服务器上创建的 Oracle 数据库。或是您配置和控制的在专用基础架构上运行的数据库，例如 Oracle Cloud Infrastructure (OCI)。以下信息描述了在 OCI 上使用 Oracle 用户管理的数据库作为在 AWS DMS 中执行更改数据捕获 (CDC) 源时所需的权限和配置。

要将由 OCI 托管的用户管理的 Oracle 数据库配置为更改数据捕获源

1. 为 OCI 上用户管理的 Oracle 源数据库授予所需的用户账户权限。有关更多信息，请参阅[自我管理 Oracle 源端点的账户权限](#)。
2. 授予使用 Binary Reader 访问重做日志时所需的账户权限。有关更多信息，请参阅[使用 Binary Reader 时所需的账户权限](#)。
3. 添加将 Binary Reader 与 Oracle Automatic Storage Management ( ASM ) 结合使用时所需的账户权限。有关更多信息，请参阅[将 Binary Reader 与 Oracle ASM 结合使用时所需的其他账户权限](#)。
4. 设置补充日志记录。有关更多信息，请参阅[设置补充日志记录](#)。
5. 设置 TDE 加密。有关更多信息，请参阅[将 Oracle 数据库用作源端点时使用的加密方法](#)。

在 Oracle Cloud Infrastructure ( OCI ) 上从 Oracle 源数据库复制数据时，存在以下限制。

限制

- DMS 不支持使用 Oracle LogMiner 访问重做日志。
- DMS 不支持自治数据库。

## 使用 AWS 由托管的 Oracle 数据库作为数据源 AWS DMS

AWS 托管数据库是位于亚马逊服务（例如 Amazon RDS、Amazon Aurora 或 Amazon S3）上的数据库。接下来，您可以找到使用 AWS 托管的 Oracle 数据库时需要设置的权限和配置。AWS DMS

需要在 AWS 托管的 Oracle 源上拥有用户帐户权限 AWS DMS

向 Oracle 源端点定义中指定的 Oracle 用户账户授予以下权限。

### Important

对于所有参数值（例如 *db\_user* 和 *any-replicated-table*），除非您使用区分大小写的标识符指定值，否则 Oracle 会假定值全部大写。例如，假设您在创建 *db\_user* 值时未使用引号，例如 `CREATE USER myuser` 或 `CREATE USER MYUSER`。在这种情况下，Oracle 会识别此值并以全部大写的形式存储（`MYUSER`）。如果使用引号（例如 `CREATE USER "MyUser"` 或 `CREATE USER 'MyUser'`），Oracle 会识别并存储您指定的区分大小写的值（`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.
```

此外，还要使用如下所示的 Amazon RDS 过程 `rdsadmin.rdsadmin_util.grant_sys_object` 为 SYS 对象授予 SELECT 和 EXECUTE 权限。有关更多信息，请参阅[向 SYS 对象授予 SELECT 或 EXECUTE 权限](#)。

```
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');
```

有关将 Amazon RDS Active Dataguard Standby ( ADG ) 与 AWS DMS 配合使用的更多信息，请参阅[将 Amazon RDS Oracle Standby \( 只读副本 \) 作为源与 Binary Reader 配合用于在 AWS DMS 中执行 CDC](#)。

有关将 Oracle TDE 与配合使用的更多信息 AWS DMS，请参阅[使用 Oracle 作为数据源的支持的加密方法 AWS DMS](#)。

处理 Oracle Standby 未结事务的先决条件

使用 3.4.6 及更高 AWS DMS 版本时，请执行以下步骤来处理 Oracle 备用版本的未结事务。

1. 在主数据库上创建名为 AWSDMS\_DBLINK 的数据库链接。*DMS\_USER* 将使用此数据库链接连接到主数据库。请注意，此数据库链接是从备用实例执行的，用于查询主数据库上运行的未结事务。请参阅以下示例。

```
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. 验证是否已建立与使用 *DMS\_USER* 的数据库链接的连接，如以下示例所示。

```
select 1 from dual@AWS_DMS_DBLINK
```

## 为配置由 AWS 托管的 Oracle 源 AWS DMS

在使用 AWS 托管的 Oracle 数据库作为源之前 AWS DMS，请对 Oracle 数据库执行以下任务：

- 启用自动备份。有关启用自动备份的更多信息，请参阅《Amazon RDS 用户指南》中的[启用自动备份](#)。
- 设置补充日志记录。
- 设置存档。存档适用于 Oracle 的 Amazon RDS 数据库实例的重做日志 AWS DMS 允许使用 Oracle LogMiner 或 Binary Reader 检索日志信息。

## 设置存档

1. 运行 `rdsadmin.rdsadmin_util.set_configuration` 命令以设置存档。

例如，要将已存档的重做日志保留 24 小时，请运行以下命令。

```
exec rdsadmin.rdsadmin_util.set_configuration('archive_log retention hours',24);
commit;
```

### Note

要使更改生效，必须执行提交操作。

2. 确保您的存储具有足够的空间，可以满足指定保留期内的存档重做日志需求。例如，如果您的保留期为 24 小时，请计算在典型的事务处理小时内累积的存档重做日志的总大小，然后将该总大小乘以 24。将计算得出的 24 小时总存储空间与您的可用存储空间进行比较，确定是否有足够的存储空间来处理整整 24 小时的事务。

## 设置补充日志记录

1. 运行以下命令在数据库级别启用补充日志记录。

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');
```

2. 运行以下命令为主键列启用补充日志记录。

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD','PRIMARY KEY');
```

3. ( 可选 ) 在表级别启用键级别补充日志记录。

启用键级别补充日志记录时，您的源数据库会产生少量开销。因此，如果您仅迁移一部分表，可能需要在表级别启用键级别补充日志记录。要在表级别启用键级别补充日志记录，请运行以下命令。

```
alter table table_name add supplemental log data (PRIMARY KEY) columns;
```

将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS

您可以配置 AWS DMS 为使用适用于 CDC 的二进制读取器访问源 Amazon RDS for Oracle 实例重做日志。

### Note

要使用 Oracle LogMiner，所需的最低用户帐户权限就足够了。有关更多信息，请参阅 [需要在 AWS 托管的 Oracle 源上拥有用户帐户权限 AWS DMS](#)。

要使用 AWS DMS Binary Reader，请根据您的 AWS DMS 版本为 Oracle 源端点指定其他设置和额外连接属性。

以下版本的 Amazon RDS for Oracle 支持 Binary Reader：

- Oracle 11.2 – 版本 11.2.0.4V11 及更高版本
- Oracle 12.1 – 版本 12.1.0.2.V7 及更高版本
- Oracle 12.2 – 所有版本
- Oracle 18.0 – 所有版本
- Oracle 19.0 – 所有版本

## 使用 Binary Reader 配置 CDC

1. 以主用户身份登录您的 Amazon RDS for Oracle 源数据库，然后运行以下存储过程来创建服务器级目录。

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

2. 向用于访问 Oracle 源端点的 Oracle 用户账户授予以下权限。

```
GRANT READ ON DIRECTORY ONLINELOG_DIR TO db_user;
GRANT READ ON DIRECTORY ARCHIVELOG_DIR TO db_user;
```

3. 在 Amazon RDS Oracle 源端点上设置以下额外连接属性：

- 对于 RDS Oracle 版本 11.2 和 12.1，请设置以下内容。

```
useLogminerReader=N;useBfile=Y;accessAlternateDirectly=false;useAlternateFolderForOnline=
oraclePathPrefix=/rdsdbdata/db/{"$DATABASE_NAME"}_A/;usePathPrefix=/rdsdbdata/
log/;replacePathPrefix=true;
```

- 对于 RDS Oracle 版本 12.2、18.0 和 19.0，请设置以下内容。

```
useLogminerReader=N;useBfile=Y;
```

### Note

对于多个属性设置，请确保分号分隔符 ( ; ) 后面没有空格 ( 例如，`oneSetting;thenAnother` )。

有关配置 CDC 任务的更多信息，请参阅[Oracle 源数据库上的 CDC 的配置](#)。

将 Amazon RDS Oracle Standby ( 只读副本 ) 作为源与 Binary Reader 配合用于在 AWS DMS 中执行 CDC

要将 Amazon RDS for Oracle Standby 作为在 AWS DMS 中使用 Binary Reader 执行 CDC 的源，请验证以下先决条件：

- 使用 Oracle 主用户设置 Binary Reader。

- 确保 AWS DMS 当前仅支持使用 Oracle 主动数据防护待机。

完成此操作后，请按照以下步骤将 RDS for Oracle Standby 作为使用 Binary Reader 执行 CDC 的源。

要在使用 Binary Reader 执行 CDC 时将 RDS for Oracle Standby 配置为源

1. 以主用户身份登录 RDS for Oracle 主实例。
2. 运行《Amazon RDS 用户指南》中记录的以下存储过程来创建服务器级目录。

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

3. 确定在步骤 2 中创建的目录。

```
SELECT directory_name, directory_path FROM all_directories
WHERE directory_name LIKE ( 'ARCHIVELOG_DIR_%' )
      OR directory_name LIKE ( 'ONLINELOG_DIR_%' )
```

例如，前面的代码显示了如下所示的目录列表。

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. 向用于访问 Oracle Standby 的 Oracle 用户账户授予上述目录的 Read 权限。

```
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. 在主实例上执行存档日志切换。这样做可以确保将对 ALL\_DIRECTORIES 的更改也移植到 Oracle Standby。
6. 在 Oracle Standby 上运行 ALL\_DIRECTORIES 查询来确认已应用更改。
7. 使用 AWS DMS 管理控制台或 AWS Command Line Interface (AWS CLI) 为 Oracle 备用服务器创建源端点。创建端点时，请指定以下额外的连接属性。

```
useLogminerReader=N;useBfile=Y;archivedLogDestId=1;additionalArchivedLogDestId=2
```

8. 创建终端节点后，使用控制台的“创建端点”页面上的测试端点连接或 AWS CLI `test-connection` 命令来验证连接是否已建立。

## 使用 Oracle 作为来源的限制 AWS DMS

将 Oracle 数据库作为 AWS DMS 源时，存在以下限制：

- AWS DMS 支持 3.5.0 及更高 AWS DMS 版本中的 Oracle 扩展数据类型。
- AWS DMS 不支持长对象名（超过 30 字节）。
- AWS DMS 不支持基于函数的索引。
- 如果您管理补充日志记录并对任何列执行转换，请确保为所有字段和列激活补充日志记录。有关设置补充日志记录的信息，请参阅以下主题：
  - 对于自管理 Oracle 源数据库，请参阅[设置补充日志记录](#)。
  - 有关 AWS 托管的 Oracle 源数据库，请参见[为配置由 AWS 托管的 Oracle 源 AWS DMS](#)。
- AWS DMS 不支持多租户容器根数据库 (CDB\$ROOT)。但它支持使用 Binary Reader 的 PDB。
- AWS DMS 不支持延迟约束。
- 在 3.5.1 及更高 AWS DMS 版本中，只有通过执行 LOB 查找才能支持安全 LOB。
- AWS DMS 支持所有支持的 Oracle 版本 11 及更高版本的 `rename table table-name to new-table-name` 语法。但不支持任何 Oracle 版本 10 的源数据库中的此语法。
- AWS DMS 不会复制 DDL 语句 `ALTER TABLE ADD column data_type DEFAULT default_value` 的结果。它将新列设置为 NULL，而不是将 `default_value` 复制到目标。
- 使用 3.4.7 或更高 AWS DMS 版本时，要复制因分区或子分区操作而产生的更改，请在启动 DMS 任务之前执行以下操作。
  - 手动创建分区表结构 (DDL) ；
  - 确保 Oracle 源和 Oracle 目标上的 DDL 相同 ；
  - 设置额外的连接属性 `enableHomogenousPartitionOps=true`。

有关 `enableHomogenousPartitionOps` 的更多信息，请参阅[使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。另请注意，在 FULL+CDC 任务中，DMS 不会复制作为缓存更改的一部分捕获的数据更改。在该使用案例中，将在 Oracle 目标上重新创建表结构并重新加载有问题的表。

3.4.7 之前的 AWS DMS 版本：

DMS 不复制由于分区或子分区操作 ( ADD、DROP、EXCHANGE 和 TRUNCATE ) 而导致的数据更改。此类更新可能会导致复制过程中出现以下错误：

- 对于 ADD 操作，对添加的数据进行更新和删除可能会引发“0 rows affected (0 行受到影响)”警告。
- 对于 DROP 和 TRUNCATE 操作，新插入可能会引发“duplicates (重复项)”错误。
- EXCHANGE 操作可能会引发“0 rows affected (0 行受到影响)”警告和“duplicates (重复项)”错误。

要复制由于分区或子分区操作而导致的更改，请重新加载有问题的表。添加新的空分区之后，对新添加的表执行的操作将正常复制到目标。

- AWS DMS 3.4 之前的版本不支持在源代码上运行 CREATE TABLE AS 语句所导致的目标数据更改。但是，在目标上会创建新表。
- AWS DMS 不捕获 Oracle DBMS\_REDEFINITION 包所做的更改，例如表元数据和 OBJECT\_ID 字段。
- AWS DMS 将空的 BLOB 和 CLOB 列映射到 NULL 目标上。
- 使用 Oracle 11 捕获更改时 LogMiner，字符串长度大于 1982 的 CLOB 列的更新会丢失，目标也不会更新。
- 在更改数据捕获 (CDC) 期间，AWS DMS 不支持对定义为主键的数字列进行批量更新。
- AWS DMS 不支持某些 UPDATE 命令。以下示例是一个不支持的 UPDATE 命令。

```
UPDATE TEST_TABLE SET KEY=KEY+1;
```

此处，TEST\_TABLE 是表名称，KEY 是定义为主键的数值列。

- AWS DMS 不支持完整的 LOB 模式来加载长整的 RAW 列和 LONG RAW 列。但您可以使用受限的 LOB 模式将这些数据类型迁移到 Oracle 目标。在受限 LOB 模式下，将所有设置为长度超过 64 KB 的 LONG 或 LONG RAW 列的数据 AWS DMS 截断为 64 KB。
- AWS DMS 不支持用于加载 XMLTYPE 列的完整 LOB 模式。但您可以使用受限的 LOB 模式将 XMLTYPE 列迁移到 Oracle 目标。在受限 LOB 模式下，DMS 会截断大于用户定义的“最大 LOB 大小”变量的任何数据。“最大 LOB 大小”的最大建议值为 100MB。
- AWS DMS 不会复制名称包含撇号的表。
- AWS DMS 从物化视图中支持 CDC。但是，DMS 不支持从其他任何视图执行 CDC。
- AWS DMS 不支持 CDC 用于按索引组织的带有溢出段的表。
- AWS DMS 不支持对按引用分区且 enableHomogenousPartitionOps 设置为 true 的表 Drop Partition 执行操作。



- 使用 Oracle LogMiner 访问重做日志时，AWS DMS 有以下限制：
  - 仅适用于 Oracle 12，AWS DMS 不会复制对 LOB 列的任何更改。
  - 对于所有 Oracle 版本，AWS DMS 不会复制对 LOB 列 XMLTYPE 和 LOB 列进行 UPDATE 操作的结果。
  - AWS DMS 使用 Oracle 时不支持复制中的 XA 事务 LogMiner。
  - Oracle LogMiner 不支持与可插拔数据库 (PDB) 的连接。要连接到 PDB，请使用 Binary Reader 访问重做日志。
  - 不支持缩小空间操作。
- 使用二进制阅读器时，AWS DMS 有以下限制：
  - 它不支持表集群。
  - 它仅支持表级别的 SHRINK SPACE 操作。此级别包括完整的表、分区和子分区。
  - 它不支持使用键压缩对索引组织的表进行更改。
  - 它不支持在原始设备上实现在线重做日志。
  - Binary Reader 仅支持自我管理 Oracle 数据库的 TDE，因为 RDS for Oracle 不支持为 TDE 加密密钥检索 Wallet 密码。
- AWS DMS 不支持使用 Oracle 自动存储管理 (ASM) 代理连接到 Amazon RDS Oracle 来源。
- AWS DMS 不支持虚拟列。
- AWS DMS 不支持 ROWID 数据类型或基于 ROWID 列的实例化视图。

AWS DMS 部分支持 Oracle 实体化视图。对于完全加载，DMS 可以执行 Oracle 实体化视图的完全加载副本。DMS 将实体化视图作为基表复制到目标系统，并忽略实体化视图中的所有 ROWID 列。对于持续复制 (CDC)，DMS 会尝试复制对实体化视图数据的更改，但结果可能并不理想。具体而言，如果实体化视图已完全刷新，DMS 会复制所有行的单个删除操作，然后复制所有行的单个插入操作。这是一种资源密集型操作，对于具有大量行的实体化视图，可能表现不佳。对于实例化视图快速刷新的持续复制，DMS 会尝试处理并复制快速刷新数据更改。无论哪种情况，DMS 都会跳过实体化视图中的所有 ROWID 列。

- AWS DMS 不会加载或捕获全局临时表。
- 对于使用复制的 S3 目标，请在每列上启用补充日志记录，以便源行更新可以捕获每列的值。下面是示例：

```
alter table yourtablename add supplemental log data (all) columns;
```
- 无法在目标上复制使用包含 null 的复合唯一键的行的更新。
- AWS DMS 不支持在同一个源端点上使用多个 Oracle TDE 加密密钥。每个端点只能有一个用于 TDE 加密密钥名称“securityDbEncryptionName”的属性，以及该密钥的一个 TDE 密码。
- 从 Amazon RDS for Oracle 进行复制时，只有加密的表空间和使用 Oracle 才支持 TDE。LogMiner

- AWS DMS 不支持快速连续执行多个表重命名操作。
- 使用 Oracle 19.0 作为源代码时，AWS DMS 不支持以下功能：
  - 数据保护 DML 重定向
  - 混合分区表
  - 仅架构的 Oracle 账户
- AWS DMS 不支持迁移类型为或的表BIN\$或视图DR\$。
- 从 Oracle 18.x 开始，AWS DMS 不支持 Oracle Express 版 ( Oracle Database XE ) 中的变更数据捕获 (CDC)。
- 从 CHAR 列迁移数据时，DMS 会截断所有尾随空格。
- AWS DMS 不支持从应用程序容器复制。
- AWS DMS 不支持执行 Oracle 闪回数据库和还原点，因为这些操作会影响 Oracle 重做日志文件的一致性。
- 在以下情况下，不支持使用并行执行选项的直接加载 INSERT 过程：
  - 超过 255 列的未压缩表
  - 行大小超过 8K
  - Exadata HCC 表
  - 在 Big Endian 平台上运行的数据库
- 既没有主键也没有唯一键的源表需要启用 ALL COLUMN 补充日志记录。它会创建更多的重做日志活动，并可能增加 DMS CDC 延迟。
- AWS DMS 不会迁移源数据库中不可见列的数据。要将这些列纳入迁移范围内，请使用 ALTER TABLE 语句使这些列可见。

## Oracle 终端节点的 SSL 支持

AWS DMS Oracle 终端节点支持适用于none和 SS verify-ca L 模式的 SSL V3。要将 SSL 用于 Oracle 终端节点，请上传终端节点的 Oracle wallet 而不是 .pem 证书文件。

### 主题

- [为 Oracle SSL 使用现有证书](#)
- [使用适用于 Oracle SSL 的自签名证书](#)

## 为 Oracle SSL 使用现有证书

要使用现有的 Oracle 客户端安装，从 CA 证书文件创建 Oracle wallet 文件，请执行以下步骤。

### 在 AWS DMS 中为 Oracle SSL 使用现有 Oracle 客户端安装

1. 通过运行以下命令将 ORACLE\_HOME 系统变量设置为 dbhome\_1 目录的位置。

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

2. 将 \$ORACLE\_HOME/lib 附加到 LD\_LIBRARY\_PATH 系统变量。

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. 为 \$ORACLE\_HOME/ssl\_wallet 处的 Oracle wallet 创建一个目录。

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

4. 将 CA 证书 .pem 文件放入 ssl\_wallet 目录中。如果使用 Amazon RDS，则可下载由 Amazon RDS 托管的 rds-ca-2015-root.pem 根 CA 证书文件。有关下载此文件的更多信息，请参阅《Amazon RDS 用户指南》中的[使用 SSL/TLS 来加密与数据库实例的连接](#)。
5. 运行下列命令以创建 Oracle wallet。

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
$ORACLE_HOME/ssl_wallet/ca-cert.pem -auto_login_only
```

完成前面的步骤之后，您可以通过指定 certificate-wallet 参数，使用 ImportCertificate API 调用来导入 wallet 文件。然后，在创建或修改 Oracle 终端节点时，如果选择 verify-ca 作为 SSL 模式，您可以使用导入的 wallet 证书。

**Note**

甲骨文钱包是二进制文件。AWS DMS 按原样接受这些文件。

使用适用于 Oracle SSL 的自签名证书

要将自签名证书用于 Oracle SSL，请假设 Oracle 钱包密码为 `oracle123`，执行以下步骤。

要使用适用于 Oracle SSL 的自签名证书 AWS DMS

1. 创建一个您将用于处理自签名证书的目录。

```
mkdir -p /u01/app/oracle/self_signed_cert
```

2. 转到您在上一步创建的目录中。

```
cd /u01/app/oracle/self_signed_cert
```

3. 创建根密钥。

```
openssl genrsa -out self-rootCA.key 2048
```

4. 使用在上一步中创建的根密钥对根证书进行自签名。

```
openssl req -x509 -new -nodes -key self-rootCA.key  
-sha256 -days 3650 -out self-rootCA.pem
```

使用以下输入参数：

- Country Name (2 letter code) [XX]，例如：AU
- State or Province Name (full name) []，例如：NSW
- Locality Name (e.g., city) [Default City]，例如：Sydney
- Organization Name (e.g., company) [Default Company Ltd]，例如：AmazonWebService
- Organizational Unit Name (e.g., section) []，例如：DBeng
- Common Name (e.g., your name or your server's hostname) []，例如：aws
- Email Address []，例如：abcd.efgh@amazonwebservice.com

## 5. 为 Oracle 数据库创建 Oracle wallet 目录。

```
mkdir -p /u01/app/oracle/wallet
```

## 6. 创建新的 Oracle wallet。

```
orapki wallet create -wallet "/u01/app/oracle/wallet" -pwd oracle123 -  
auto_login_local
```

## 7. 将根证书添加到 Oracle wallet。

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123 -trusted_cert  
-cert /u01/app/oracle/self_signed_cert/self-rootCA.pem
```

## 8. 列出 Oracle wallet 的内容。该列表应包含根证书。

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

例如，这可能显示以下类似内容。

```
Requested Certificates:  
User Certificates:  
Trusted Certificates:  
Subject:          CN=aws,OU=DBeng,O= AmazonWebService,L=Sydney,ST=NSW,C=AU
```

## 9. 使用 ORAPKI 实用程序生成证书签名请求 (CSR)。

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123  
-dn "CN=aws" -keysize 2048 -sign_alg sha256
```

## 10. 运行以下命令。

```
openssl pkcs12 -in /u01/app/oracle/wallet/ewallet.p12 -nodes -out /u01/app/oracle/  
wallet/nonoracle_wallet.pem
```

这会产生类似下面的输出。

```
Enter Import Password:  
MAC verified OK  
Warning unsupported bag type: secretBag
```

## 11. 将“dms”用作公用名。

```
openssl req -new -key /u01/app/oracle/wallet/nonoracle_wallet.pem -out certdms.csr
```

使用以下输入参数：

- Country Name (2 letter code) [XX]，例如：AU
- State or Province Name (full name) []，例如：NSW
- Locality Name (e.g., city) [Default City]，例如：Sydney
- Organization Name (e.g., company) [Default Company Ltd]，例如：AmazonWebService
- Organizational Unit Name (e.g., section) []，例如：aws
- Common Name (e.g., your name or your server's hostname) []，例如：aws
- Email Address []，例如：abcd.efgh@amazonweb-service.com

请确保这与步骤 4 不同。例如，您可以通过将组织单位名称更改为其他名称来实现此目的，如下所示。

输入要随证书申请一起发送的以下其他属性。

- A challenge password []，例如：oracle123
- An optional company name []，例如：aws

## 12. 获取证书签名。

```
openssl req -noout -text -in certdms.csr | grep -i signature
```

这篇文章的签名密钥是 sha256WithRSAEncryption。

## 13. 运行以下命令以生成证书 (.crt) 文件。

```
openssl x509 -req -in certdms.csr -CA self-rootCA.pem -CAkey self-rootCA.key  
-CAcreateserial -out certdms.crt -days 365 -sha256
```

这会产生类似下面的输出。

```
Signature ok
```

```
subject=C=AU/ST=NSW/L=Sydney/O=awsweb/OU=DBGeng/CN=aws
```

## Getting CA Private Key

14. 将证书添加到 wallet。

```
orapki wallet add -wallet /u01/app/oracle/wallet -pwd oracle123 -user_cert -cert certdms.crt
```

15. 查看 Wallet。它应该包含两个条目。请参阅以下代码。

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

16. 配置 sqlnet.ora 文件 ( \$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. 停止 Oracle 侦听器。

```
lsnrctl stop
```

18. 在 listener.ora 文件 ( \$ORACLE\_HOME/network/admin/listener.ora ) 中添加 SSL 条目。

```
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 =
(AADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
(AADDRESS = (PROTOCOL = TCPS)(HOST = localhost.localdomain)(PORT = 1522))
(AADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
)
)
)

```

## 19. 配置 tnsnames.ora 文件 ( \$ORACLE\_HOME/network/admin/tnsnames.ora )。

```

<SID>=
(DESCRIPTION=
(AADDRESS_LIST =
(AADDRESS=(PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT =
1521))
)
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = <SID>)
)
)

<SID>_ssl=
(DESCRIPTION=
(AADDRESS_LIST =
(AADDRESS=(PROTOCOL = TCPS)(HOST = localhost.localdomain)(PORT =
1522))
)
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = <SID>)
)
)
)

```

## 20. 重新启动 Oracle 侦听器。



```
lsnrctl start
```

21. 显示 Oracle 侦听器状态。

```
lsnrctl status
```

22. 使用 sqlplus 和 SSL tnsnames 条目测试从本地主机到数据库的 SSL 连接。

```
sqlplus -L ORACLE_USER@SID_ssl
```

23. 验证您已使用 SSL 成功连接。

```
SELECT SYS_CONTEXT('USERENV', 'network_protocol') FROM DUAL;

SYS_CONTEXT('USERENV', 'NETWORK_PROTOCOL')
-----
tcps
```

24. 将目录更改为具有自签名证书的目录。

```
cd /u01/app/oracle/self_signed_cert
```

25. 创建一个新的客户端 Oracle 钱包 AWS DMS 以供使用。

```
orapki wallet create -wallet ./ -auto_login_only
```

26. 将自签名根证书添加到 Oracle wallet。

```
orapki wallet add -wallet ./ -trusted_cert -cert self-rootCA.pem -auto_login_only
```

27. 列出 AWS DMS 要使用的 Oracle 钱包中的内容。该列表应包含自签名根证书。

```
orapki wallet display -wallet ./
```

这会产生类似下面的输出。

```
Trusted Certificates:
Subject:          CN=aws,OU=DBeng,O=AmazonWebService,L=Sydney,ST=NSW,C=AU
```

28. 将您刚刚创建的 Oracle 钱包上传到 AWS DMS。

## 使用 Oracle 作为数据源的支持的加密方法 AWS DMS

在下表中，您可以找到使用 Oracle 源数据库时 AWS DMS 支持的透明数据加密 (TDE) 方法。

重做日志访问方法	TDE 表空间	TDE 列
甲骨文 LogMiner	是	是
Binary Reader	是	是

AWS DMS 在列级别和表空间级别使用二进制读取器时都支持 Oracle TDE。要将 TDE 加密与一起使用 AWS DMS，请先确定存储 TDE 加密密钥和 TDE 密码的 Oracle 钱包位置。然后为您的 Oracle 源端点确定正确的 TDE 加密密钥和密码。

要识别并指定用于 TDE 加密的加密密钥和密码

1. 运行以下查询，在 Oracle 数据库主机上查找 Oracle 加密 Wallet。

```
SQL> SELECT WRL_PARAMETER FROM V$ENCRYPTION_WALLET;

WRL_PARAMETER
-----
/u01/oracle/product/12.2.0/dbhome_1/data/wallet/
```

在这里，/u01/oracle/product/12.2.0/dbhome\_1/data/wallet/ 是 Wallet 所在的位置。

2. 使用以下加密选项之一（具体取决于哪个加密选项返回此值）获取主密钥 ID。
  - a. 对于表级别或列级别加密，请运行以下查询。

```
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
```

在这里，AWGDC9g1Sk8Xv+3bVveiVSg 是主密钥 ID ( MKEYID )。如果获取了 MKEYID 的值，则可继续执行步骤 3。否则，请继续执行步骤 2.2。

### Note

尾随字符串的 'A' 字符 ( AAA... ) 不是值的一部分。

- b. 对于表空间级别加密，请运行以下查询。

```
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$tablespace t, x$kcbbek x
WHERE t.ts#=x.ts#)
WHERE name = 'TEST_ENCRYPT';

NAME                      MASTERKEYID_BASE64
-----
TEST_ENCRYPT               AWGDC9g1Sk8Xv+3bVveiVSg=
```

在这里，AWGDC9g1Sk8Xv+3bVveiVSg 是主密钥 ID ( TEST\_ENCRYPT )。如果步骤 2.1 和 2.2 都返回一个值，则它们始终相同。

尾随的 '=' 字符不是值的一部分。

3. 在命令行中，列出源 Oracle 数据库主机上的加密 Wallet 条目。

```
$ 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
```

找到包含您在步骤 2 ( AWGDC9g1Sk8Xv+3bVveiVSg ) 中找到的包含主密钥 ID 的条目。此条目是 TDE 加密密钥名称。

4. 查看您在上一步中找到的条目的详细信息。

```
$ 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==
```

输入 Wallet 密码以查看结果。

在这里， '=' 右边的值是 TDE 密码。

5. 通过设置额外连接属性 `securityDbEncryptionName` 为 Oracle 源端点指定 TDE 加密密钥名称。

```
securityDbEncryptionName=ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9g1Sk8Xv
+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

6. 在控制台上提供此密钥的关联 TDE 密码，将其作为 Oracle 源的密码值的一部分。按以下顺序设置以逗号分隔的密码值的格式，以 TDE 密码值结尾。

```
Oracle_db_password,ASM_Password,AEMAASAASGYs0phWHfNt9J5mEMkkegGFid4LLfQszDojgDzbfoYDEACv0x3
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

无论 Oracle 数据库配置如何，都按此顺序指定密码值。例如，如果您使用的是 TDE，但 Oracle 数据库未使用 ASM，请按照以下逗号分隔顺序指定密码值。

```
Oracle_db_password,,AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3pJC+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

如果您指定的 TDE 凭证不正确，则 AWS DMS 迁移任务不会失败。但是，该任务也不会读取持续复制更改或将其应用到目标数据库。启动任务后，在控制台迁移任务页面上监控表统计数据，确保已复制更改。

如果数据库管理员在运行任务时更改 Oracle 数据库的 TDE 凭证值，任务会失败。错误消息中包含新的 TDE 加密密钥名称。要指定新值并重新启动任务，请使用前面的步骤。

### ⚠ Important

您无法操作在 Oracle Automatic Storage Management ( ASM ) 所在位置创建的 TDE Wallet，因为操作系统级别的命令（例如 cp、mv、orapki 和 mkstore）会损坏存储在 ASM 所在位置的 Wallet 文件。此限制仅适用于只存储在 ASM 所在位置的 TDE Wallet 文件，但不适用于存储在本地操作系统目录中的 TDE Wallet 文件。

要使用操作系统级别的命令操作存储在 ASM 中的 TDE Wallet，请创建本地密钥库，并按如下方式将 ASM 密钥库合并到本地密钥库中：

1. 创建本地密钥库。

```
ADMINISTER KEY MANAGEMENT create keystore file system wallet location identified by wallet password;
```

2. 将 ASM 密钥库合并到本地密钥库中。

```
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;
```

然后，要列出加密 Wallet 条目和 TDE 密码，请对本地密钥库运行步骤 3 和 4。

## 使用 Oracle 作为源时支持的压缩方法 AWS DMS

在下表中，您可以找到使用 Oracle 源数据库时 AWS DMS 支持的压缩方法。如表所示，压缩支持取决于您的 Oracle 数据库版本以及 DMS 是否配置为使用 Oracle LogMiner 访问重做日志。

版本	基本	OLTP	HCC ( 来自 Oracle 11g R2 或更高版本 )	其他
Oracle 10	否	不适用	不适用	否
甲骨文 11 或更新版本 — 甲骨文 LogMiner	是	是	是	是 — Oracle 支持的任何压缩方法 LogMiner。
Oracle 11 或更高版本 – Binary Reader	是	是	是 – 有关更多信息，请参阅以下注释。	是

### Note

当 Oracle 源终端节点配置为使用 Binary Reader 时，只有满载任务才支持 HCC 压缩方法的查询低级别。

## 使用 Oracle 作为源来复制嵌套表 AWS DMS

AWS DMS 支持复制包含嵌套表或已定义类型的列的 Oracle 表。要启用此功能，请将以下额外连接属性设置添加到 Oracle 源端点。

```
allowSelectNestedTables=true;
```

AWS DMS 从 Oracle 嵌套表中创建目标表，作为目标上的常规父表和子表，没有唯一约束。要访问目标上的正确数据，请联接父表和子表。为此，首先在目标子表中的 NESTED\_TABLE\_ID 列上手动创建非唯一索引。然后，您可以将 ON 联接子句中的 NESTED\_TABLE\_ID 列与子表名称对应的父列一起使用。此外，通过更新或删除目标子表数据时，创建这样的索引可以提高性能 AWS DMS。有关示例，请参阅[目标上的父表和子表联接示例](#)。

我们建议您将任务配置为在满载完成后停止。然后，为目标上的所有复制子表创建这些非唯一索引，然后恢复任务。

如果将捕获的嵌套表添加到现有的父表中（捕获或未捕获），请正确 AWS DMS 处理该表。但是，不会创建相应目标表的非唯一索引。在此情况下，如果目标子表变得非常大，性能可能会受到影响。在这种情况下，我们建议您停止任务，创建索引，然后再恢复任务。

在嵌套表复制到目标后，让 DBA 运行父表和相应子表上的联接以平展数据。

将 Oracle 嵌套表复制为源的先决条件

确保复制所有复制的嵌套表的父表。在表映射中同时包括父表（包含嵌套表的表）和子表（即嵌套）。AWS DMS

支持的 Oracle 嵌套表类型作为源

AWS DMS 支持以下 Oracle 嵌套表类型作为源：

- 数据类型
- 用户定义的对象

对 Oracle 嵌套表作为源的 AWS DMS 支持的限制

AWS DMS 在支持 Oracle 嵌套表作为源方面存在以下限制：

- AWS DMS 仅支持一层表嵌套。
- AWS DMS 表映射不会检查父表和子表或表是否都被选中进行复制。也就是说，可以选择没有子表的父表或没有父表的子表。

AWS DMS 如何将 Oracle 嵌套表复制为源

AWS DMS 将父表和嵌套表复制到目标表，如下所示：

- AWS DMS 创建与源表相同的父表。然后，它将父表中的嵌套列定义为 RAW(16)，并在其 NESTED\_TABLE\_ID 列中包含对父表的嵌套表的引用。
- AWS DMS 创建与嵌套源相同的子表，但还有一个名为的附加列 NESTED\_TABLE\_ID。此列的类型和值与相应的父嵌套列相同，含义也相同。

## 目标上的父表和子表联接示例

要平展父表，请运行父表和子表之间的联接，如以下示例所示：

### 1. 创建 Type 表。

```
CREATE OR REPLACE TYPE NESTED_TEST_T AS TABLE OF VARCHAR(50);
```

### 2. 使用前面定义的列（类型为 NESTED\_TEST\_T）列创建父表。

```
CREATE TABLE NESTED_PARENT_TEST (ID NUMBER(10,0) PRIMARY KEY, NAME NESTED_TEST_T)  
  NESTED TABLE NAME STORE AS NAME_KEY;
```

### 3. 通过对其 CHILD.NESTED\_TABLE\_ID 与 PARENT.NAME 匹配的 NAME\_KEY 子表使用联接来平展表 NESTED\_PARENT\_TEST。

```
SELECT ... FROM NESTED_PARENT_TEST PARENT, NAME_KEY CHILD WHERE CHILD.NESTED_  
TABLE_ID = PARENT.NAME;
```

## 使用 Oracle 作为源时在 Oracle ASM 上存储 REDO AWS DMS

对于 REDO 生成量较高的 Oracle 源，在 Oracle ASM 上存储 REDO 可以提高性能，尤其是在 RAC 配置中，因为您可以将 DMS 配置为跨所有 ASM 节点分配 ASM REDO 读取。

要使用此配置，请使用 `asmServer` 连接属性。例如，以下连接字符串将跨 3 个 ASM 节点分配 DMS REDO 读取：

```
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)))
```

使用 NFS 存储 Oracle REDO 时，请务必确保应用了适用的 DNFS ( Direct NFS ) 客户端补丁，特别是解决 Oracle 错误 25224242 的任何补丁。有关更多信息，请查阅以下有关 Direct NFS Client 相关补丁的 Oracle 出版物 [Direct NFS Client 的推荐补丁](#)。

此外，为了提高 NFS 读取性能，我们建议您增加 NFS 卷的 `fstab` 中的 `rsize` 和 `wsizes` 的值，如下示例所示。



```
NAS_name_here:/ora_DATA1_archive /u09/oradata/DATA1 nfs
rw,bg,hard,nointr,tcp,nfsvers=3,_netdev,
timeo=600,rsize=262144,wsiz=262144
```

另外，按如下方式调整 `tcp-max-xfer-size` 值：

```
vserver nfs modify -vserver vserver -tcp-max-xfer-size 262144
```

## 使用 Oracle 作为来源时的终端节点设置 AWS DMS

您可以使用端点设置来配置 Oracle 源数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建源端点时指定设置，或者使用中带有 `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` [AWS CLI](#) 命令来指定设置。

下表显示了将 Oracle 用作源时可以使用的端点设置。

名称	描述
AccessAlternateDirectly	<p>将此属性设置为 <code>false</code>，以便使用 Binary Reader 捕获作为源的 Amazon RDS for Oracle 的更改数据。这将指示 DMS 实例不要使用直接文件访问通过任何指定的路径前缀替换来访问重做日志。有关更多信息，请参阅 <a href="#">将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS</a>。</p> <p>默认值：<code>true</code></p> <p>有效值：<code>true/false</code></p> <p>例如：<code>--oracle-settings '{"AccessAlternateDirectly": false}'</code></p>
AdditionalArchivedLogDestId	<p>在主/备用设置中将此属性与 <code>ArchivedLogDestId</code> 一起设置。当您使用 Oracle Data Guard 数据库作为源时，此属性在切换中非常有用。在这种情况下，AWS DMS 需要知道从哪个目的地获取存档重做日志以读取更改。这是因为之前的主实例现在是切换后的备用实例。</p> <p>尽管 AWS DMS 支持使用 Oracle <code>RESETLOGS</code> 选项打开数据库，但 <code>RESETLOGS</code> 除非必要，否则切勿使用。有关 <code>RESETLOGS</code> 的其他信息，</p>

名称	描述
	<p>请参阅《Oracle® 数据库备份和还原用户指南》中的 <a href="#">RMAN 数据修复概念</a>。</p> <p>有效值：存档目标 ID</p> <p>例如：<code>--oracle-settings '{"AdditionalArchivedLogDestId": 2}'</code></p>
AddSupplementalLogging	<p>设置此属性可设置 Oracle 数据库的表级别补充日志记录。根据表元数据的不同，此属性可在为迁移任务选择的所有表上启用以下操作之一：</p> <ul style="list-style-type: none"> <li>• 主键列补充日志记录</li> <li>• 唯一键列补充日志记录</li> <li>• 所有列补充日志记录</li> </ul> <p>默认值：false</p> <p>有效值：true/false</p> <p>例如：<code>--oracle-settings '{"AddSupplementalLogging": false}'</code></p> <div data-bbox="461 1184 1508 1402" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>如果您使用此选项，则仍需启用前面讨论过的数据库级别补充日志记录。</p> </div>
AllowSelectNestedTables	<p>将此属性设置为 true 可复制包含作为嵌套表或已定义类型的列的 Oracle 表。有关更多信息，请参阅 <a href="#">使用 Oracle 作为源来复制嵌套表 AWS DMS</a>。</p> <p>默认值：false</p> <p>有效值：true/false</p> <p>例如：<code>--oracle-settings '{"AllowSelectNestedTables": true}'</code></p>

名称	描述
ArchivedLogDestId	<p>指定已存档重做日志的目标的 ID。该值应与 v\$archived_log 视图的 dest_id 列中的数相同。如果您使用其他重做日志目标，建议您使用 AdditionalArchivedLogDestId 属性来指定其他目标 ID。这样做将确保从一开始便访问正确的日志，从而提高性能。</p> <p>默认值：1</p> <p>有效值：数字</p> <p>例如：--oracle-settings '{"ArchivedLogDestId": 1}'</p>
ArchivedLogsOnly	<p>当此字段设置为 Y 时，AWS DMS 仅访问已存档的重做日志。如果存档的重做日志仅存储在 Oracle ASM 上，则需要向 AWS DMS 用户帐户授予 ASM 权限。</p> <p>默认值：N</p> <p>有效值：Y/N</p> <p>例如：--oracle-settings '{"ArchivedLogsOnly": Y}'</p>
asmUsePLSQLArray (仅限 ECA)	<p>使用捕获源更改时，请使用此额外连接属性 (ECA) BinaryReader。此设置允许 DMS 在 ASM 级别为每个读取线程缓冲 50 次读取，同时使用 parallelASMReadThread 属性控制线程的数量。设置此属性时，AWS DMS 二进制读取器使用匿名的 PL/SQL 块来捕获重做数据并将其作为大型缓冲区发送回复制实例。这减少了往返源的次数。这可以显著提高源捕获性能，但确实会导致 ASM 实例上的 PGA 内存消耗增加。如果内存目标不足，可能会出现稳定性问题。您可以使用以下公式，估算单个 DMS 任务的 ASM 实例 PGA 内存使用总量：  <math display="block">\text{number\_of\_redo\_threads} * \text{parallelASMReadThreads} * 7 \text{ MB}</math></p> <p>默认值：false</p> <p>有效值：true/false</p> <p>ECA 示例：asmUsePLSQLArray=true;</p>

名称	描述
ConvertTimestampWithZoneToUTC	<p>将此属性设置为 <code>true</code>，以便将“TIMESTAMP WITH TIME ZONE”和“TIMESTAMP WITH LOCAL TIME ZONE”列的时间戳值转换为 UTC。默认情况下，此属性的值为“<code>false</code>”，数据将使用源数据库时区进行复制。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true/false</code></p> <p>例如：<code>--oracle-settings '{"ConvertTimestampWithZoneToUTC": true}'</code></p>
EnableHomogenousPartitionOps	<p>将此属性设置为 <code>true</code>，以便为 Oracle 同构迁移启用 Oracle 分区和子分区 DDL 操作的复制。</p> <p>请注意，此功能和增强功能已在 3.4.7 AWS DMS 版本中引入。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true/false</code></p> <p>例如：<code>--oracle-settings '{"EnableHomogenousPartitionOps": true}'</code></p>
EnableHomogenousTablespace	<p>将此属性设置为启用同类表空间复制，并在目标上的相同表空间下创建现有的表或索引。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true/false</code></p> <p>例如：<code>--oracle-settings '{"EnableHomogenousTablespace": true}'</code></p>

名称	描述
EscapeCharacter	<p>将此属性设置为转义字符。此转义字符允许您在表映射表达式中将单个通配符用作一个普通字符。有关更多信息，请参阅 <a href="#">表映射中的通配符</a>。</p> <p>默认值：Null</p> <p>有效值：除通配符之外的任何字符</p> <p>例如：<code>--oracle-settings '{"EscapeCharacter": "#"}'</code></p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>您只能将 <code>escapeCharacter</code> 用于表名。它不会对架构名或列名中的字符进行转义。</p> </div>
ExposeViews	<p>使用此属性可从视图中提取一次数据；您无法将其用于持续复制。从视图中提取数据时，此视图将显示为目标架构上的表。</p> <p>默认值：false</p> <p>有效值：true/false</p> <p>例如：<code>--oracle-settings '{"ExposeViews": true}'</code></p>
ExtraArchivedLogDestIds	<p>指定一个或多个已存档重做日志的多个目标的 ID。这些 ID 是 <code>v\$archive_d_log</code> 视图中 <code>dest_id</code> 列的值。在设置或设置中将此 <code>primary-to-single</code> 设置与 <code>ArchivedLogDestId</code> 额外的连接属性一起 <code>primary-to-multiple-standby</code> 使用。</p> <p>当您使用 Oracle Data Guard 数据库作为源时，此设置在切换中非常有用。在这种情况下，AWS DMS 需要有关从哪个目的地获取存档重做日志以读取更改的信息。AWS DMS 需要这个，因为在切换之后，先前的主实例是备用实例。</p> <p>有效值：存档目标 ID</p> <p>例如：<code>--oracle-settings '{"ExtraArchivedLogDestIds": 1}'</code></p>

名称	描述
FailTasksOnLobTruncation	<p>当设置为 <code>true</code> 时，如果 LOB 列的实际大小大于指定的 <code>LobMaxSize</code>，此属性将导致任务失败。</p> <p>如果任务设置为受限 LOB 模式并且此选项设置为 <code>true</code>，任务将失败，而不是截断 LOB 数据。</p> <p>默认值：<code>false</code></p> <p>有效值：布尔值</p> <p>例如：<code>--oracle-settings '{"FailTasksOnLobTruncation": true}'</code></p>
filterTransactionsOfUser (仅限 ECA)	<p>使用此额外连接属性 (ECA) 允许 DMS 在使用时从 Oracle 复制数据时忽略来自指定用户的事务。LogMiner 您可以传递以逗号分隔的用户名值，但这些值必须全部使用大写字母。</p> <p>ECA 示例：<code>filterTransactionsOfUser= <i>USERNAME</i>;</code></p>
NumberDataTypeScale	<p>指定数字的小数位。可以选择最大为 38 的小数位，也可以选择 -1 (对于 FLOAT) 或 -2 (VARCHAR)。默认情况下，NUMBER 数据类型将转换为精度 38，小数位数 10。</p> <p>默认值：10</p> <p>有效值：-2 到 38 (对于 VARCHAR 为 -2，对于 FLOAT 为 -1)</p> <p>例如：<code>--oracle-settings '{"NumberDataTypeScale": 12}'</code></p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>选择精度小数位组合，即 -1 (FLOAT) 或 -2 (VARCHAR)。DMS 支持 Oracle 支持的任何精度小数位组合。如果精度大于等于 39，请选择 -2 (VARCHAR)。Oracle 数据库的 <code>NumberDataTypeScale</code> 设置仅用于 NUMBER 数据类型 (没有明确的精度和小数位定义)。</p> </div>

名称	描述
OpenTransactionWindow	<p>提供为仅 CDC 的任务检查任何未结事务的时间范围（以分钟为单位）。</p> <div data-bbox="461 302 1507 667" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>当您设置OpenTransactionWindow 为 1 或更高时，DMS 使用将 SCN 值SCN_TO_TIMESTAMP 转换为时间戳值。由于 Oracle 数据库的限制，如果您指定的 SCN 过早作为 CDC 起点，则 SCN_TO_TIMESTAMP 将失败并ORA-08181 显示错误，并且您无法启动仅限 CDC 的任务。</p> </div> <p>默认值：0</p> <p>有效值：从 0 到 240 的整数</p> <p>例如：openTransactionWindow=15;</p>
OraclePathPrefix	<p>将此字符串属性设置为必需值，以便使用 Binary Reader 捕获作为源的 Amazon RDS for Oracle 的更改数据。此值指定用于访问重做日志的默认 Oracle root。有关更多信息，请参阅 <a href="#">将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS</a>。</p> <p>默认值：none</p> <p>有效值：/rdsdbdata/db/ORCL_A/</p> <p>例如：<code>--oracle-settings '{"OraclePathPrefix": " /rdsdbdata/db/ORCL_A/ "'</code></p>

名称	描述
ParallelASMReadThreads	<p>设置此属性可更改 DMS 配置通过 Oracle Automatic Storage Management (ASM) 执行更改数据捕获 (CDC) 所用的线程数。您可以指定一个介于 2 (默认值) 和 8 (最大值) 之间的整数值。将此属性与 ReadAheadBlocks 属性一起使用。有关更多信息, 请参阅 <a href="#">将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS</a>。</p> <p>默认值: 2</p> <p>有效值: 从 2 到 8 的整数</p> <p>例如: <code>--oracle-settings '{"ParallelASMReadThreads": 6;}'</code></p>
ReadAheadBlocks	<p>设置此属性可更改 DMS 配置通过 Oracle Automatic Storage Management (ASM) 和非 ASM NAS 存储执行更改数据捕获 (CDC) 所用的预读数据块数。您可以指定一个介于 1000 (默认值) 和 200000 (最大值) 之间的整数值。将此属性与 ParallelASMReadThreads 属性一起使用。有关更多信息, 请参阅 <a href="#">将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS</a>。</p> <p>默认值: 1000</p> <p>有效值: 从 1000 到 200000 的整数</p> <p>例如: <code>--oracle-settings '{"ReadAheadBlocks": 150000}'</code></p>
ReadTableSpaceName	<p>当设置为 true 时, 此属性支持表空间复制。</p> <p>默认值: false</p> <p>有效值: 布尔值</p> <p>例如: <code>--oracle-settings '{"ReadTableSpaceName": true}'</code></p>



名称	描述
ReplacePathPrefix	<p>将此属性设置为 true，以便使用 Binary Reader 捕获作为源的 Amazon RDS for Oracle 的更改数据。此设置指示 DMS 实例使用指定的 UsePathPrefix 设置替换默认的 Oracle root，以访问重做日志。有关更多信息，请参阅 <a href="#">将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS</a>。</p> <p>默认值：false</p> <p>有效值：true/false</p> <p>例如：--oracle-settings '{"ReplacePathPrefix": true}'</p>
RetryInterval	<p>指定系统在重新发送查询之前等待的秒数。</p> <p>默认值：5</p> <p>有效值：从 1 开始的数字</p> <p>例如：--oracle-settings '{"RetryInterval": 6}'</p>
SecurityDbEncryptionName	<p>指定一个密钥的名称，该密钥用于对 Oracle 源数据库中的列和表空间进行透明数据加密 (TDE)。有关在 Oracle 源终端节点上设置此属性及其关联密码的更多信息，请参阅 <a href="#">使用 Oracle 作为数据源的支持的加密方法 AWS DMS</a>。</p> <p>默认值：""</p> <p>有效值：字符串</p> <p>例如：--oracle-settings '{"SecurityDbEncryptionName": "ORACLE.SECURITY.DB.ENCRYPTION.Adg8m2dhkU/0v/m5QUaaNJEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}'</p>

名称	描述
SpatialSdo2GeoJsonFunctionName	<p>对于迁移到 PostgreSQL 目标的 Oracle 12.1 版或更早版本的源，请使用此属性将 SDO_GEOMETRY 格式转换为 GEOJSON 格式。</p> <p>默认情况下，AWS DMS 调用SDO2GEOJSON 自定义函数，该函数必须存在且 AWS DMS 用户可以访问。或者，您也可以自行创建模仿 SDOGEOJSON 操作的自定义函数，并设置 SpatialSdo2GeoJsonFunctionName 以改为调用它。</p> <p>默认值：SDO2GEOJSON</p> <p>有效值：字符串</p> <p>例如：<code>--oracle-settings '{"SpatialSdo2GeoJsonFunctionName": "myCustomSDO2GEOJSONFunction"}'</code></p>
StandbyDelayTime	<p>使用此属性指定备用同步延迟时间 (以分钟为单位)。如果源是 Active Data Guard 备用数据库，请使用此属性指定主数据库和备用数据库之间的时间滞后。</p> <p>在中 AWS DMS，您可以创建一个 Oracle CDC 任务，该任务使用 Active Data Guard 备用实例作为复制正在进行的更改的来源。这样一来，便无需连接到可能处于生产状态的活动数据库。</p> <p>默认值：0</p> <p>有效值：数字</p> <p>例如：<code>--oracle-settings '{"StandbyDelayTime": 1}'</code></p> <p>注意：使用 DMS 3.4.6、3.4.7 及更高版本时，可以选用此连接设置。在最新版本的 DMS 3.4.6 和 3.4.7 中，<i>dms_user</i> 应具有 V_\$DATAGUARD_STATS 的 select 权限，这样才能允许 DMS 计算待机延迟时间。</p>

名称	描述
UseAlternateFolderForOnline	<p>将此属性设置为 true，以便使用 Binary Reader 捕获作为源的 Amazon RDS for Oracle 的更改数据。这将指示 DMS 实例使用任何指定的前缀替换来访问所有在线重做日志。有关更多信息，请参阅 <a href="#">将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS。</a></p> <p>默认值：false</p> <p>有效值：true/false</p> <p>例如：--oracle-settings '{"UseAlternateFolderForOnline": true}'</p>
UseBfile	<p>将此属性设置为 Y，以便使用 Binary Reader 实用程序捕获更改数据。将 UseLogminerReader 设置为 N 可将此属性设置为 Y。要在将 Amazon RDS for Oracle 作为源时使用 Binary Reader，请设置其他属性。有关此设置以及使用 Oracle Automatic Storage Management (ASM) 的更多信息，请参阅<a href="#">在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器。</a></p> <p>注意：将此值设置为额外连接属性 (ECA) 时，有效值为“Y”和“N”。将此值设置为端点设置时，有效值为 true 和 false。</p> <p>默认值：N</p> <p>有效值：Y/N (将此值设置为 ECA 时)；true/false (将此值设置为端点设置时)。</p> <p>例如：--oracle-settings '{"UseBfile": Y}'</p>

名称	描述
UseLogminerReader	<p>将此属性设置为 Y 以使用 LogMiner 实用程序 ( 默认 ) 捕获变更数据。如果希望 AWS DMS 将重做日志作为二进制文件进行访问, 请将此选项设置为 N。将此选项设置为 N 时, 还要添加设置 useBfile=Y。有关此设置和使用 Oracle Automatic Storage Management ( ASM ) 的更多信息, 请参阅 <a href="#">在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器</a>。</p> <p>注意: 将此值设置为额外连接属性 ( ECA ) 时, 有效值为“Y”和“N”。将此值设置为端点设置时, 有效值为 true 和 false。</p> <p>默认值: Y</p> <p>有效值: Y/N ( 将此值设置为 ECA 时 ); true/false ( 将此值设置为端点设置时 )。</p> <p>例如: <code>--oracle-settings '{"UseLogminerReader": Y}'</code></p>
UsePathPrefix	<p>将此字符串属性设置为必需值, 以便使用 Binary Reader 捕获作为源的 Amazon RDS for Oracle 的更改数据。此值指定用于替换访问重做日志的默认 Oracle root 的路径前缀。有关更多信息, 请参阅 <a href="#">将 CDC 任务配置为将二进制读取器与 RDS for Oracle 源一起使用 AWS DMS</a>。</p> <p>默认值: none</p> <p>有效值: /rdsdbdata/log/</p> <p>例如: <code>--oracle-settings '{"UsePathPrefix": " /rdsdbdata/log/ "'}</code></p>

## Oracle 的源数据类型

的 Oracle 终端节点 AWS DMS 支持大多数 Oracle 数据类型。下表显示了使用时支持的 Oracle 源数据类型 AWS DMS 以及与 AWS DMS 数据类型的默认映射。

**Note**

除了 LONG 和 LONG RAW 数据类型之外，从 Oracle 源复制到 Oracle 目标（同构复制）时，所有源数据类型和目标数据类型都相同。但是 LONG 数据类型将映射到 CLOB，LONG RAW 数据类型将映射到 BLOB。

有关如何查看目标中映射的数据类型的信息，请参阅有关所使用的目标终端节点的部分。

有关 AWS DMS 数据类型的更多信息，请参见[AWS Database Migration Service 的数据类型](#)。

Oracle 数据类型	AWS DMS 数据类型
BINARY_FLOAT	REAL4
BINARY_DOUBLE	REAL8
BINARY	BYTES
FLOAT (P)	如果精度小于或等于 24，请使用 REAL4。 如果精度大于 24，请使用 REAL8。
NUMBER (P,S)	当小数位数大于 0 时，请使用 NUMERIC。  当小数位数为 0 时： <ul style="list-style-type: none"> <li>• 如果精度小于或等于 2，请使用 INT1。</li> <li>• 如果精度大于 2 且小于或等于 4，请使用 INT2。</li> <li>• 如果精度大于 4 且小于或等于 9，请使用 INT4。</li> <li>• 如果精度大于 9，请使用 NUMERIC。</li> <li>• 如果精度大于或等于小数位数，请使用 NUMERIC。</li> </ul> 当小数位数小于 0 时，请使用 REAL8。
DATE	DATETIME
INTERVAL_YEAR TO MONTH	STRING (带 interval year_to_month 指示)

Oracle 数据类型	AWS DMS 数据类型
INTERVAL_DAY TO SECOND	STRING (带 interval day_to_second 指示)
TIMESTAMP	DATETIME
TIMESTAMP (有时区)	STRING (带 timestamp_with_timezone 指示)
带本地时区的 TIMESTAMP	STRING (带 timestamp_with_local_timezone 指示)
CHAR	string
VARCHAR2	string
NCHAR	WSTRING
NVARCHAR2	WSTRING
RAW	BYTES
REAL	REAL8
BLOB	BLOB  要将此数据类型与一起使用 AWS DMS，必须允许将 BLOB 数据类型用于特定任务。AWS DMS 仅在包含主键的表中支持 BLOB 数据类型。
CLOB	CLOB  要将此数据类型与一起使用 AWS DMS，必须为特定任务启用 CLOB 数据类型。在 CDC 期间，仅在包含主键的表中 AWS DMS 支持 CLOB 数据类型。
NCLOB	NCLOB  要将此数据类型与一起使用 AWS DMS，必须允许将 NCLOB 数据类型用于特定任务。在 CDC 期间，仅在包含主键的表中 AWS DMS 支持 NCLOB 数据类型。

Oracle 数据类型	AWS DMS 数据类型
LONG	<p data-bbox="544 226 634 258">CLOB</p> <p data-bbox="544 306 1498 386">批量优化应用模式 ( TurboStream CDC 模式 ) 不支持 LONG 数据类型。</p> <p data-bbox="544 434 1487 514">要将此数据类型与一起使用 AWS DMS , 请启用 LOB 来执行特定任务。</p> <p data-bbox="544 562 1487 642">在 CDC 或满载期间 , 仅在具有主键的表中 AWS DMS 支持 LOB 数据类型。</p> <p data-bbox="544 690 1498 911">此外 , AWS DMS 不支持用于加载长列的完整 LOB 模式。但您可以使用受限的 LOB 模式将 LONG 列迁移到 Oracle 目标。在受限 LOB 模式下 , 将您设置为长度超过 64 KB 的长列的所有数据 AWS DMS 截断为 64 KB。有关 LOB 支持的更多信息 AWS DMS , 请参阅 <a href="#">在任务中为源数据库设置 LOB 支持 AWS DMS</a></p>
LONG RAW	<p data-bbox="544 961 634 993">BLOB</p> <p data-bbox="544 1041 1479 1121">批量优化应用模式 ( TurboStream CDC 模式 ) 不支持 LONG RAW 数据类型。</p> <p data-bbox="544 1169 1487 1249">要将此数据类型与一起使用 AWS DMS , 请启用 LOB 来执行特定任务。</p> <p data-bbox="544 1297 1487 1377">在 CDC 或满载期间 , 仅在具有主键的表中 AWS DMS 支持 LOB 数据类型。</p> <p data-bbox="544 1425 1498 1692">此外 , AWS DMS 不支持用于加载 LONG RAW 列的完整 LOB 模式。但您可以使用受限 LOB 模式将 LONG RAW 列迁移到 Oracle 目标。在受限 LOB 模式下 , 如果您设置为长度超过 64 KB 的 LONG RAW 列 , 则会将所有数据 AWS DMS 截断为 64 KB。有关 LOB 支持的更多信息 AWS DMS , 请参阅 <a href="#">在任务中为源数据库设置 LOB 支持 AWS DMS</a></p>
XMLTYPE	CLOB

Oracle 数据类型	AWS DMS 数据类型
SDO_GEOMETRY	BLOB ( 当 Oracle 迁移到 Oracle 时 ) CLOB ( 当 Oracle 迁移到 PostgreSQL 时 )

不支持且无法复制作为源的包含以下数据类型的列的 Oracle 表。复制带这些数据类型的列会生成 null 列。

- BFILE
- ROWID
- REF
- UROWID
- 用户定义的数据类型
- ANYDATA
- VARRAY

#### Note

不支持虚拟列。

## 迁移 Oracle 空间数据类型

空间数据 可识别空间中对象或位置的几何信息。在 Oracle 数据库中，空间对象的几何描述存储在 SDO\_GEOMETRY 类型的对象中。在此对象中，几何描述存储在用户定义表的单列中的单行中。

AWS DMS 支持将 Oracle 类型 SDO\_GEOMETRY 从 Oracle 源迁移到 Oracle 或 PostgreSQL 目标。

使用迁移 Oracle 空间数据类型时 AWS DMS，请注意以下注意事项：

- 迁移到 Oracle 目标时，请确保手动传输包含类型信息的 USER\_SDO\_GEOM\_METADATA 条目。
- 从 Oracle 源端点迁移到 PostgreSQL 目标端点时 AWS DMS，会创建目标列。这些列具有默认的几何和地理类型信息，其 2D 维度和空间参考标识符 (SRID) 等于零 (0)。例如，GEOMETRY, 2, 0。



- 对于迁移到 PostgreSQL 目标的 Oracle 12.1 版或更早版本的源，请使用 `SDO2GEOJSON` 函数或 `spatialSdo2GeoJsonFunctionName` 额外的连接属性，将 `SDO_GEOMETRY` 对象转换为 `GEOJSON` 格式。有关更多信息，请参阅 [使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。
- AWS DMS 仅在完整 LOB 模式下支持 Oracle 空间列迁移。AWS DMS 不支持受限 LOB 或内联 LOB 模式。有关 LOB 模式的更多信息，请参阅 [在任务中为源数据库设置 LOB 支持 AWS DMS](#)。
- 由于 AWS DMS 仅支持用于迁移 Oracle Spatial Columns 的完整 LOB 模式，因此列的表需要主键和唯一键。如果表没有主键和唯一键，在迁移时会跳过此表。

## 使用微软 SQL Server 数据库作为来源 AWS DMS

使用迁移一个或多个 Microsoft SQL Server 数据库中的数据 AWS DMS。使用 SQL Server 数据库作为源，您可以将数据迁移到另一个 SQL Server 数据库或其他 AWS DMS 支持的数据库中。

有关 AWS DMS 支持作为源的 SQL Server 版本的信息，请参见 [的来源 AWS DMS](#)。

源 SQL Server 数据库可安装在您网络中的任意计算机上。在与 AWS DMS 一起使用时，需要 SQL Server 账户，该账户对源数据库具有与选定任务类型相对应的访问权限。此账户必须具有 `view definition` 和 `view server state` 权限。您可以使用以下命令添加此权限：

```
grant view definition to [user]
grant view server state to [user]
```

AWS DMS 支持从 SQL Server 的命名实例迁移数据。当您创建源终端节点时，您可以在服务器名称中使用以下表示法。

```
IPAddress\InstanceName
```

例如，以下是正确的源终端节点服务器名称。在这里，名称的第一部分是服务器的 IP 地址，第二部分是 SQL Server 实例名称 (本例中为 `SQLTest`)。

```
10.0.0.25\SQLTest
```

另外，获取您的命名 SQL Server 实例监听的端口号，然后使用它来配置您的 AWS DMS 源端点。

**Note**

端口 1433 是 Microsoft SQL Server 的默认端口。但是，也经常使用每次启动 SQL Server 时更改的动态端口，以及用于通过防火墙连接到 SQL Server 的特定静态端口号。因此，在创建 AWS DMS 源端点时，您想知道命名的 SQL Server 实例的实际端口号。

可以使用 SSL 加密 SQL Server 终端节点与复制实例之间的连接。有关将 SSL 用于 SQL Server 终端节点的更多信息，请参阅[将 SSL 与 AWS Database Migration Service](#)。

有关使用 SQL Server 源数据库和的更多详细信息 AWS DMS，请参阅以下内容。

**主题**

- [使用 SQL Server 作为源代码的限制 AWS DMS](#)
- [仅完全加载任务的权限](#)
- [在 SQL Server 源中使用持续复制 \( CDC \) 的先决条件](#)
- [捕获本地或 Amazon EC2 上的自管理 SQL Server 的数据更改](#)
- [在云 SQL Server 数据库实例上设置持续复制](#)
- [使用适用于 SQL Server 的 Amazon RDS 作为来源时的推荐设置 AWS DMS](#)
- [支持的 SQL Server 压缩方法](#)
- [使用自行管理的 SQL Server AlwaysOn 可用性组](#)
- [使用 SQL Server 作为源时的安全要求 AWS Database Migration Service](#)
- [使用 SQL Server 作为源时的端点设置 AWS DMS](#)
- [SQL Server 的源数据类型](#)

**使用 SQL Server 作为源代码的限制 AWS DMS**

将 SQL Server 数据库作为 AWS DMS 源时，存在以下限制：

- 列的身份属性不迁移到目标数据库列。
- SQL Server 端点不支持使用带有稀疏列的表。
- 不支持 Windows 身份验证。
- 不复制对 SQL Server 中计算字段的更改。
- 不支持临时表。

- 不支持 SQL Server 分区切换。
- 使用 WRITETEXT 和 UPDATETEXT 实用程序时，AWS DMS 不会捕获应用于源数据库的事件。
- 不支持以下数据操作语言 ( DML ) 模式。

```
SELECT * INTO new_table FROM existing_table
```

- 在将 SQL Server 作为源时，不支持列级别加密。
- AWS DMS 不支持将 SQL Server 2008 或 SQL Server 2008 R2 作为来源进行服务器级审计。这是因为 SQL Server 2008 和 2008 R2 存在已知问题。例如，运行以下命令会 AWS DMS 导致失败。

```
USE [master]
GO
ALTER SERVER AUDIT [my_audit_test-20140710] WITH (STATE=on)
GO
```

- 使用 SQL Server 作为源时，在完整 LOB 模式下不支持几何列。应改用受限 LOB 模式，或将 InlineLobMaxSize 任务设置设为使用内联 LOB 模式。
- 在复制任务中使用 Microsoft SQL Server 源数据库时，如果删除该任务，则不会删除 SQL Server Replication Publisher 定义。Microsoft SQL Server 系统管理员必须从 Microsoft SQL Server 删除这些定义。
- 仅限满载的任务支持从架构绑定和 non-schema-bound 视图迁移数据。
- 不支持使用 sp\_rename 重命名表 ( 例如，sp\_rename 'Sales.SalesRegion', 'SalesReg; )
- 不支持使用 sp\_rename 重命名列 ( 例如，sp\_rename 'Sales.Sales.Region', 'RegID', 'COLUMN'; )
- AWS DMS 不支持更改处理以设置和取消设置列默认值 ( 使用带 ALTER TABLE 语 ALTER COLUMN SET DEFAULT 句的子句 )。
- AWS DMS 不支持通过更改处理来设置列可为空性 ( 使用带 ALTER TABLE 语 ALTER COLUMN [SET|DROP] NOT NULL 句的子句 )。
- 在 SQL Server 2012 和 SQL Server 2014 中，将 DMS 复制与可用性组一起使用时，无法将分发数据库放入可用性组。SQL 2016 支持将分发数据库放入可用性组中，但合并、双向或 peer-to-peer 复制拓扑中使用的分布数据库除外。
- 对于分区表，AWS DMS 不支持为每个分区设置不同的数据压缩设置。
- 将值插入 SQL Server 空间数据类型 ( GEOGRAPHY 和 GEOMETRY ) 时，可以忽略空间参考系统标识符 ( SRID ) 属性或指定其他数字。复制具有空间数据类型的表时，AWS DMS 将 SRID 替换为默认 SRID ( 几何为 0，地理位置为 4326 )。

- 如果您的数据库未配置为进行 MS-REPLICATION 或 MS-CDC，您仍然可以捕获没有主键的表，但只捕获 INSERT/DELETE DML 事件。将忽略 UPDATE 和 TRUNCATE TABLE 事件。
- 不支持列存储索引。
- 不支持内存优化表（使用内存中 OLTP）。
- 当复制主键由多个列组成的表时，不支持在完全加载期间更新主键列。
- 不支持延迟的持久性。
- 由于 RDS 执行备份的方式，readBackupOnly=Y 端点设置（额外连接属性）在 RDS for SQL Server 源实例上不起作用。
- EXCLUSIVE\_AUTOMATIC\_TRUNCATION 在 Amazon RDS SQL Server 源实例上不起作用，因为 RDS 用户无权运行 SQL Server 存储过程 sp\_repldone。
- AWS DMS 不捕获截断命令。
- AWS DMS 不支持从开启了加速数据库恢复 (ADR) 的数据库进行复制。
- AWS DMS 不支持在单个事务中捕获数据定义语言 (DDL) 和数据操作语言 (DML) 语句。
- AWS DMS 不支持复制数据层应用程序包 (DACPAC)。
- 当您将更改应用于目标数据库时，涉及主键或唯一索引并更新多个数据行的 UPDATE 语句可能会导致冲突。例如，当目标数据库通过 INSERT 和 DELETE 语句而不是单个 UPDATE 语句应用更新时，可能会发生这种情况。在批量优化应用模式中，该表可能会被忽略。在事务应用模式下，UPDATE 操作可能会导致违反约束。要避免出现此问题，请重新加载相关表。或者，在 Apply Exceptions 控制表 (dmslogs.awsdms\_apply\_exceptions) 中找到有问题的记录，然后在目标数据库中手动对其进行编辑。有关更多信息，请参阅[更改处理优化设置](#)。
- AWS DMS 不支持表和架构的复制，其中名称包含以下集合中的特殊字符。

```
\\ -- \n \" \b \r ' \t ;
```

- 不支持数据屏蔽。AWS DMS 在不进行屏蔽的情况下迁移被屏蔽的数据。
- AWS DMS 最多复制 32,767 个带主键的表，每个表最多可复制 1,000 个列。这是因为 AWS DMS 会为每个复制的表创建一个 SQL Server 复制文章，而 SQL Server 复制文章有这些限制。
- 使用更改数据捕获 (CDC) 时，必须将构成唯一索引的所有列定义为 NOT NULL。如果不满足此要求，将导致出现 SQL Server 系统错误 22838。

访问备份事务日志时适用以下限制：

- 不支持加密的备份。
- 不支持存储在 URL 或 Windows Azure 上的备份。

- AWS DMS 不支持直接处理来自备用共享文件夹的文件级别的事务日志备份。

## 仅完全加载任务的权限

要执行仅完全加载的任务，需要以下权限。请注意，这 AWS DMS 不会创建 `dms_user` 登录名。有关为 SQL Server 创建登录角色的信息，请参阅[使用 Microsoft SQL Server 创建数据库用户](#)。

```
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;
```

## 在 SQL Server 源中使用持续复制 ( CDC ) 的先决条件

您可以针对本地或 Amazon EC2 上的自管理 SQL Server 数据库、诸如 Amazon RDS 之类的云数据库或者 Microsoft Azure SQL 托管实例使用持续复制 ( 更改数据捕获 ( CDC ) )。

尤其在作为 AWS DMS 源的 SQL Server 数据库使用持续复制时，存在以下要求：

- 必须配置 SQL Server 进行完整备份，并且您必须在开始复制数据之前先进行一次备份。
- 恢复模式必须设置为 Bulk logged 或 Full。
- 不支持 SQL Server 备份到多个磁盘。如果将备份定义为将数据库备份写入不同磁盘上的多个文件，则 AWS DMS 无法读取数据，AWS DMS 任务将失败。
- 对于自管理 SQL Server 源，在您删除任务时，DMS CDC 任务中使用的源数据库的 SQL Server Replication Publisher 定义不会被删除。SQL Server 系统管理员必须从 SQL Server 删除自管理源的这些定义。
- 在 CDC 期间，AWS DMS 需要查找 SQL Server 事务日志备份才能读取更改。AWS DMS 不支持使用非本机格式的第三方备份软件创建的 SQL Server 事务日志备份。要支持采用本机格式并且使用第三方备份软件创建的事务日志备份，请将 `use3rdPartyBackupDevice=Y` 连接属性添加到源端点。
- 对于自管理 SQL Server 源，请注意，在新创建的表发布前，SQL Server 不会捕获对它们的更改。将表添加到 SQL Server 源时，会 AWS DMS 管理发布的创建。但是，该过程可能需要几分钟时间。不会将在该延迟期间对新创建的表执行的操作捕获或复制到目标中。

- AWS DMS 更改数据捕获需要在 SQL Server 中开启完整的事务日志记录。要在 SQL Server 中启用完整事务日志记录，请启用 MS-REPLICATION 或 CHANGE DATA CAPTURE ( CDC )。
- 在 MS CDC 捕获作业处理这些更改之前，SQL Server tlog 条目不会被标记为可重复使用。
- 内存优化表上不支持 CDC 操作。该限制适用于 SQL Server 2014 ( 在此版本中首次引入该功能 ) 及更高版本。
- AWS DMS 默认情况下，更改数据捕获需要将 Amazon EC2 或 On-Prem SQL 服务器上的分布数据库作为源。因此，请确保在为带有主键的表配置 MS 复制时已激活分发服务器。

## 捕获本地或 Amazon EC2 上的自管理 SQL Server 的数据更改

要从源 Microsoft SQL Server 数据库中捕获更改，请确保该数据库已配置为进行完整备份。将数据库配置为完全恢复模式或批量日志记录模式。

对于自行管理的 SQL Server 源，请 AWS DMS 使用以下内容：

### MS-Replication

用于捕获带主键的表的更改。您可以通过向源 SQL Server 实例上的 AWS DMS 端点用户授予系统管理员权限来自动进行配置。或者，您可以按照本节中的步骤准备源代码并使用对端点不具有 sysadmin 权限的 AWS DMS 用户。

### MS-CDC

用于捕获不带主键的表的更改。MS-CDC 必须在数据库级别上启用，并且为各个表分别启用。

设置 SQL Server 数据库进行持续复制 ( CDC ) 时，可以执行下列操作之一：

- 使用 sysadmin 角色设置持续复制。
- 将持续复制设置为不使用 sysadmin 角色。

### 在自管理 SQL Server 上设置持续复制

本部分包含有关使用或不使用 sysadmin 角色在自管理 SQL Server 上设置持续复制的信息。

### 主题

- [在自管理 SQL Server 上设置持续复制：使用 sysadmin 角色](#)
- [在独立 SQL Server 上设置持续复制：不使用 sysadmin 角色](#)

## 在自管理 SQL Server 上设置持续复制：使用 sysadmin 角色

AWS DMS SQL Server 的持续复制对带主键的表使用本机 SQL Server 复制，对没有主键的表使用更改数据捕获 (CDC)。

在设置持续复制之前，请参阅[在 SQL Server 源中使用持续复制 \(CDC\) 的先决条件](#)。

对于带有主键的表，通常 AWS DMS 可以在源上配置所需的工件。但对于自管理 SQL Server 源数据库实例，必须先手动配置 SQL Server 分发。执行此操作后，具有 sysadmin 权限的 AWS DMS 源用户可以自动为具有主键的表创建发布。

要检查是否已配置分发，请运行以下命令。

```
sp_get_distributor
```

如果列分发的结果是 NULL，则未配置分发。请使用以下过程设置分发。

### 设置分发

1. 使用 SQL Server Management Studio (SSMS) 工具连接到 SQL Server 源数据库。
2. 打开复制文件夹的上下文 (右键单击) 菜单，然后选择配置分发。此时将显示“配置分发向导”。
3. 按照向导输入默认值并创建分发。

### 设置 CDC

AWS DMS 如果您不使用只读副本，3.4.7 及更高版本可以自动为您的数据库和所有表设置 MS CDC。要使用此功能，请将 `SetUpMsCdcForTables ECA` 设置为 `true`。有关 ECA 的信息，请参阅[端点设置](#)。

对于 3.4.7 AWS DMS 之前的版本或作为源的只读副本，请执行以下步骤：

1. 对于无主键的表，请为数据库设置 MS-CDC。要执行此操作，请使用分配了 sysadmin 角色的账户，然后运行以下命令。

```
use [DBname]
EXEC sys.sp_cdc_enable_db
```

2. 接下来，为每个源表设置 MS-CDC。对于具有唯一键但没有主键的每个表，运行以下查询来设置 MS-CDC。

```
exec sys.sp_cdc_enable_table
```

```
@source_schema = N'schema_name',  
@source_name = N'table_name',  
@index_name = N'unique_index_name',  
@role_name = NULL,  
@supports_net_changes = 1  
GO
```

3. 对于没有主键或唯一键的每个表，运行以下查询来设置 MS-CDC。

```
exec sys.sp_cdc_enable_table  
@source_schema = N'schema_name',  
@source_name = N'table_name',  
@role_name = NULL  
GO
```

有关为特定表设置 MS-CDC 的更多信息，请参阅 [SQL Server 文档](#)。

在独立 SQL Server 上设置持续复制：不使用 sysadmin 角色

有关不使用 sysadmin 角色在独立 SQL Server 上设置持续复制的信息，请参阅[在独立 SQL Server 上设置持续复制：不使用 sysadmin 角色](#)。

在云 SQL Server 数据库实例上设置持续复制

此部分介绍如何在云托管 SQL Server 数据库实例上设置 CDC。云托管的 SQL 服务器实例是在 Amazon RDS for SQL Server、Azure SQL 托管实例或任何其他托管云 SQL Server 实例上运行的实例。有关每种数据库类型的持续复制限制的信息，请参阅[使用 SQL Server 作为源代码的限制 AWS DMS](#)。

在设置持续复制之前，请参阅[在 SQL Server 源中使用持续复制 \(CDC\) 的先决条件](#)。

不同于自管理 Microsoft SQL Server 源，Amazon RDS for SQL Server 不支持 MS-Replication。因此，AWS DMS 需要对带或不带主键的表使用 MS-CDC。

Amazon RDS 不授予系统管理员设置用于源 SQL Server 实例中持续更改的复制项目的权限。AWS DMS 请确保在以下过程中使用主用户权限为 Amazon RDS 实例启用 MS-CDC。

在云 SQL Server 数据库实例上启用 MS-CDC

1. 在数据库级别运行以下查询之一。

对于 RDS for SQL Server 数据库实例，请使用此查询。



```
exec msdb.dbo.rds_cdc_enable_db 'DB_name'
```

对于 Azure SQL 托管数据库实例，请使用此查询。

```
USE DB_name  
GO  
EXEC sys.sp_cdc_enable_db  
GO
```

2. 对于带主键的每个表，运行以下查询来启用 MS-CDC。

```
exec sys.sp_cdc_enable_table  
@source_schema = N'schema_name',  
@source_name = N'table_name',  
@role_name = NULL,  
@supports_net_changes = 1  
GO
```

对于有唯一键但没有主键的每个表，运行以下查询来启用 MS-CDC。

```
exec sys.sp_cdc_enable_table  
@source_schema = N'schema_name',  
@source_name = N'table_name',  
@index_name = N'unique_index_name',  
@role_name = NULL,  
@supports_net_changes = 1  
GO
```

对于既没有主键又没有唯一键的每个表，运行以下查询来启用 MS-CDC。

```
exec sys.sp_cdc_enable_table  
@source_schema = N'schema_name',  
@source_name = N'table_name',  
@role_name = NULL  
GO
```

3. 使用以下命令设置要在源上可用的更改的保留期。

```
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'
```

参数 @pollinginterval 按秒计，建议值设置为 86399。当该参数为 @pollinginterval = 86399 时，这意味着事务日志将更改保留 86399 秒（一天）。过程 exec sp\_cdc\_start\_job 'capture' 启动这些设置。

#### Note

对于某些 SQL Server 版本，如果 pollinginterval 的值设置为超过 3599 秒，该值将重置为默认的五秒。发生这种情况时，T-Log 条目会在读取之前 AWS DMS 被清除。要确定哪些 SQL Server 版本受到此已知问题的影响，请参阅[这篇 Microsoft 知识库文章](#)。

如果您使用带有多可用区的 Amazon RDS，请确保也设置辅助项，以便在失效转移时具有正确的值。

```
exec rdsadmin..rds_set_configuration 'cdc_capture_pollinginterval' , 86399
```

如果捕获 SQL Server 源正在进行的更改的 AWS DMS 复制任务停止超过一小时，请使用以下步骤。

在执行 AWS DMS 复制任务期间保持保留期

1. 使用以下命令停止截断事务日志的作业。

```
exec sp_cdc_stop_job 'capture'
```

2. 在 AWS DMS 控制台上找到您的任务并继续执行任务。
3. 选择监控选项卡，然后勾选 CDCLatencySource 指标。
4. 如果 CDCLatencySource 指标等于 0（零）并保持不变，请使用以下命令重新启动截断事务日志的作业。

```
exec sp_cdc_start_job 'capture'
```

记得启动截断 SQL Server 事务日志的作业。否则，您的 SQL Server 实例上的存储空间可能会被填满。

## 云 SQL Server 数据库实例上的持续复制限制

- AWS DMS 仅支持活动事务日志的持续复制 (CDC)。您不能将备份日志用于 CDC。
- 如果将事件从活动事务日志移动到备份日志，或者将其从活动事务日志中截断，则可能丢失事件。

## 使用适用于 SQL Server 的 Amazon RDS 作为来源时的推荐设置 AWS DMS

当您使用 Amazon RDS for SQL Server 作为源时，捕获作业依赖于参数 `maxscans` 和 `maxtrans`。这些参数控制捕获对事务日志执行的最大扫描次数以及每次扫描处理的事务数。

对于事务数大于 `maxtrans*maxscans` 的数据库，增加其 `polling_interval` 值可能会导致活动事务日志记录的累积。这种累积反过来又会导致事务日志的大小增加。

请注意，这 AWS DMS 并不依赖于 MS-CDC 捕获作业。MS-CDC 捕获作业将事务日志条目标记为已处理。这样可允许事务日志备份作业从事务日志中删除条目。

我们建议您监控事务日志的大小和 MS-CDC 作业是否成功。如果 MS-CDC 作业失败，事务日志可能会过度增长并导致 AWS DMS 复制失败。您可以使用源数据库中的 `sys.dm_cdc_errors` 动态管理视图来监控 MS-CDC 捕获作业错误。您可以使用 `DBCC SQLPERF(LOGSPACE)` 管理命令来监控事务日志的大小。

### 解决由 MS-CDC 引起的事务日志增加问题

1. 检查数据库是否 AWS DMS 正在从中复制，并验证它是否持续增加。Log Space Used %

```
DBCC SQLPERF(LOGSPACE)
```

2. 确定阻碍事务日志备份过程的原因。

```
Select log_reuse_wait, log_reuse_wait_desc, name from sys.databases where name = db_name();
```

如果 `log_reuse_wait_desc` 值等于 `REPLICATION`，则说明日志备份保留是 MS-CDC 中的延迟所导致。

3. 通过增加 `maxtrans` 和 `maxscans` 参数值来增加捕获作业处理的事件数。

```
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'
```

要解决此问题，请将 `maxscans` 和 `maxtrans` 的值设置 `maxtrans*maxscans` 为等于每天为从源数据库 AWS DMS 复制的表生成的事件的平均数。

如果将这些参数设置为高于建议值，则捕获作业会处理事务日志中的所有事件。如果将这些参数设置为低于建议值，则 MS-CDC 延迟会增加，事务日志也会增加。

为 `maxscans` 和 `maxtrans` 确定合适的值可能很困难，因为工作负载的变化会产生不同数量的事件。在这种情况下，建议您对 MS-CDC 延迟设置监控。有关更多信息，请参阅 SQL Server 文档中的 [监控流程](#)。然后根据监控结果动态配置 `maxtrans` 和 `maxscans`。

如果 AWS DMS 任务找不到恢复或继续任务所需的日志序列号 (LSN)，则该任务可能会失败并需要完全重新加载。

#### Note

使用 AWS DMS 从 RDS for SQL Server 源复制数据时，在 Amazon RDS 实例的停止启动事件之后尝试恢复复制时，您可能会遇到错误。这是由于 SQL Server Agent 进程在停止启动事件后重新启动时，会重新启动捕获作业进程。这绕过了 MS-CDC 轮询间隔。

因此，在事务量低于 MS-CDC 捕获任务处理的数据库上，这可能会导致数据在从停止位置恢复之前 AWS DMS 被处理或标记为已复制和备份，从而导致以下错误：

```
[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)
```

要缓解此问题，请按照之前的建议设置 `maxtrans` 和 `maxscans` 值。

## 支持的 SQL Server 压缩方法

请注意以下有关 AWS DMS 支持的 SQL Server 压缩方法的内容：

- AWS DMS 在 SQL Server 版本 2008 及更高版本中支持行/页压缩。
- AWS DMS 不支持 Vardecimal 存储格式。
- AWS DMS 不支持稀疏列和列结构压缩。

## 使用自行管理的 SQL Server AlwaysOn 可用性组

SQL Server Always On 可用性组配有高可用性和灾难恢复解决方案，提供数据库镜像的企业级替代功能。

在中 AWS DMS，您可以从单个主可用性组或辅助可用性组副本迁移更改。

### 使用主可用性组副本

要在中使用主可用性组作为源 AWS DMS，请执行以下操作：

1. 为可用性副本中的所有 SQL Server 实例启用分发选项。有关更多信息，请参阅[在自我管理 SQL Server 上设置持续复制](#)。
2. 在 AWS DMS 控制台中，打开 SQL Server 源数据库设置。对于服务器名称，请指定为可用性组侦听器配置的域名服务 (DNS) 名称或 IP 地址。

首次启动 AWS DMS 任务时，可能需要比平时更长的时间才能启动。这种缓慢是因为可用性组服务器正在复制表项目的创建。

### 使用辅助可用性组副本

要在中使用辅助可用性组作为源 AWS DMS，请执行以下操作：

1. 使用与 AWS DMS 源端点用户相同的凭据来连接各个副本。
2. 确保您的 AWS DMS 复制实例可以解析所有现有副本的 DNS 名称并连接到它们。您可以使用以下 SQL 查询来获取所有副本的 DNS 名称。

```
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. 创建源端点时，请为端点的服务器名称或端点密钥的服务器地址指定可用性组侦听器的 DNS 名称。有关可用性组侦听器的更多信息，请参阅 SQL Server 文档中的[可用性组侦听器是什么？](#)。

您可以使用公有 DNS 服务器或本地 DNS 服务器来解析可用性组侦听器、主副本和辅助副本。要使用本地 DNS 服务器，请配置 Amazon Route 53 Resolver。有关更多信息，请参阅[使用您自己的本地名称服务器](#)。

4. 请将以下额外连接属性添加到您的源端点。

额外连接属性	值	注意事项
applicationIntent	ReadOnly	如果没有此 ODBC 设置，复制任务将路由到主可用性组副本。有关更多信息，请参阅 SQL Server 文档中的 <a href="#">对高可用性、灾难恢复的 SQL Server Native Client 支持</a> 。
multiSubnetFailover	yes	有关更多信息，请参阅 SQL Server 文档中的 <a href="#">对高可用性、灾难恢复的 SQL Server Native Client 支持</a> 。
alwaysOnSharedBackupIsEnabled	false	有关更多信息，请参阅 <a href="#">使用 SQL Server 作为源时的端点设置 AWS DMS</a> 。
activateSafeguard	false	有关更多信息，请参阅下面的 <a href="#">限制</a> 。
setUpMsDcForTables	false	有关更多信息，请参阅下面的 <a href="#">限制</a> 。

5. 在您可用性组的所有副本上启用分发选项。将所有节点添加到分发服务器列表中。有关更多信息，请参阅[设置分发](#)。
6. 对主读写副本运行以下查询以启用数据库的发布。您只对数据库运行一次此查询。

```
sp_replicationdboption @dbname = N'<source DB name>', @optname = N'publish', @value = N'true';
```

## 限制

以下是使用辅助可用性组副本的限制：

- AWS DMS 使用只读可用性组副本作为源时，不支持 Safeguard。有关更多信息，请参阅[使用 SQL Server 作为源时的端点设置 AWS DMS](#)。

- AWS DMS 使用只读可用性组副本作为源时，不支持 `setUpMsCdcForTables` 额外的连接属性。有关更多信息，请参阅 [使用 SQL Server 作为源时的端点设置 AWS DMS](#)。
- AWS DMS 从版本 3.4.7 开始，可以使用自我管理的辅助可用性组副本作为源数据库，用于持续复制（更改数据捕获或 CDC）。不支持云 SQL Server 多可用区只读副本。如果您使用早期版本的 AWS DMS，请确保使用主可用性组副本作为 CDC 的源数据库。

## 失效转移到其他节点

如果您将终端节点的 `ApplicationIntent` 额外连接属性设置为 `ReadOnly`，则您的 AWS DMS 任务将连接到只读路由优先级最高的只读节点。当优先级最高的只读节点不可用时，它会失效转移到可用性组中的其他只读节点。如果未设置 `ApplicationIntent`，则您的 AWS DMS 任务将仅连接到可用性组中的主（读/写）节点。

## 使用 SQL Server 作为源时的安全要求 AWS Database Migration Service

AWS DMS 用户帐户必须至少具有您要连接的源 SQL Server 数据库上的 `db_owner` 用户角色。

## 使用 SQL Server 作为源时的端点设置 AWS DMS

您可以使用端点设置来配置 SQL Server 源数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建源端点时指定设置，或者使用中带有 `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` [AWS CLI](#) 命令来指定设置。

下表显示了将 SQL Server 作为源时您可以使用的端点设置。

名称	描述
<code>ActivateSafeguard</code>	<p>此属性可开启或关闭 Safeguard。有关 Safeguard 的更多信息，请参阅下文的 <code>SafeguardPolicy</code>。</p> <p>默认值：<code>true</code></p> <p>有效值：<code>{false, true}</code></p> <p>例如：<code>'{"ActivateSafeguard": true}'</code></p>
<code>AlwaysOnSharedSync hedBackupIsEnabled</code>	<p>此属性调整从作为 Always On 可用性组群集一部分托管的 SQL Server 源数据库迁移 AWS DMS 时的行为。</p>

名称	描述
	<p>AWS DMS 增强了对配置为在 Always On 群集中运行的 SQL Server 源数据库的支持。在这种情况下，AWS DMS 会尝试跟踪事务备份是否是在 Always On 集群中的节点（而非托管源数据库实例的节点）上发生。在迁移任务启动 AWS DMS 时，尝试连接到集群中的每个节点，但如果无法连接到任何一个节点，则会失败。</p> <p>如果您 AWS DMS 需要轮询 Always On 集群中的所有节点以获取事务备份，请将此属性设置为 false。</p> <p>默认值：true</p> <p>有效值：true 或 false</p> <p>示例：'<code>{"AlwaysOnSharedSynchedBackupIsEnabled": false}</code>'</p>
<p>"ApplicationIntent": "readonly"</p>	<p>使用此 ODBC 驱动程序属性设置，SQL Server 可将您的复制任务路由到优先级最高的只读节点。如果不使用此设置，SQL Server 会将您的复制任务路由到主读写节点。</p>
<p>EnableNonSysadminWrapper</p>	<p>在不使用 sysadmin 用户的情况下在独立 SQL 服务器上设置持续复制时，请使用此端点设置。3.4.7 及更高 AWS DMS 版本支持此参数。有关在独立 SQL Server 上设置持续复制的信息，请参阅<a href="#">在独立 SQL Server 上设置持续复制：不使用 sysadmin 角色</a>。</p> <p>默认值：false</p> <p>有效值：true、false</p> <p>例如：'<code>{"EnableNonSysadminWrapper": true}</code>'</p>
<p>ExecuteTimeout</p>	<p>使用此额外连接属性（ECA）设置 SQL Server 实例的客户端语句超时时间（以秒为单位）。默认值为 60 秒。</p> <p>例如：'<code>{"ExecuteTimeout": 100}</code>'</p>



名称	描述
FatalOnSimpleModel	<p>如果设置为 true，当 SQL Server 数据库恢复模式设置为 simple 时，此设置会生成致命错误。</p> <p>默认值：false</p> <p>有效值：true 或 false</p> <p>示例：'{"FatalOnSimpleModel": true}'</p>
ForceLobLookup	<p>强制在内联 LOB 上执行 LOB 查找。</p> <p>默认值：false</p> <p>有效值：true、false</p> <p>例如：'{"ForceLobLookup": false}'</p>
"MultiSubnetFailover": "Yes"	<p>借助此 ODBC 驱动程序属性，DMS 可在可用性组失效转移时连接到新的主服务器。此属性专为连接中断或侦听器 IP 地址不正确的情况而设计。在这些情况下，AWS DMS 会尝试连接到与可用性组侦听器关联的所有 IP 地址。</p>
ReadBackupOnly	<p>使用此属性需要 sysadmin 权限。当此属性设置为 Y，在正在进行的复制过程中，只能从事务日志备份中 AWS DMS 读取更改，而不会从活动事务日志文件中读取更改。通过将此参数设置为 Y，可以在完全加载和持续复制任务期间控制活动事务日志文件的生长。但是，它会向持续复制添加一些源延迟。</p> <p>有效值：N 或 Y。默认值为 N。</p> <p>例如：'{"ReadBackupOnly": Y}'</p> <p>注意：由于 RDS 执行备份的方式，此参数在 Amazon RDS SQL Server 源实例上不起作用。</p>

名称	描述
SafeguardPolicy	<p>为了获得最佳性能，请 AWS DMS 尝试从活动事务日志 (TLOG) 中捕获所有未读更改。不过，有时由于截断，有效的 TLOG 可能未包含所有未读取的更改。发生这种情况时，AWS DMS 访问日志备份以捕获缺少的更改。为了最大限度地减少访问日志备份的需求，请使用以下方法之一 AWS DMS 防止截断：</p> <ol style="list-style-type: none"><li data-bbox="688 541 1503 1037">1. <b>RELY_ON_SQL_SERVER_REPLICATION_AGENT</b> (启动数据库中的事务)：这是的默认设置 AWS DMS。  使用此设置时，AWS DMS 要求运行 SQL Server 日志读取器代理，以便 AWS DMS 可以从活动 TLOG 中移动标记为复制的事务。请注意，如果未运行日志读取器代理，活动的 TLOG 可能会变满，从而导致源数据库切换到只读模式，直到您能够解决这个问题。如果您出于其他目的需要在数据库中启用 Microsoft 复制 AWS DMS，则必须选择此设置。</li><li data-bbox="688 1079 1503 1843">2. <b>EXCLUSIVE_AUTOMATIC_TRUNCATION</b> (仅用于 <b>sp_repldone</b> 单个任务)：使用此设置时 AWS DMS，可以完全控制将日志条目标记为 <b>ready for truncation</b> 使用的复制代理进程 <b>sp_repldone</b>。使用此设置时，AWS DMS 不会像 <b>RELY_ON_SQL_SERVER_REPLICATION_AGENT</b> (默认) 设置那样使用虚拟事务。只有当 MS Replication 不用于源数据库以外 AWS DMS 的任何其他目的时，才能使用此</li></ol>

名称	描述
	<p>设置。此外，使用此设置时，只有一个 AWS DMS 任务可以访问数据库。如果需要对同一个数据库运行 parallel AWS DMS 任务，请使用 <code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code>。</p> <ul style="list-style-type: none"> <li>此设置要求在数据库中停止日志读取器代理。如果任务启动时日志阅读器代理正在运行，则该 AWS DMS 任务将强制其停止。或者，也可以在启动此任务之前手动停止日志读取器代理。</li> <li>将此方法用于 MS-CDC 时，应停止并禁用 MS-CDC 捕获和 MS-CDC 清理作业。</li> <li>当 Microsoft SQL Server 迁移作业在远程 Distributor 计算机上运行时，你 AWS DMS 无法使用此设置，因为无法访问远程计算机。</li> <li><code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code> 在 Amazon RDS for SQL Server 源实例上不起作用，因为 Amazon RDS 用户无权运行 <code>sp_repldone</code> 存储过程。</li> <li>如果在不使用 <code>sysadmin</code> 角色的情况下将 <code>SafeguardPolicy</code> 设置为 <code>EXCLUSIVE_AUTOMATIC_TRUNCATION</code>，则必须向 <code>dmsuser</code> 用户授予 <code>dbo.syscategories</code> 和 <code>dbo.sysjobs</code> 对象的权限。</li> </ul> <p>默认值：<code>RELY_ON_SQL_SERVER_REPLICATION_AGENT</code></p> <p>有效值：<code>{EXCLUSIVE_AUTOMATIC_TRUNCATION、RELY_ON_SQL_SERVER_REPLICATION_AGENT}</code></p> <p>例如：<code>'{"SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION"}'</code></p>

名称	描述
SetUpMsCdcForTables	<p>此属性可为源数据库和任务映射中未启用 MS-Replication 的表启用 MS-CDC。将此值设置为 <code>true</code>，即可在源数据库上运行 <code>sp_cdc_enable_db</code> 存储过程，并在任务中未在源数据库中启用 MS-Replication 的每个表上运行 <code>sp_cdc_enable_table</code> 存储过程。有关启用分发的更多信息，请参阅<a href="#">在自我管理 SQL Server 上设置持续复制</a>。</p> <p>有效值：<code>{true、false}</code></p> <p>例如：<code>'{"SetUpMsCdcForTables": true}'</code></p>
TlogAccessMode	<p>表示用于获取 CDC 数据的模式。</p> <p>默认值：<code>PreferTlog</code></p> <p>有效值：<code>BackupOnly</code>、<code>PreferBackup</code>、<code>PreferTlog</code>、<code>TlogOnly</code></p> <p>例如：<code>'{"TlogAccessMode": "PreferTlog}"</code></p>
Use3rdPartyBackupDevice	<p>当此属性设置为 <code>Y</code> 时，如果第三方事务日志备份是以本机格式创建的，AWS DMS 会处理这些备份。</p>

## SQL Server 的源数据类型

使用 SQL Server 作为源的数据迁移 AWS DMS 支持大多数 SQL Server 数据类型。下表显示了使用时支持的 SQL Server 源数据类型 AWS DMS 以及 AWS DMS 数据类型的默认映射。

有关如何查看目标中映射的数据类型的信息，请参阅有关所使用的目标终端节点的部分。

有关 AWS DMS 数据类型的更多信息，请参见[AWS Database Migration Service 的数据类型](#)。

SQL Server 数据类型	AWS DMS 数据类型
BIGINT	INT8

SQL Server 数据类型	AWS DMS 数据类型
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 及更高版本)	DATETIME
SMALLDATETIME	DATETIME
DATE	DATE
TIME	TIME
DATETIMEOFFSET	WSTRING
CHAR	string
VARCHAR	string

SQL Server 数据类型	AWS DMS 数据类型
VARCHAR (max)	<p>CLOB</p> <p>TEXT</p> <p>要将此数据类型与一起使用 AWS DMS，必须为特定任务启用 CLOB 数据类型。</p> <p>对于 SQL Server 表，AWS DMS 即使对于不更改 SQL Server 中 LOB 列值的 UPDATE 语句，也会更新目标中的 LOB 列。</p> <p>在 CDC 期间，仅在包含主键的表中 AWS DMS 支持 CLOB 数据类型。</p>
NCHAR	WSTRING
NVARCHAR (length)	WSTRING
NVARCHAR (max)	<p>NCLOB</p> <p>NTEXT</p> <p>要将此数据类型与一起使用 AWS DMS，必须 SupportLobs 为特定任务启用使用。有关启用 Lob 支持的更多信息，请参阅<a href="#">在任务中为源数据库设置 LOB 支持 AWS DMS</a>。</p> <p>对于 SQL Server 表，AWS DMS 即使对于不更改 SQL Server 中 LOB 列值的 UPDATE 语句，也会更新目标中的 LOB 列。</p> <p>在 CDC 期间，仅在包含主键的表中 AWS DMS 支持 CLOB 数据类型。</p>
BINARY	BYTES
VARBINARY	BYTES

SQL Server 数据类型	AWS DMS 数据类型
VARBINARY (max)	<p>BLOB</p> <p>IMAGE</p> <p>对于 SQL Server 表，AWS DMS 即使对于不更改 SQL Server 中 LOB 列值的 UPDATE 语句，也会更新目标中的 LOB 列。</p> <p>要将此数据类型与一起使用 AWS DMS，必须允许将 BLOB 数据类型用于特定任务。</p> <p>AWS DMS 仅在包含主键的表中支持 BLOB 数据类型。</p>
TIMESTAMP	BYTES
UNIQUEIDENTIFIER	string
HIERARCHYID	<p>在复制到 SQL Server 目标终端节点时，使用 HIERARCHYID。</p> <p>在复制到所有其他目标终端节点时，使用 WSTRING (250)。</p>
XML	<p>NCLOB</p> <p>对于 SQL Server 表，AWS DMS 即使对于不更改 SQL Server 中 LOB 列值的 UPDATE 语句，也会更新目标中的 LOB 列。</p> <p>要将此数据类型与一起使用 AWS DMS，必须允许将 NCLOB 数据类型用于特定任务。</p> <p>在 CDC 期间，仅在包含主键的表中 AWS DMS 支持 NCLOB 数据类型。</p>

SQL Server 数据类型	AWS DMS 数据类型
GEOMETRY	<p>在复制到支持此数据类型的目标终端节点时，使用 GEOMETRY。</p> <p>在复制到不支持此数据类型的目标终端节点时，使用 CLOB。</p>
GEOGRAPHY	<p>在复制到支持此数据类型的目标终端节点时，使用 GEOGRAPHY。</p> <p>在复制到不支持此数据类型的目标终端节点时，使用 CLOB。</p>

AWS DMS 不支持包含以下数据类型的字段的表。

- CURSOR
- SQL\_VARIANT
- TABLE

#### Note

根据用户定义的数据类型的基类型来支持这些类型。例如，基于 DATETIME 的用户定义的数据类型将作为 DATETIME 数据类型进行处理。

## 将 Microsoft Azure SQL 数据库作为 AWS DMS 源

在使用 AWS DMS 时，可以将 Microsoft Azure SQL 数据库作为来源，其方式与 SQL Server 大致相同。作为来源，AWS DMS 支持的数据库版本列表，与在本地或 Amazon EC2 实例上运行的 SQL Server 支持的数据库版本列表相同。

有关更多信息，请参阅[使用微软 SQL Server 数据库作为来源 AWS DMS](#)。

#### Note

AWS DMS 在 Azure SQL 数据库中不支持更改数据捕获 (CDC) 操作。



## 使用 Microsoft Azure SQL 托管实例作为 AWS DMS 的来源

在使用 AWS DMS 时，可以将 Microsoft Azure SQL 托管实例作为来源，其方式与 SQL Server 大致相同。作为来源，AWS DMS 支持的数据库版本列表，与在本地或 Amazon EC2 实例上运行的 SQL Server 支持的数据库版本列表相同。

有关更多信息，请参阅[使用微软 SQL Server 数据库作为来源 AWS DMS](#)。

## 将 Microsoft Azure Database for PostgreSQL 灵活服务器作为 AWS DMS 的来源

在使用 AWS DMS 时，您可以将 Microsoft Azure Database for PostgreSQL 灵活服务器作为来源，其方式与 PostgreSQL 大致相同。

有关 AWS DMS 支持作为来源的 Microsoft Azure Database for PostgreSQL 灵活服务器版本的信息，请参阅[的来源 AWS DMS](#)。

## 设置 Microsoft Azure Database for PostgreSQL 灵活服务器以进行逻辑复制和解码

在数据库迁移期间，您可以使用 Microsoft Azure Database for PostgreSQL 灵活服务器中的逻辑复制和解码功能。

对于逻辑解码，DMS 使用 `test_decoding` 或 `pglogical` 插件。如果 `pglogical` 插件在源 PostgreSQL 数据库上可用，则 DMS 使用 `pglogical` 创建复制插槽，否则使用 `test_decoding` 插件。

要将您的 Microsoft Azure for PostgreSQL 灵活服务器配置为 DMS 的来源端点，请执行以下步骤：

1. 在门户上打开“服务器参数”页面。
2. 将 `wal_level` 服务器参数设置为 `LOGICAL`。
3. 如果要使用 `pglogical` 扩展，请将 `shared_preload_libraries` 和 `azure.extensions` 参数设置为 `pglogical`。
4. 将 `max_replication_slots` 参数设置为您计划同时运行的最大 DMS 任务数。在 Microsoft Azure 中，此参数的默认值是 10。此参数的最大值取决于 PostgreSQL 实例的可用内存，每 GB 内存允许使用 2 到 8 个复制插槽。
5. 将 `max_wal_senders` 参数设置为一个大于 1 的值。`max_wal_senders` 参数用于设置可以运行的并发任务数。默认值是 10。
6. 将 `max_worker_processes` 参数值设置为至少 16。否则，您可能会看到如下错误：

```
WARNING: out of background worker slots.
```

7. 保存更改。重启服务器以应用更改。
8. 确认您的 PostgreSQL 实例允许来自所连接资源的网络流量。
9. 使用以下命令向现有用户授予复制权限，或创建具有复制权限的新用户。
  - 使用以下命令向现有用户授予复制权限：

```
ALTER USER <existing_user> WITH REPLICATION;
```

- 使用以下命令创建具有复制权限的新用户：

```
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;
```

有关使用 PostgreSQL 的逻辑复制的更多信息，请参阅以下主题：

- [使用逻辑复制启用更改数据捕获 \( CDC \)](#)
- [使用本机 CDC 开始点设置 PostgreSQL 源的 CDC 加载](#)
- [Azure Database for PostgreSQL 文档](#)中的 [Azure Database for PostgreSQL – 灵活服务器中的逻辑复制和逻辑解码](#)。

## 将 Microsoft Azure Database for MySQL 灵活服务器作为 AWS DMS 的来源

在使用 AWS DMS 时，您可以将 Microsoft Azure Database for MySQL 灵活服务器作为来源，其方式与 MySQL 大致相同。

有关 AWS DMS 支持作为来源的 Microsoft Azure Database for MySQL 灵活服务器版本的信息，请参阅 [的来源 AWS DMS](#)。

有关将客户管理的 MySQL 兼容数据库与 AWS DMS 一起使用的更多信息，请参阅[使用自管理的 MySQL 兼容数据库作为源 AWS DMS](#)。

## 将 Azure MySQL 作为 AWS Database Migration Service 的来源时的限制

- Azure MySQL 灵活服务器系统变量 `sql_generate_invisible_primary_key` 的默认值为 ON，并且服务器会自动将生成的不可见主键 (GIPK) 添加到创建的任何没有显式主键的表中。AWS DMS 不支持对具有 GIPK 约束的 MySQL 表进行持续复制。

## 使用 OCI MySQL Heatwave 作为 AWS DMS 的来源

使用 AWS DMS，您可以将 OCI MySQL Heatwave 作为来源，其方式与 MySQL 大致相同。使用 OCI MySQL Heatwave 作为来源需要进行一些额外的配置更改。

有关 AWS DMS 支持作为来源的 OCI MySQL Heatwave 版本的信息，请参阅[的来源 AWS DMS](#)。

### 设置 OCI MySQL Heatwave 进行逻辑复制

要将您的 OCI MySQL Heatwave 实例配置为 DMS 的源端点，请执行以下操作：

1. 登录 OCI 控制台，然后打开左上角的主汉堡菜单 (≡)。
2. 选择数据库、数据库系统。
3. 打开配置菜单。
4. 选择 Create configuration (创建配置)。
5. 输入配置名称，例如 **dms\_configuration**。
6. 选择当前 OCI MySQL Heatwave 实例的形状。您可以在实例的数据库系统配置属性选项卡的数据库系统配置：形状部分下找到形状。
7. 在用户变量部分中，选择 `binlog_row_value_options` 系统变量。其默认值为 `PARTIAL_JSON`。清除该值。
8. 选择创建按钮。
9. 打开 OCI MySQL Heatwave 实例，然后选择编辑按钮。
10. 在配置部分，选择更改配置按钮，然后选择您在步骤 4 中创建的形状配置。
11. 更改生效后，实例便已准备就绪，可以进行逻辑复制。

## 将 Google Cloud for MySQL 作为 AWS DMS 的来源

使用 AWS DMS，您可以将 Google Cloud for MySQL 作为来源，其方式与 MySQL 大致相同。

有关 AWS DMS 支持作为来源的 GCP MySQL 版本的信息，请参阅[的来源 AWS DMS](#)。

有关更多信息，请参阅[将与 MySQL 兼容的数据库作为 AWS DMS 源](#)。

#### Note

AWS DMS 版本 3.4.6 支持将 GCP MySQL 8.0 作为来源。

对于 GCP for MySQL 实例，AWS DMS 不支持 SSL 模式 `verify-full`。

不支持 GCP MySQL 安全设置 `Allow only SSL connections`，因为它需要服务器和客户端证书验证。AWS DMS 仅支持服务器证书验证。

对于 `binlog_checksum` 数据库标志，AWS DMS 支持默认 GCP CloudSQL for MySQL 值 `CRC32`。

## 将 Google Cloud for PostgreSQL 作为 AWS DMS 的来源

在使用 AWS DMS 时，您可以将 Google Cloud for PostgreSQL 作为来源，其方式与自管理 PostgreSQL 数据库大致相同。

有关 AWS DMS 支持作为来源的 GCP PostgreSQL 版本的信息，请参阅[的来源 AWS DMS](#)。

有关更多信息，请参阅[将 PostgreSQL 数据库作为 AWS DMS 源](#)。

### 设置 Google Cloud for PostgreSQL 以进行逻辑复制和解码

在数据库迁移期间，您可以使用 Google Cloud for PostgreSQL 中的逻辑复制和解码功能。

对于逻辑解码，DMS 使用以下插件之一：

- `test_decoding`
- `pglogical`

如果 `pglogical` 插件在源 PostgreSQL 数据库上可用，则 DMS 使用 `pglogical` 创建复制插槽，否则使用 `test_decoding` 插件。

请注意以下通过 AWS DMS 使用逻辑解码的相关注意事项：

1. 使用 Google Cloud for PostgreSQL，通过将 `cloudsql.logical_decoding` 标志设置为 `on` 来启用逻辑解码。
2. 要启用 `pglogical`，请将 `cloudsql.enable_pglogical` 标志设置为 `on`，然后重新启动数据库。

3. 要使用逻辑解码功能，请使用 REPLICATION 属性创建一个 PostgreSQL 用户。使用 pglogical 扩展时，用户必须具有 cloudsqsuperuser 角色。要创建具有 cloudsqsuperuser 角色的用户，请执行以下操作：

```
CREATE USER new_aws_dms_user WITH REPLICATION
IN ROLE cloudsqsuperuser LOGIN PASSWORD 'new_aws_dms_user_password';
```

要在现有用户上设置此属性，请执行以下操作：

```
ALTER USER existing_user WITH REPLICATION;
```

4. 将 max\_replication\_slots 参数设置为您计划同时运行的最大 DMS 任务数。在 Google Cloud SQL 中，此参数的默认值是 10。此参数的最大值取决于 PostgreSQL 实例的可用内存，每 GB 内存允许使用 2 到 8 个复制插槽。

有关使用 PostgreSQL 的逻辑复制的更多信息，请参阅以下主题：

- [使用逻辑复制启用更改数据捕获 \( CDC \)](#)
- [使用本机 CDC 开始点设置 PostgreSQL 源的 CDC 加载](#)
- [Cloud SQL for PostgreSQL 文档中的设置逻辑复制和解码。](#)

## 将 PostgreSQL 数据库作为 AWS DMS 源

您可以使用迁移一个或多个 PostgreSQL 数据库中的数据。AWS DMS 将 PostgreSQL 数据库作为源时，您可以将数据迁移到另一个 PostgreSQL 数据库或其他支持的数据库。

有关支持作为源代码的 PostgreSQL 版本的信息 AWS DMS ，请参阅。[的来源 AWS DMS](#)

AWS DMS 对于以下类型的数据库，支持 PostgreSQL：

- 本地数据库
- Amazon EC2 实例上的数据库
- Amazon RDS 数据库实例上的数据库
- 基于 Amazon Aurora PostgreSQL 兼容版的数据库实例上的数据库
- 基于 Amazon Aurora PostgreSQL 兼容无服务器版的数据库实例上的数据库

**Note**

DMS 仅支持将 Amazon Aurora PostgreSQL – Serverless V1 作为完全加载的源。但您可以使用 Amazon Aurora PostgreSQL – Serverless V2 作为完全加载、完全加载 + CDC 和仅限 CDC 的任务的源。

**AWS DMS 要使用的版本**

使用任何可用 AWS DMS 版本。

使用 3.4.3 及更高 AWS DMS 版本。

使用 3.4.7 及更高 AWS DMS 版本。

使用 3.5.1 及更高 AWS DMS 版本。

使用 3.5.3 及更高 AWS DMS 版本。

可以使用安全套接字层 ( SSL ) 加密 PostgreSQL 端点与复制实例之间的连接。有关将 SSL 用于 PostgreSQL 终端节点的更多信息，请参阅[将 SSL 与 AWS Database Migration Service](#)。

使用 PostgreSQL 作为源的附加安全要求是，指定的用户账户必须是 PostgreSQL 数据库中的注册用户。

要将 PostgreSQL 数据库配置为 AWS DMS 源端点，请执行以下操作：

- 创建具有相应权限的 PostgreSQL 用户，以 AWS DMS 提供对您的 PostgreSQL 源数据库的访问权限。

**Note**

- 如果您的 PostgreSQL 源数据库是自管理数据库，请参阅[使用自我管理的 PostgreSQL 数据库作为源代码 AWS DMS](#)了解更多信息。
- 如果您的 PostgreSQL 源数据库由 Amazon RDS 管理，请参阅[使用 AWS 托管的 PostgreSQL 数据库作为 DMS 源](#)了解更多信息。

- 创建符合所选 PostgreSQL 数据库配置的 PostgreSQL 源端点。
- 创建一项或一组任务来迁移您的表。

要创建 full-load-only 任务，无需进一步配置端点。

在创建更改数据捕获任务（仅 CDC，或者完全加载和 CDC 任务）之前，请参阅[使用自我管理的 PostgreSQL 数据库作为源启用 CDC AWS DMS](#)或[使用 AWS 托管的 PostgreSQL 数据库实例启用 CDC AWS DMS](#)。

**主题**

- [使用自我管理的 PostgreSQL 数据库作为源代码 AWS DMS](#)
- [使用 AWS 托管的 PostgreSQL 数据库作为 DMS 源](#)
- [使用逻辑复制启用更改数据捕获（CDC）](#)
- [使用本机 CDC 开始点设置 PostgreSQL 源的 CDC 加载](#)
- [使用从 PostgreSQL 迁移到 PostgreSQL AWS DMS](#)
- [使用 Babelfish 为亚马逊 Aurora PostgreSQL 迁移 AWS DMS](#)
- [从 PostgreSQL 源数据库中移除 AWS DMS 构件](#)
- [将 PostgreSQL 数据库作为 DMS 源时的额外配置设置](#)
- [使用 P MapBooleanAsBoolean PostgreSQL 端点设置](#)
- [使用 PostgreSQL 作为 DMS 源时的端点设置和额外连接属性 \(ECA\)](#)
- [将 PostgreSQL 数据库作为 DMS 源的限制](#)
- [PostgreSQL 的源数据类型](#)

## 使用自我管理的 PostgreSQL 数据库作为源代码 AWS DMS

使用自管理的 PostgreSQL 数据库作为源，您可以将数据迁移到另一个 PostgreSQL 数据库，也可以将数据迁移到支持的其他目标数据库。AWS DMS 数据库源可以是在 Amazon EC2 实例上运行的本地数据库，也可以是自我管理引擎。您可以将数据库实例用于完全加载任务，也可以用于更改数据捕获 ( CDC ) 任务。

### 使用自行管理的 PostgreSQL 数据库作为源的先决条件 AWS DMS

在从自我管理 PostgreSQL 源数据库迁移数据之前，请执行以下操作：

- 请确保使用版本为 9.4.x 或更高版本的 PostgreSQL 数据库。
- 对于完全加载加上 CDC 任务或仅限 CDC 任务，请向为 PostgreSQL 源数据库指定的用户账户授予超级用户权限。该用户账户访问源中特定于复制的函数需要超级用户权限。对于仅完全加载任务，该用户账户需要表的 SELECT 权限以迁移它们。
- 将 AWS DMS 复制服务器的 IP 地址添加到 `pg_hba.conf` 配置文件中，并启用复制和套接字连接。下面是一个示例。

```
# 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
```

PostgreSQL 的 `pg_hba.conf` 配置文件控制客户端身份验证。( HBA 代表基于主机的身份验证。 ) 文件通常存储在数据库集群的数据目录中。

- 如果您要使用将数据库配置为逻辑复制源，AWS DMS 请参阅 [使用自我管理的 PostgreSQL 数据库作为源启用 CDC AWS DMS](#)

#### Note

有些 AWS DMS 事务会闲置一段时间，然后才会让 DMS 引擎再次使用它们。在 PostgreSQL 版本 9.6 及更高版本中使用参数 `idle_in_transaction_session_timeout`，能够导致空闲事务超时并失败。当您使用 AWS DMS 时，请勿结束空闲事务。



## 使用自我管理的 PostgreSQL 数据库作为源启用 CDC AWS DMS

AWS DMS 支持使用逻辑复制捕获变更数据 (CDC)。要启用自我管理 PostgreSQL 源数据库的逻辑复制，请在 `postgresql.conf` 配置文件中设置以下参数和值：

- 设置 `wal_level = logical`。
- 将 `max_replication_slots` 设置为一个大于 1 的值。

请根据您需要运行的任务数设置 `max_replication_slots` 值。例如，要运行五个任务，您需要设置至少五个时隙。在任务启动时时隙立即打开，即使任务不再运行仍保持打开。请务必手动删除空闲时隙。请注意，如果 DMS 创建了时隙，在删除任务后，DMS 会自动删除重复的时隙。

- 将 `max_wal_senders` 设置为一个大于 1 的值。

`max_wal_senders` 参数用于设置可以运行的并发任务数。

- `wal_sender_timeout` 参数会终止处于不活动状态的时间超过指定毫秒数的复制连接。本地 PostgreSQL 数据库的默认值为 60000 毫秒 (60 秒)。将此值设置为 0 (零) 会禁用超时机制，这是 DMS 的有效设置。

将 `wal_sender_timeout` 设置为非零值时，执行 CDC 的 DMS 任务至少需要 10000 毫秒 (10 秒)，如果此值小于 10000，则任务会失败。将此值保持至少 5 分钟，以免在 DMS 复制实例执行多可用区失效转移期间造成延迟。

有些参数是静态的，只能在服务器启动时设置。在服务器重新启动之前，对配置文件 (对于自我管理数据库) 或数据库参数组 (对于 RDS for PostgreSQL 数据库) 中的条目所做的任何更改都将被忽略。有关更多信息，请参阅 [PostgreSQL 文档](#)。

有关启用 CDC 的更多信息，请参阅 [使用逻辑复制启用更改数据捕获 \(CDC\)](#)。

## 使用 AWS 托管的 PostgreSQL 数据库作为 DMS 源

您可以使用 AWS 托管的 PostgreSQL 数据库实例作为源。AWS DMS 您可以使用 AWS 托管的 PostgreSQL 源执行完全加载任务和更改数据捕获 (CDC) 任务。

使用 AWS 托管的 PostgreSQL 数据库作为 DMS 源的先决条件

从 AWS 托管的 PostgreSQL 源数据库迁移数据之前，请执行以下操作：

- 我们建议您使用具有 PostgreSQL 数据库实例所需最低权限的 AWS 用户账户作为 PostgreSQL 源终端节点的用户账户。AWS DMS 建议不要使用主账户。该账户必须具有 `rds_superuser` 和

`rds_replication` 角色。`rds_replication` 角色可授予管理逻辑时隙和使用逻辑时隙流式处理数据的权限。

请务必使用您所用账户的主用户账户创建多个对象。有关创建这些项目的信息，请参阅[不使用主用户账户迁移 Amazon RDS for PostgreSQL 数据库](#)。

- 如果源数据库位于虚拟私有云 ( VPC ) 中，请选择一个 VPC 安全组，该安全组提供对数据库所在的数据库实例的访问权限。这是 DMS 复制实例成功连接到源数据库实例所必需的。当数据库和 DMS 复制实例位于同一 VPC 中时，将相应的安全组添加到其自己的入站规则中。

#### Note

有些 AWS DMS 事务会闲置一段时间，然后才会让 DMS 引擎再次使用它们。在 PostgreSQL 版本 9.6 及更高版本中使用参数 `idle_in_transaction_session_timeout`，能够导致空闲事务超时并失败。当您使用 AWS DMS 时，请勿结束空闲事务。

使用 AWS 托管的 PostgreSQL 数据库实例启用 CDC AWS DMS

AWS DMS 当数据库实例配置为使用逻辑复制时，支持 Amazon RDS PostgreSQL 数据库上的 CDC。下表汇总了每个 AWS 托管的 PostgreSQL 版本的逻辑复制兼容性。

不能将 RDS PostgreSQL 只读副本用于 CDC ( 持续复制 )。

PostgreSQL 版本	AWS DMS 满载支持	AWS DMS 疾病预防控制中心支持
Aurora PostgreSQL 版本 2.1 ( 与 PostgreSQL 10.5 兼容 ) 或更低版本	是	不支持
Aurora PostgreSQL 版本 2.2 ( 与 PostgreSQL 10.6 兼容 ) 或更高版本	支持	是
RDS for PostgreSQL ( 与 PostgreSQL 10.21 兼容 ) 或更高版本	支持	是

## 为 RDS PostgreSQL 数据库实例启用逻辑复制

1. 使用 Po AWS stgreSQL 数据库实例的主用户账户作为 PostgreSQL 源终端节点的用户账户。主用户账户具有允许它设置 CDC 所需的角色。

如果您使用主用户账户之外的账户，请确保从要使用的账户的主账户创建多个对象。有关更多信息，请参阅 [不使用主用户账户迁移 Amazon RDS for PostgreSQL 数据库](#)。

2. 在数据库集群参数组中将 `rds.logical_replication` 参数设置为 1。此静态参数要求重启数据库实例才能生效。在应用该参数期间，AWS DMS 设置 `wal_level`、`max_wal_senders`、`max_replication_slots` 和 `max_connections` 参数。这些参数更改可以增加预写日志 (WAL) 生成，因此，只有在您使用逻辑复制槽时才设置 `rds.logical_replication`。
3. `wal_sender_timeout` 参数会终止处于不活动状态的时间超过指定毫秒数的复制连接。AWS 由托管的 PostgreSQL 数据库的默认值为 30000 毫秒 (30 秒)。将此值设置为 0 (零) 会禁用超时机制，这是 DMS 的有效设置。

将 `wal_sender_timeout` 设置为非零值时，执行 CDC 的 DMS 任务至少需要 10000 毫秒 (10 秒)，如果此值介于 0 和 10000 之间，则会失败。将此值保持至少 5 分钟，以免在 DMS 复制实例执行多可用区失效转移期间造成延迟。

4. 确保数据库集群参数组中 `max_worker_processes` 参数的值大于或等于 `max_logical_replication_workers`、`autovacuum_max_workers` 和 `max_parallel_workers` 的总组合值。大量后台工作进程可能会影响小型实例上的应用程序工作负载。因此，如果为 `max_worker_processes` 设置的值高于默认值，请监控数据库的性能。
5. 使用 Aurora PostgreSQL 作为 CDC 的来源时，请设置为 `synchronous_commit ON`

## 不使用主用户账户迁移 Amazon RDS for PostgreSQL 数据库

在某些情况下，您可能不会将主用户账户用于将作为源的 Amazon RDS PostgreSQL 数据库实例。在这些情况下，您需要创建多个对象来捕获数据定义语言 (DDL) 事件。您可以在主用户账户之外的账户中创建这些对象，然后在主用户账户中创建触发器。

### Note

如果在源端点上将 `captureDDLs` 端点设置为 `false`，您不需要在源数据库上创建以下表和触发器。

使用以下过程创建这些对象。

## 创建对象

1. 选择要在其中创建对象的架构。默认架构为 `public`。确保架构存在并且可由 *OtherThanMaster* 账户访问。
2. 使用除主账户之外的用户账户（此处为 *OtherThanMaster* 账户）登录 PostgreSQL 数据库实例。
3. 运行以下命令以创建表 `awsdms_ddl_audit`，并将以下代码中的 *objects\_schema* 替换为要使用的架构名称。

```
CREATE TABLE objects_schema.awsdms_ddl_audit
(
  c_key    bigserial primary key,
  c_time   timestamp,      -- Informational
  c_user   varchar(64),    -- Informational: current_user
  c_txn    varchar(16),    -- Informational: current transaction
  c_tag    varchar(24),    -- Either 'CREATE TABLE' or 'ALTER TABLE' or 'DROP TABLE'
  c_oid    integer,       -- For future use - TG_OBJECTID
  c_name   varchar(64),    -- For future use - TG_OBJECTNAME
  c_schema varchar(64),    -- For future use - TG_SCHEMANAME. For now - holds
  current_schema
  c_ddlqry text           -- The DDL query associated with the current DDL event
);
```

4. 运行以下命令以创建函数 `awsdms_intercept_ddl`，并将以下代码中的 *objects\_schema* 替换为要使用的架构名称。

```
CREATE OR REPLACE FUNCTION objects_schema.awsdms_intercept_ddl()
  RETURNS event_trigger
LANGUAGE plpgsql
SECURITY DEFINER
AS $$
  declare _qry text;
BEGIN
  if (tg_tag='CREATE TABLE' or tg_tag='ALTER TABLE' or tg_tag='DROP TABLE' or
  tg_tag = 'CREATE TABLE AS') then
    SELECT current_query() into _qry;
```

```

insert into objects_schema.awsdms_ddl_audit
values
(
default,current_timestamp,current_user,cast(TXID_CURRENT()as
varchar(16)),tg_tag,0,'',current_schema,_qry
);
delete from objects_schema.awsdms_ddl_audit;
end if;
END;
$$;

```

5. 注销 *OtherThanMaster* 账户并使用分配了 `rds_superuser` 角色的账户登录。
6. 运行以下命令以创建事件触发器 `awsdms_intercept_ddl`。

```

CREATE EVENT TRIGGER awsdms_intercept_ddl ON ddl_command_end
EXECUTE PROCEDURE objects_schema.awsdms_intercept_ddl();

```

7. 确保访问这些事件的所有用户和角色都具有必要的 DDL 权限。例如：

```

grant all on public.awsdms_ddl_audit to public;
grant all on public.awsdms_ddl_audit_c_key_seq to public;

```

在完成前面的过程后，可以使用 *OtherThanMaster* 账户创建 AWS DMS 源终端节点。

### Note

这些事件由 CREATE TABLE、ALTER TABLE 和 DROP TABLE 语句触发。

## 使用逻辑复制启用更改数据捕获 ( CDC )

在 PostgreSQL 源的数据库迁移期间，您可以使用 PostgreSQL 的本机逻辑复制功能启用更改数据捕获 ( CDC )。您可以将此功能与自我管理 PostgreSQL 数据库实例以及 Amazon RDS for PostgreSQL SQL 数据库实例一起使用。这种方法可减少停机时间，并确保目标数据库与源 PostgreSQL 数据库同步。

AWS DMS 支持使用主键的 PostgreSQL 表的 CDC。如果某个表没有主键，提前写入日志 ( WAL ) 将不包含数据库行的前映像。在这种情况下，DMS 无法更新此表。在这里，您可以使用其他配置设置，

并使用表副本身份作为解决方法。但是，这种方法可能会生成额外的日志。我们建议您仅在仔细测试后才使用表副本身份作为解决方法。有关更多信息，请参阅 [将 PostgreSQL 数据库作为 DMS 源时的额外配置设置](#)。

### Note

逻辑解码插件支持 REPLICA IDENTITY FULL，但 pglogical 插件不支持。有关更多信息，请参阅 [pglogical 文档](#)。

对于满负荷以及仅限 CDC 和 CDC 的任务，AWS DMS 使用逻辑复制槽保留 WAL 日志以进行复制，直到日志被解码。对于完全加载和 CDC 任务或仅限 CDC 的任务，在重启（而非恢复）时，系统会重新创建复制槽。

### Note

对于逻辑解码，DMS 使用 test\_decoding 或 pglogical 插件。如果 pglogical 插件在源 PostgreSQL 数据库上可用，DMS 会使用 pglogical 创建复制槽，否则使用 test\_decoding 插件。有关 test\_decoding 插件的更多信息，请参阅 [PostgreSQL 文档](#)。

如果将数据库参数 max\_slot\_wal\_keep\_size 设置为非默认值，并且复制槽的 restart\_lsn 落后于当前 LSN 的大小超过此大小，DMS 任务会因为删除了所需的 WAL 文件而失败。

## 配置 pglogical 插件

pglogical 插件是作为 PostgreSQL 扩展实现的，是一种逻辑复制系统和选择性数据复制模型。下表列出了支持 pglogical 插件的源 PostgreSQL 数据库版本。

PostgreSQL 源	支持 pglogical
自管理 PostgreSQL 9.4 或更高版本	是
Amazon RDS PostgreSQL 9.5 或更低版本	否
Amazon RDS PostgreSQL 9.6 或更高版本	是
Aurora PostgreSQL 1.x 到 2.5.x	否

PostgreSQL 源	支持 pglogical
Aurora PostgreSQL 2.6.x 或更高版本	是
Aurora PostgreSQL 3.3.x 或更高版本	是

在配置 pglogical 以与一起使用之前 AWS DMS，请先在 PostgreSQL 源数据库上启用更改数据捕获 (CDC) 的逻辑复制。

- 有关在自管理 PostgreSQL 源数据库上为 CDC 启用逻辑复制的信息，请参阅[使用自我管理的 PostgreSQL 数据库作为源启用 CDC AWS DMS](#)。
- 有关在 AWS 托管的 PostgreSQL 源数据库上为 CDC 启用逻辑复制的信息，请参阅[使用 AWS 托管的 PostgreSQL 数据库实例启用 CDC AWS DMS](#)。

在 PostgreSQL 源数据库上启用逻辑复制后，请使用以下步骤配置用于 DMS 的 pglogical。

使用 pglogical 插件在 PostgreSQL 源数据库上进行逻辑复制 AWS DMS

1. 在 PostgreSQL 源数据库上创建一个 pglogical 扩展：
  - a. 设置正确的参数：
    - 对于自管理 PostgreSQL 数据库，请设置数据库参数 `shared_preload_libraries='pglogical'`。
    - 对于 Amazon RDS 上的 PostgreSQL 和 Amazon Aurora PostgreSQL 兼容版数据库，请在同一个 RDS 参数组中将参数 `shared_preload_libraries` 设置为 `pglogical`。
  - b. 重启您的 PostgreSQL 源数据库。
  - c. 在 PostgreSQL 数据库上，运行此命令 `create extension pglogical;`
2. 运行以下命令验证是否已成功安装 pglogical：

```
select * FROM pg_catalog.pg_extension
```

现在，您可以创建一个 AWS DMS 任务，为 PostgreSQL 源数据库端点执行更改数据捕获。

**Note**

如果您未在 PostgreSQL 源数据库上启用 `pglogical`，AWS DMS 默认使用 `test_decoding` 插件。当启用 `pglogical` 进行逻辑解码时，默认情况下 AWS DMS 使用 `pglogical`。但是您可以设置额外连接属性，`PluginName` 会改用 `test_decoding` 插件。

## 使用本机 CDC 开始点设置 PostgreSQL 源的 CDC 加载

要使用 PostgreSQL 作为源启用本机 CDC 开始点，请在创建端点时将 `slotName` 额外连接属性设置为现有逻辑复制槽的名称。此逻辑复制槽保留从创建终端节点时起持续进行的更改，因此它支持从前一个时间点进行复制。

PostgreSQL 将数据库更改写入 WAL 文件，这些文件只有在 AWS DMS 成功读取逻辑复制槽中的更改后才会丢弃。使用逻辑复制槽可以防止记录的更改在它们被复制引擎使用之前删除。

但是，根据更改速率和消耗量，保留在逻辑复制槽中的更改可能会导致磁盘使用率提高。我们建议您在逻辑复制槽时，在源 PostgreSQL 实例中设置空间使用情况警报。有关设置 `slotName` 额外连接属性的更多信息，请参阅 [使用 PostgreSQL 作为 DMS 源时的端点设置和额外连接属性 \(ECA\)](#)。

以下过程将更详细地演练这种方法。

### 使用本机 CDC 开始点设置 PostgreSQL 源终端节点的 CDC 加载

1. 确定您要用作开始点的更早复制任务（父任务）所使用的逻辑复制槽。然后，查询源数据库上的 `pg_replication_slots` 视图，以确保此槽没有任何活动的连接。如果有，请先予以解决并关闭它们，然后再继续。

对于以下步骤，假定您的逻辑复制槽为

```
abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef.
```

2. 创建一个新的源终端节点，其中包含以下额外连接属性设置。

```
slotName=abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef;
```

3. 使用控制台 AWS CLI 或 AWS DMS API 创建仅限 CDC 的新任务。例如，通过使用 CLI，您可以运行以下 `create-replication-task` 命令。

```
aws dms create-replication-task --replication-task-identifier postgresql-slot-name-test
```



```
--source-endpoint-arn arn:aws:dms:us-  
west-2:012345678901:endpoint:ABCD1EFGHIJK2LMNOPQRST3UV4  
--target-endpoint-arn arn:aws:dms:us-  
west-2:012345678901:endpoint:ZYX9WVUTSRQ0NM8LKJIHGF7ED6  
--replication-instance-arn arn:aws:dms:us-  
west-2:012345678901:rep:AAAAAAAAAAAA5BB4CCC3DDDD2EE  
--migration-type cdc --table-mappings "file://mappings.json" --cdc-start-position  
"4AF/B00000D0"  
--replication-task-settings "file://task-pg.json"
```

在前面的命令中，设置了以下选项：

- `source-endpoint-arn` 选项设置为您在步骤 2 中创建的新值。
- `replication-instance-arn` 选项设置为与步骤 1 中的父任务相同的值。
- `table-mappings` 和 `replication-task-settings` 选项设置为与步骤 1 中的父任务相同的值。
- `cdc-start-position` 选项设置为起始位置值。要查找此起始位置，请查询源数据库上的 `pg_replication_slots` 视图，或查看步骤 1 中父任务的控制台详细信息。有关更多信息，请参阅 [确定 CDC 本机开始点](#)。

要在使用控制 AWS DMS 台创建仅限 CDC 的新任务时启用自定义 CDC 启动模式，请执行以下操作：

- 在任务设置部分，对于源事务的 CDC 启动模式，请选择启用自定义 CDC 启动模式。
- 对于源事务的自定义 CDC 开始点，请选择指定日志序列号。指定系统更改号或选择指定恢复检查点，然后提供恢复检查点。

当此 CDC 任务运行时，如果指定的逻辑复制槽不存在，则会 AWS DMS 引发错误。如果未使用有效的 `cdc-start-position` 设置创建任务，它也会引发错误。

在 `pglogical` 插件中使用本机 CDC 开始点并且想要使用新的复制槽时，请在创建 CDC 任务之前完成以下设置步骤。

使用以前未在另一个 DMS 任务中创建的新复制槽

1. 创建复制槽，如下所示：

```
SELECT * FROM pg_create_logical_replication_slot('replication_slot_name',  
'pglogical');
```

2. 数据库创建复制槽后，将获取并记下此槽的 `restart_lsn` 和 `confirmed_flush_lsn` 值：

```
select * from pg_replication_slots where slot_name like 'replication_slot_name';
```

请注意，在复制槽之后创建的 CDC 任务的本机 CDC 开始位置不能早于 `confirmed_flush_lsn` 值。

有关 `restart_lsn` 和 `confirmed_flush_lsn` 值的信息，请参阅 [pg\\_replication\\_slots](#)

3. 创建一个 `pglogical` 节点。

```
SELECT pglogical.create_node(node_name := 'node_name', dsn := 'your_dsn_name');
```

4. 使用 `pglogical.create_replication_set` 函数创建两个复制集。第一个复制集跟踪有主键的表的更新和删除。第二个复制集仅跟踪插入，其名称与第一个复制集相同，只是添加了前缀“i”。

```
SELECT pglogical.create_replication_set('replication_slot_name', false, true, true,  
false);  
SELECT pglogical.create_replication_set('ireplication_slot_name', true, false,  
false, true);
```

5. 向复制集添加表。

```
SELECT pglogical.replication_set_add_table('replication_slot_name',  
'schemaname.tablename', true);  
SELECT pglogical.replication_set_add_table('ireplication_slot_name',  
'schemaname.tablename', true);
```

6. 创建源端点时，请设置以下额外连接属性（ECA）。

```
PluginName=PGLOGICAL;slotName=slot_name;
```

现在，您可以使用新的复制槽创建具有 PostgreSQL 本机开始点的仅限 CDC 的任务。有关 `pglogical` 插件的更多信息，请参阅 [pglogical 3.7 文档](#)。

## 使用从 PostgreSQL 迁移到 PostgreSQL AWS DMS

当你从 PostgreSQL 以外的数据库引擎迁移到 PostgreSQL 数据库时 AWS DMS，几乎总是最好的迁移工具。但是，当您从 PostgreSQL 数据库迁移到 PostgreSQL 数据库时，PostgreSQL 工具的效率更高。

### 使用 PostgreSQL 本机工具迁移数据

我们建议您在以下条件下使用 PostgreSQL 数据库迁移工具（例如 `pg_dump`）：

- 您有一个从源 PostgreSQL 数据库向目标 PostgreSQL 数据库的同构迁移。
- 您正在迁移整个数据库。
- 利用这些本机工具，可以在迁移您的数据时最大程度地减少停机时间。

`pg_dump` 实用程序使用 `COPY` 命令创建 PostgreSQL 数据库的架构和数据转储。`pg_dump` 生成的转储脚本将数据加载到一个同名的数据库中，然后重新创建表、索引和外键。要将数据恢复到具有不同名称的数据库，您可以使用 `pg_restore` 命令和 `-d` 参数。

如果您要将数据从在 EC2 上运行的 PostgreSQL 源数据库迁移到 Amazon RDS for PostgreSQL 目标，则可以使用 `pglogical` 插件。

有关将 PostgreSQL 数据库导入 Amazon RDS for PostgreSQL 或 Amazon Aurora PostgreSQL 兼容版的更多信息，请参阅<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/PostgreSQL.Procedural.Importing.html>。

### 使用 DMS 将数据从 PostgreSQL 迁移到 PostgreSQL

AWS DMS 例如，可以将数据从本地的 PostgreSQL 源数据库迁移到适用于 PostgreSQL 的目标 Amazon RDS 或 Aurora PostgreSQL 实例。核心或基本 PostgreSQL 数据类型通常可以成功迁移。

#### Note

将分区表从 PostgreSQL 源复制到 PostgreSQL 目标时，您无需在 DMS 任务的选择标准中提及父表。提及父表会导致目标上子表中的数据重复，这可能造成 PK 违规。在表映射选择标准中单独选择子表，即会自动填充父表。

源数据库支持但目标数据库不支持的数据类型可能无法成功迁移。AWS DMS 如果数据类型未知，则将某些数据类型作为字符串流式传输。对于某些数据类型（例如 XML 和 JSON），如果文件较小则可以成功迁移，但如果是较大的文档，迁移会失败。

执行数据类型迁移时，需要注意以下事项：

- 在某些情况下，PostgreSQL NUMERIC(p,s) 数据类型不指定任何精度和小数位数。对于 DMS 版本 3.4.2 及更早版本，DMS 默认使用的精度为 28，小数位数为 6，即 NUMERIC(28,6)。例如，源中的值 0.611111104488373 在 PostgreSQL 目标中将转换为 0.611111。
- 数据类型为 ARRAY 的表必须具有主键。缺少主键的 ARRAY 数据类型的表在完全加载期间会暂停。

下表显示了源 PostgreSQL 数据类型以及是否可以成功迁移这些数据类型。

数据类型	成功迁移	部分迁移	不迁移	注释
INTEGER	X			
SMALLINT	X			
BIGINT	X			
NUMERIC/DECIMAL(p,s)		X		当 $0 < p < 39$ 且 $0 < s$ 时
NUMERIC/DECIMAL		X		其中 $p > 38$ 或 $p = s = 0$
REAL	X			
DOUBLE	X			
SMALLSERIAL	X			
SERIAL	X			
BIGSERIAL	X			
MONEY	X			
CHAR		X		未指定精度
CHAR(n)	X			
VARCHAR		X		未指定精度

数据类型	成功迁移	部分迁移	不迁移	注释
VARCHAR (n)	X			
TEXT	X			
BYTEA	X			
TIMESTAMP	X			正无穷值和负无穷值分别被截断为“99 99-12-31 23:59:59”和“4713-01 -01 00:00:00 BC”。
TIMESTAMP ( 有时区 )		X		
DATE	X			
TIME	X			
带时区的时间		X		
INTERVAL		X		
BOOLEAN	X			
ENUM			X	
CIDR	X			
INET			X	
MACADDR			X	
TSVECTOR			X	
TSQUERY			X	

数据类型	成功迁移	部分迁移	不迁移	注释
XML		X		
POINT	X			PostGIS 空间数据类型
LINE			X	
LSEG			X	
BOX			X	
路径			X	
POLYGON	X			PostGIS 空间数据类型
CIRCLE			X	
JSON		X		
ARRAY	X			需要主键
COMPOSITE			X	
RANGE			X	
LINestring	X			PostGIS 空间数据类型
MULTIPOINT	X			PostGIS 空间数据类型
MULTILINestring	X			PostGIS 空间数据类型
MULTIPOLYGON	X			PostGIS 空间数据类型

数据类型	成功迁移	部分迁移	不迁移	注释
GEOMETRYCOLLECTION	X			PostGIS 空间数据类型

## 迁移 PostGIS 空间数据类型

空间数据 标识空间中对象或位置的几何信息。PostgreSQL 对象关系数据库支持 PostGIS 空间数据类型。

在迁移 PostgreSQL 空间数据对象之前，请确保在全局级别启用 PostGIS 插件。这样做可以确保为 PostgreSQL 目标数据库实例 AWS DMS 创建精确的源空间数据列。

对于 PostgreSQL 到 PostgreSQL 的同构 AWS DMS 迁移，支持 PostGIS 几何和地理（大地坐标）数据对象类型和子类型的迁移，例如：

- POINT
- LINESTRING
- POLYGON
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION

## 使用 Babelfish 为亚马逊 Aurora PostgreSQL 迁移 AWS DMS

您可以使用将 Babelfish for Aurora PostgreSQL 源表迁移到任何支持的目标终端节点。AWS DMS

在使用 DMS 控制台、API 或 CLI 命令创建 AWS DMS 源终端节点时，将源设置为 Amazon Aurora PostgreSQL，将数据库名称设置为 **babelfish\_db**。在“端点设置”部分中，确保设置 DatabaseMode 为 Babelfish，并将其 BabelfishDatabaseName 设置为源 Babelfish T-SQL 数据库的名称。与其使用 Babelfish TCP 端口 **1433**，不如使用 Aurora PostgreSQL TCP 端口 **5432**。

在迁移数据之前，您必须创建表，以确保 DMS 使用正确的数据类型和表元数据。如果您在运行迁移之前没有在目标上创建表，则 DMS 可能会使用不正确的数据类型和权限创建表。

## 向迁移任务添加转换规则

在为 Babelfish 源创建迁移任务时，需要包含转换规则，以确保 DMS 使用预先创建的目标表。

如果您在定义 Babelfish for PostgreSQL 集群时设置了多数据库迁移模式，请添加一条转换规则，将架构名称重命名为 T-SQL 架构。例如，如果 T-SQL 架构名称为 dbo，而您的 Babelfish for PostgreSQL 架构名称为 mydb\_dbo，则使用转换规则将架构重命名为。dbo 要查找 PostgreSQL 架构名称，[请参阅 亚马逊 Aurora 用户指南中的 Babelfish 架构](#)。

如果您使用单数据库模式，则无需使用转换规则来重命名数据库架构。PostgreSQL 架构名称与 T-SQL 数据库中的架构名称存在映射。one-to-one

以下转换规则示例显示了如何将架构名称从 mydb\_dbo 回重命名为 dbo：

```
{
  "rules": [
    {
      "rule-type": "transformation",
      "rule-id": "566251737",
      "rule-name": "566251737",
      "rule-target": "schema",
      "object-locator": {
        "schema-name": "mydb_dbo"
      },
      "rule-action": "rename",
      "value": "dbo",
      "old-value": null
    },
    {
      "rule-type": "selection",
      "rule-id": "566111704",
      "rule-name": "566111704",
      "object-locator": {
        "schema-name": "mydb_dbo",
        "table-name": "%"
      },
      "rule-action": "include",
      "filters": []
    }
  ]
}
```



## 在 Babelfish 表中使用 PostgreSQL 源端点的限制

在 Babelfish 表中使用 PostgreSQL 源端点时，存在以下限制：

- DMS 仅支持从 Babelfish 版本 16.2/15.6 及更高版本以及 DMS 版本 3.5.3 及更高版本迁移。
- DMS 不会将 Babelfish 表定义更改复制到目标端点。解决此限制的一种解决方法是先将表定义更改应用于目标，然后在 Babelfish 源代码上更改表定义。
- 当您创建数据类型为 BYTEA 的 Babelfish 表时，DMS 会在迁移到 SQL Server 作为目标时将其转换为该 varbinary(max) 数据类型。
- DMS 不支持二进制数据类型的完整 LOB 模式。改为对二进制数据类型使用受限 LOB 模式。
- DMS 不支持将 Babelfish 作为数据源进行数据验证。
- 对于“目标表格准备模式”任务设置，请仅使用“什么都不做”或“截断”模式。不要使用删除目标中的表模式。在目标上使用 Drop tables 时，DMS 可能会使用不正确的数据类型创建表。
- 使用持续复制 (CDC 或满载和 CDC) 时，请将 PluginName 额外连接属性 (ECA) 设置为 TEST\_DECODING。

## 从 PostgreSQL 源数据库中移除 AWS DMS 构件

要捕获 DDL 事件，请在迁 AWS DMS 移任务启动时在 PostgreSQL 数据库中创建各种构件。完成任务后，您可能需要删除这些项目。

要删除项目，请发布以下语句（按显示顺序），{AmazonRDSMigration} 是在其中创建项目的架构：在删除架构时应尤为谨慎。请勿删除正在使用的架构，特别是非公有的操作架构。

```
drop event trigger awsdms_intercept_ddl;
```

事件触发器不属于特定架构。

```
drop function {AmazonRDSMigration}.awsdms_intercept_ddl()  
drop table {AmazonRDSMigration}.awsdms_ddl_audit  
drop schema {AmazonRDSMigration}
```

## 将 PostgreSQL 数据库作为 DMS 源时的额外配置设置

在从 PostgreSQL 数据库迁移数据时，您可以使用两种方法来添加额外配置设置：

- 您可以将值添加到额外连接属性来捕获 DDL 事件，以及指定在其中创建操作 DDL 数据库项目的架构。有关更多信息，请参阅 [使用 PostgreSQL 作为 DMS 源时的端点设置和额外连接属性 \(ECA\)](#)。

- 您可以覆盖连接字符串参数。选择此选项可执行以下任一操作：
  - 指定内部 AWS DMS 参数。此类参数很少需要，因此不在用户界面中公开。
  - 为特定数据库客户端指定直通 ( 直通 ) 值。AWS DMS 在传递给数据库客户端的连接 sting 中包含直通参数。
- 通过在 PostgreSQL 9.4 及更高版本中使用表级参数 REPLICA IDENTITY，您可以控制写入预写日志 ( WAL ) 的信息。特别是，对于标识已更新或删除行的 WAL，它会这样做。REPLICA IDENTITY FULL 记录此行中所有列的旧值。将 REPLICA IDENTITY FULL 用于每个表时请慎重，因为 FULL 会生成额外的 WAL 量 ( 可能是不必要的 )。有关更多信息，请参阅 [ALTER TABLE-REPLICA IDENTITY](#)。

## 使用 P MapBooleanAsBoolean ostgreSQL 端点设置

您可以使用 PostgreSQL 端点设置将一个布尔值作为从 PostgreSQL 源映射到 Amazon Redshift 目标的布尔值。默认情况下，BOOLEAN 类型迁移为 varchar(5)。您可以指定 MapBooleanAsBoolean 让 PostgreSQL 将布尔类型迁移为布尔值，如以下示例所示。

```
--postgre-sql-settings '{"MapBooleanAsBoolean": true}'
```

请注意，您必须在源端点和目标端点上都设置此设置才能使其生效。

由于 MySQL 没有 BOOLEAN 类型，因此在将 BOOLEAN 数据迁移到 MySQL 时，请使用转换规则而不是此设置。

## 使用 PostgreSQL 作为 DMS 源时的端点设置和额外连接属性 (ECA)

您可以使用端点设置和额外连接属性 (ECA) 来配置您的 PostgreSQL 源数据库。在使用 AWS DMS 控制台创建源终端节点时，您可以指定终端节点设置，或者使用中带有 `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` [AWS CLI](#) 命令来指定终端节点设置。

下表显示了可以将 PostgreSQL 用作源的端点设置和 ECA。

属性名称	描述
CaptureDDLs	要捕获 DDL 事件，请在任务 AWS DMS 启动时在 PostgreSQL 数据库中创建各种工件。稍后可以删除这些项目，如 <a href="#">从 PostgreSQL 源数据库中移除 AWS DMS 构件</a> 中所述。

属性名称	描述
	<p>如果此值设置为 <code>false</code>，您无需在源数据库上创建表或触发器。</p> <p>捕获流式处理的 DDL 事件。</p> <p>默认值：<code>true</code></p> <p>有效值：<code>true/false</code></p> <p>例如：<code>--postgre-sql-settings '{"CaptureDDLs": true}'</code></p>
ConsumeMonotonicEvents	<p>用于控制如何复制具有重复日志序列号 (LSN) 的整体事务。当此参数为 <code>false</code> 时，将在目标上使用和复制具有重复 LSN 的事件。当此参数为 <code>true</code> 时，仅复制第一个事件，而不会在目标上使用或复制具有重复 LSN 的事件。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>false/true</code></p> <p>例如：<code>--postgre-sql-settings '{"ConsumeMonotonicEvents": true}'</code></p>
DdlArtifactsSchema	<p>设置在其中创建操作 DDL 数据库项目的架构。</p> <p>默认值：<code>public</code></p> <p>有效值：字符串</p> <p>例如：<code>--postgre-sql-settings '{"DdlArtifactsSchema": "xyzddl_schema"}'</code></p>
ExecuteTimeout	<p>设置 PostgreSQL 实例的客户端语句超时 (以秒为单位)。默认值为 60 秒。</p> <p>例如：<code>--postgre-sql-settings '{"ExecuteTimeout": 100}'</code></p>

属性名称	描述
FailTasksOnLobTruncation	<p>在设置为 true 时，如果 LOB 列的实际大小大于指定的 LobMaxSize ，则此值将导致任务失败。</p> <p>如果任务设置为“受限 LOB 模式”并且该选项设置为 true ，任务将失败，而不是截断 LOB 数据。</p> <p>默认值：false</p> <p>有效值：布尔值</p> <p>例如：<code>--postgre-sql-settings '{"FailTasksOnLobTruncation": true}'</code></p>
fetchCacheSize	<p>此额外连接属性 ( ECA ) 用于设置光标在完全加载操作期间将提取的行数。根据复制实例中可用的资源，您可以将此值调高或调低。</p> <p>默认值：10000</p> <p>有效值：数字</p> <p>ECA 示例：<code>fetchCacheSize=10000;</code></p>
HeartbeatFrequency	<p>设置 WAL 检测信号频率 ( 以分钟为单位 ) 。</p> <p>默认值：5</p> <p>有效值：数字</p> <p>例如：<code>--postgre-sql-settings '{"HeartbeatFrequency": 1}'</code></p>

属性名称	描述
HeartbeatSchema	<p>设置在其中创建检测信号构件的架构。</p> <p>默认值：public</p> <p>有效值：字符串</p> <p>例如：<code>--postgre-sql-settings '{"HeartbeatSchema": "xyzheartbeatSchema"}'</code></p>
MapJsonbAsClob	<p>默认情况下，将 JSONB AWS DMS 映射到 NCLOB。您可以指定 MapJsonbAsClob 让 PostgreSQL 将 JSONB 类型迁移为 CLOB。</p> <p>例如：<code>--postgre-sql-settings='{"MapJsonbAsClob": "true"}'</code></p>
MapLongVarcharAs	<p>默认情况下，将 VARCHAR AWS DMS 映射到 WSTRING。您可以指定 MapLongVarcharAs 让 PostgreSQL 将 VARCHAR(N) 类型（其中 N 大于 16387）迁移到以下类型：</p> <ul style="list-style-type: none"><li>• WSTRING</li><li>• CLOB</li><li>• NCLOB</li></ul> <p>例如：<code>--postgre-sql-settings='{"MapLongVarcharAs": "CLOB"}'</code></p>

属性名称	描述
MapUnboundedNumericAsString	<p>此参数将具有无界 NUMERIC 数据类型的列视为 STRING，以便在不损失数值精度的前提下成功迁移。此参数仅用于从 PostgreSQL 源复制到 PostgreSQL 目标或与 PostgreSQL 兼容的数据库。</p> <p>默认值：false</p> <p>有效值：false/true</p> <p>例如：<code>--postgre-sql-settings '{"MapUnboundedNumericAsString": true}'</code></p> <p>使用此参数可能会导致某些复制性能下降，因为需要从数字转换为字符串，然后再转换回数字。DMS 版本 3.4.4 及更高版本支持使用此参数</p> <div data-bbox="688 909 1507 1465" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>只能同时在 PostgreSQL 源端点和目标端点中使用 MapUnboundedNumericAsString。在 CDC 期间，在源 PostgreSQL 端点上使用 MapUnboundedNumericAsString 会将精度限制在 28 以内。在目标端点上使用 MapUnboundedNumericAsString 时，使用精度 28、小数位数 6 迁移数据。请勿将 MapUnboundedNumericAsString 与非 PostgreSQL 目标一起使用。</p> </div>
PluginName	<p>指定用于创建副本槽的插件。</p> <p>有效值：pglogical、test_decoding</p> <p>例如：<code>--postgre-sql-settings '{"PluginName": "test_decoding"}'</code></p>

属性名称	描述
SlotName	<p>为 PostgreSQL 源实例的 CDC 加载设置先前创建的逻辑复制槽的名称。</p> <p>与 AWS DMS API <code>CdcStartPosition</code> 请求参数一起使用时，此属性还允许使用本地 CDC 起点。DMS 在启动 CDC 加载任务之前验证指定的逻辑复制槽是否存在。它还验证此任务是否使用有效的 <code>CdcStartPosition</code> 设置创建的。如果指定的槽不存在或任务没有有效的 <code>CdcStartPosition</code> 设置，DMS 将引发错误。</p> <p>有关设置 <code>CdcStartPosition</code> 请求参数的更多信息，请参阅 <a href="#">确定 CDC 本机开始点</a>。有关使用 <code>CdcStartPosition</code> 的更多信息，请参阅 <a href="#">AWS Database Migration Service API 参考</a> 中的 <code>CreateReplicationTask</code>、<code>StartReplicationTask</code> 和 <code>ModifyReplicationTask</code> API 操作的文档。</p> <p>有效值：字符串</p> <p>例如：<code>--postgre-sql-settings '{"SlotName": "abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef"}'</code></p>
unboundedVarcharMaxSize	<p>此额外连接属性 (ECA) 定义了定义为 <code>VarChar</code> 没有最大长度说明符的类型的数据库列的最大大小。默认值为 8000 字节。最大值为 10485760 字节。</p>

## 将 PostgreSQL 数据库作为 DMS 源的限制

将 PostgreSQL 作为 AWS DMS 的源时，以下限制适用：

- AWS DMS 不适用于亚马逊 RDS for PostgreSQL 10.4 或 Amazon Aurora PostgreSQL 10.4 作为源或目标。
- 捕获的表必须具有一个主键。如果表没有主键，则 AWS DMS 忽略该表的 DELETE 和 UPDATE 记录操作。有关解决方法，请参阅 [使用逻辑复制启用更改数据捕获 \(CDC\)](#)。

注意：我们不建议在没有主键/唯一索引的情况下进行迁移，否则会有其他限制，例如“没有”批量应用功能、完整 LOB 功能、数据验证，以及无法高效复制到 Redshift 目标。

- AWS DMS 忽略更新主键段的尝试。在这些情况下，目标会将更新标识为未更新任何行的更新。但是，由于更新 PostgreSQL 中的主键的结果是不可预测的，不会将任何记录写入异常表。
- AWS DMS 不支持“从时间戳开始处理更改”运行选项。
- AWS DMS 不会复制因分区或子分区操作（ADDDROP、或TRUNCATE）而产生的更改。
- 复制多个同名但名称具有不同大小写的表（例如，table1、TABLE1 和 Table1）会导致无法预测的行为。由于这个问题，AWS DMS 不支持这种类型的复制。
- 在大多数情况下，AWS DMS 支持对表的 CREATE、ALTER 和 DROP DDL 语句进行更改处理。AWS DMS 如果表保存在内部函数或过程主体块或其他嵌套结构中，则不支持这种变更处理。

例如，不捕获以下更改。

```
CREATE OR REPLACE FUNCTION attu.create_distributors1() RETURNS void
LANGUAGE plpgsql
AS $$
BEGIN
create table attu.distributors1(did serial PRIMARY KEY,name
varchar(40) NOT NULL);
END;
$$;
```

- 目前，PostgreSQL 源中的 boolean 数据类型会作为具有不一致值的 bit 数据类型迁移到 SQL Server 目标。解决方法是，使用列 VARCHAR(1) 的数据类型预先创建表（或者让 AWS DMS 创建表）。然后，让下游处理将“F”视为 False，将“T”视为 True。
- AWS DMS 不支持 TRUNCATE 操作的更改处理。
- OID LOB 数据类型不会迁移到目标。
- AWS DMS 仅支持同构迁移的 PostGIS 数据类型。
- 如果您的源是本地或 Amazon EC2 实例上的 PostgreSQL 数据库，请确保将 test\_decoding 输出插件安装到源端点上。您可以在 PostgreSQL contrib 包中找到此插件。有关测试解码插件的更多信息，请参阅 [PostgreSQL 文档](#)。
- AWS DMS 不支持更改处理以设置和取消设置列的默认值（在 ALTER TABLE 语句上使用 ALTER COLUMN SET DEFAULT 子句）。
- AWS DMS 不支持通过更改处理来设置列的空性（在 ALTER TABLE 语句上使用 ALTER COLUMN [SET|DROP] NOT NULL 子句）。



- 启用逻辑复制后，每个事务在内存中保留的最大更改数为 4MB。之后，更改会溢出到磁盘。因此，ReplicationSlotDiskUsage 会增加，并且在事务完成/停止并且回滚完成之前，restart\_lsn 不会有进展。因为这是一个长时间运行的事务，可能需要花费较长时间回滚。因此，在启用逻辑复制时，请避免长时间运行的事务或许多子事务。请改为将事务分成几个较小的事务。

在 Aurora PostgreSQL 版本 13 及更高版本中，您可以调整 `logical_decoding_work_mem` 参数以控制 DMS 何时将更改数据溢出到磁盘。有关更多信息，请参阅 [在 Aurora PostgreSQL 中泄露文件](#)。

- 数据类型为 ARRAY 的表必须具有主键。缺少主键的 ARRAY 数据类型的表在完全加载期间会暂停。
- AWS DMS 不支持分区表的复制。检测到分区表时，将显示以下内容：
  - 终端节点报告列出父表和子表。
  - AWS DMS 将目标表创建为常规表，其属性与所选表相同。
  - 如果源数据库中的父表具有与子表相同的主键值，则会产生“重复的键”错误。
- 要将分区表从 PostgreSQL 源复制到 PostgreSQL 目标，请先在目标上手动创建父表和子表。然后，定义单独的任务来向这些表进行复制。在这种情况下，请将任务配置设置为 Truncate before loading。
- PostgreSQL NUMERIC 数据类型不是固定大小。当传输的数据属于 NUMERIC 数据类型，但没有精度和小数位数时，默认情况下，DMS 使用 NUMERIC(28,6) (精度为 28，小数位数为 6)。例如，源中的值 0.6111111104488373 在 PostgreSQL 目标中将转换为 0.611111。
- AWS DMS 仅支持 Aurora PostgreSQL Serverless V1 作为满载任务的来源。AWS DMS 支持 Aurora PostgreSQL Serverless V2 作为满载、满载和 CDC 以及仅限 CDC 的任务的来源。
- AWS DMS 不支持复制使用合并函数创建的具有唯一索引的表。
- 使用 LOB 模式时，源表和相应的目标表必须具有相同的主键。如果其中一个表没有主键，DELETE 和 UPDATE 记录操作的结果将不可预测。
- 使用并行加载功能时，不支持根据分区或子分区对表进行分段。有关并行加载的更多信息，请参阅[对所选表、视图和集合使用并行加载](#)。
- AWS DMS 不支持延迟约束。
- AWS DMS 版本 3.4.7 支持 PostgreSQL 14.x 作为源代码，但有以下限制：
  - AWS DMS 不支持对两阶段提交的更改处理。
  - AWS DMS 不支持逻辑复制以流式传输正在进行的长时间事务。
- AWS DMS 不支持 CDC for Amazon RDS Proxy for PostgreSQL 作为来源。
- 使用不包含主键列的[源过滤器](#)时，不会捕获 DELETE 操作。

- 如果您的源数据库也是另一个第三方复制系统的目标，DDL 更改可能不会在 CDC 期间迁移。因为这种情况可能会防止触发 `awsdms_intercept_ddl` 事件触发器。要解决这种情况，请在源数据库上修改此触发器，如下所示：

```
alter event trigger awsdms_intercept_ddl enable always;
```

- AWS DMS 不支持 CDC for Amazon RDS 多可用区数据库集群 PostgreSQL 作为源，因为适用于 PostgreSQL 的 RDS 多可用区数据库集群不支持逻辑复制。

## PostgreSQL 的源数据类型

下表显示了 AWS DMS 使用时支持的 PostgreSQL 源数据类型以及与数据类型的默认映射 AWS DMS。

有关如何查看目标中映射的数据类型的信息，请参阅有关所使用的目标终端节点的部分。

有关 AWS DMS 数据类型的更多信息，请参见 [AWS Database Migration Service 的数据类型](#)。

PostgreSQL 数据类型	DMS 数据类型
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
NUMERIC (p,s)	如果精度介于 0 和 38 之间，则使用 NUMERIC。 如果精度为 39 或更大的值，则使用 STRING。
DECIMAL(P,S)	如果精度介于 0 和 38 之间，则使用 NUMERIC。 如果精度为 39 或更大的值，则使用 STRING。
REAL	REAL4
DOUBLE	REAL8

PostgreSQL 数据类型	DMS 数据类型
SMALLSERIAL	INT2
SERIAL	INT4
BIGSERIAL	INT8
MONEY	NUMERIC(38,4)  MONEY 数据类型将映射到 SQL Server 中的 FLOAT。
CHAR	WSTRING (1)
CHAR(N)	WSTRING (n)
VARCHAR(N)	WSTRING (n)
TEXT	NCLOB
CITEXT	NCLOB
BYTEA	BLOB
TIMESTAMP	DATETIME
TIMESTAMP ( 有时区 )	DATETIME
DATE	DATE
TIME	TIME
带时区的时间	TIME
INTERVAL	STRING(128) – 1 YEAR、2 MONTHS、3 DAYS、4 HOURS、5 MINUTES、6 SECONDS
BOOLEAN	CHAR (5) false 或 true
ENUM	STRING (64)

PostgreSQL 数据类型	DMS 数据类型
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))"
路径	CLOB "((x1,y1),(xn,yn))"
POLYGON	CLOB "((x1,y1),(xn,yn))"
CIRCLE	STRING (255) "(x,y),r"
JSON	NCLOB
JSONB	NCLOB
ARRAY	NCLOB
COMPOSITE	NCLOB

PostgreSQL 数据类型	DMS 数据类型
HSTORE	NCLOB
INT4RANGE	STRING (255)
INT8RANGE	STRING (255)
NUMRANGE	STRING (255)
STRRANGE	STRING (255)

### 使用 PostgreSQL 的 LOB 源数据类型

PostgreSQL 列大小将影响 PostgreSQL LOB 数据类型到 AWS DMS 数据类型的转换。为此，请对以下 AWS DMS 数据类型执行下列步骤：

- BLOB – 在创建任务时，将 LOB 大小限制为设置为最大 LOB 大小 ( KB ) 值。
- CLOB – 复制将每个字符作为 UTF8 字符处理。因此，找到列中最长字符文本的长度，此处显示为 `max_num_chars_text`。使用此长度指定将 LOB 大小限制为的值。如果数据包含 4 字节字符，请乘以 2 以指定 Limit LOB size to (将 LOB 大小限制为) 值 (以字节为单位)。在此示例中，Limit LOB size to (将 LOB 大小限制为) 等于 `max_num_chars_text` 乘以 2。
- NCLOB – 复制将每个字符作为双字节字符处理。因此，找到列中最长字符文本的长度 ( `max_num_chars_text` ) 并乘以 2。执行此操作来指定将 LOB 大小限制为的值。在此示例中，Limit LOB size to (将 LOB 大小限制为) 等于 `max_num_chars_text` 乘以 2。如果数据包含 4 字节字符，请再次乘以 2。在此示例中，Limit LOB size to (将 LOB 大小限制为) 等于 `max_num_chars_text` 乘以 4。

## 将与 MySQL 兼容的数据库作为 AWS DMS 源

您可以使用 Database Migration Service 从任何兼容 MySQL 的数据库 ( MySQL、MariaDB 或 Amazon Aurora MySQL ) 迁移数据。AWS

有关 AWS DMS 支持作为来源的 MySQL 版本的信息，请参阅 [来源 AWS DMS](#)。

您可以使用 SSL 来加密 MySQL 兼容终端节点与复制实例之间的连接。有关将 SSL 用于 MySQL 兼容终端节点的更多信息，请参阅 [将 SSL 与 AWS Database Migration Service](#)。

在以下各节中，术语“自管理”适用于在本地安装或在 Amazon EC2 上安装的任何数据库。术语“AWS托管”适用于 Amazon RDS、Amazon Aurora 或 Amazon S3 上的任何数据库。

有关使用与 MySQL 兼容的数据库和的更多详细信息 AWS DMS，请参阅以下章节。

## 主题

- [使用 AWS DMS从 MySQL 迁移到 MySQL。](#)
- [使用任何与 MySQL 兼容的数据库作为源 AWS DMS](#)
- [使用自管理的 MySQL 兼容数据库作为源 AWS DMS](#)
- [使用 AWS托管的 MySQL 兼容数据库作为源 AWS DMS](#)
- [使用 MySQL 数据库作为源代码的限制 AWS DMS](#)
- [支持 XA 事务](#)
- [使用 MySQL 作为来源时的端点设置 AWS DMS](#)
- [MySQL 的源数据类型](#)

## 使用 AWS DMS从 MySQL 迁移到 MySQL。

对于异构迁移，从MySQL以外的数据库引擎迁移到MySQL数据库，几乎 AWS DMS 总是最好的迁移工具。但是，对于同构迁移，即从 MySQL 数据库迁移到 MySQL 数据库，我们建议您使用同构数据迁移项目。与 AWS DMS相比，同构数据迁移使用原生数据库工具来提供更好的数据迁移性能和准确性。

## 使用任何与 MySQL 兼容的数据库作为源 AWS DMS

在开始使用 MySQL 数据库作为源之前 AWS DMS，请确保满足以下先决条件。这些先决条件适用于自行管理或由管理的 AWS来源。

您必须拥有一个 AWS DMS 具有复制管理员角色的帐户。该角色需要以下权限：

- REPLICATION CLIENT – 仅 CDC 任务需要此权限。换句话说，full-load-only 任务不需要此权限。
- REPLICATION SLAVE – 仅 CDC 任务需要此权限。换句话说，full-load-only 任务不需要此权限。
- SUPER - 仅 MySQL 5.6.6 之前的版本需要此权限。

AWS DMS 用户还必须对指定用于复制的源表具有 SELECT 权限。

## 使用自管理的 MySQL 兼容数据库作为源 AWS DMS

您可以使用以下与 MySQL 兼容的自管理数据库作为 AWS DMS的源：

- MySQL Community Edition
- MySQL Standard Edition
- MySQL Enterprise Edition
- MySQL Cluster Carrier Grade Edition
- MariaDB Community Edition
- MariaDB Enterprise Edition
- MariaDB Column Store

要使用 CDC，请确保启用二进制日志记录。要启用二进制日志记录，则必须在 MySQL 的 `my.ini` (Windows) 或 `my.cnf` (UNIX) 文件中配置以下参数。

参数	值
<code>server_id</code>	将该参数设置为 1 或更大的值。
<code>log-bin</code>	设置二进制日志文件的路径，例如 <code>log-bin=E:\MySQL_Logs\BinLog</code> 。请勿包含文件扩展名。
<code>binlog_format</code>	将该参数设置为 ROW。我们建议在复制期间使用此设置，因为在某些情况下，如果 <code>binlog_format</code> 设置为 STATEMENT，那么在将数据复制到目标时可能会导致不一致。如果 <code>binlog_format</code> 设置为 MIXED，数据库引擎还会向目标写入类似的不一致数据，因为数据库引擎会自动切换到基于 STATEMENT 的日志记录，这可能导致在目标数据库上写入不一致的数据。
<code>expire_logs_days</code>	将该参数设置为 1 或更大的值。为防止过度使用磁盘空间，我们建议您不要使用默认值 0。
<code>binlog_checksum</code>	NONE 对于 DMS 版本 3.4.7 或更早版本，请将此参数设置为。
<code>binlog_row_image</code>	将该参数设置为 FULL。
<code>log_slave_updates</code>	如果您使用 MySQL 或 MariaDB 只读副本作为源，请将此参数设置为 TRUE。

如果您的源使用了 NDB ( 集群式 ) 数据库引擎，则必须配置以下参数以对使用该存储引擎的表启用 CDC。请在 MySQL 的 `my.ini` (Windows) 或 `my.cnf` (UNIX) 文件中添加这些更改。

参数	值
<code>ndb_log_bin</code>	将该参数设置为 ON。此值确保在集群表中进行的更改保存到二进制日志。
<code>ndb_log_u pdate_as_write</code>	将该参数设置为 OFF。此值可以防止将 UPDATE 语句作为 INSERT 语句写入二进制日志。
<code>ndb_log_u pdated_only</code>	将该参数设置为 OFF。此值可确保二进制日志包含整个行而不仅仅是更改的列。

## 使用 AWS 托管的 MySQL 兼容数据库作为源 AWS DMS

您可以使用以下 AWS 托管的 MySQL 兼容数据库作为源：AWS DMS

- MySQL Community Edition
- MariaDB Community Edition
- Amazon Aurora MySQL 兼容版

使用与 AWS 托管的 MySQL 兼容数据库作为源时 AWS DMS，请确保您满足以下使用 CDC 的先决条件：

- 要为 RDS for MySQL 和 RDS for MariaDB 启用二进制日志，请在实例级别启用自动备份。要为 Aurora MySQL 集群启用二进制日志，请更改参数组中的变量 `binlog_format`。

有关设置自动备份的更多信息，请参阅《Amazon RDS 用户指南》中的[使用自动备份](#)。

有关为 Amazon RDS for MySQL 数据库设置二进制日志记录的更多信息，请参阅《Amazon RDS 用户指南》中的[设置二进制日志记录格式](#)。

有关为 Aurora MySQL 集群设置二进制日志记录的更多信息，请参阅[如何为 Amazon Aurora MySQL 集群开启二进制日志记录？](#)。

- 如果您计划使用 CDC，请启用二进制日志记录。有关为 Amazon RDS for MySQL 数据库设置二进制日志记录的更多信息，请参阅《Amazon RDS 用户指南》中的[设置二进制日志记录格式](#)。



- 确保二进制日志可供使用 AWS DMS。由于与 AWS MySQL 兼容的数据库会尽快清除二进制日志，因此您应该延长日志的可用时间。例如，要将日志保留时间增加到 24 小时，您应运行以下命令。

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- 将 `binlog_format` 参数设置为 "ROW"。

#### Note

在 MySQL 或 MariaDB 上，`binlog_format` 是一个动态参数，因此您不必重新启动即可使新值生效。但是，新值仅适用于新会话。如果您出于复制目的将 `binlog_format` 切换为 ROW，则如果这些会话在您更改值之前已启动，则数据库仍可以使用 MIXED 格式创建后续的二进制日志。这可能会妨碍 AWS DMS 正确捕获源数据库上的所有更改。更改 MariaDB 或 MySQL 数据库的 `binlog_format` 设置时，请务必重新启动数据库以关闭所有现有会话，或者重新启动任何执行 DML (Data Manipulation Language, 数据操作语言) 操作的应用程序。将 `binlog_format` 参数更改为 ROW 将强制数据库重新启动所有会话，确保您的数据库使用正确的格式写入所有后续的源数据库更改，以便 AWS DMS 可以正确捕获这些更改。

- 将 `binlog_row_image` 参数设置为 "Full"。
- 对 "NONE" 于 DMS 版本 3.4.7 或更早版本，将该 `binlog_checksum` 参数设置为。有关在 Amazon RDS MySQL 中设置参数的更多信息，请参阅《Amazon RDS 用户指南》中的 [使用自动备份](#)。
- 如果您使用 Amazon RDS MySQL 或 Amazon RDS MariaDB 只读副本作为源，请在只读副本上启用备份，并确保 `log_slave_updates` 参数设置为 TRUE。

## 使用 MySQL 数据库作为源代码的限制 AWS DMS

使用 MySQL 数据库作为源时，请考虑以下事项：

- Amazon RDS MySQL 5.5 或更低版本不支持更改数据捕获 (CDC)。对于 Amazon RDS MySQL，您必须使用版本 5.6、5.7 或 8.0 才能启用 CDC。自管理 MySQL 5.5 源支持 CDC。
- 对于 CDC，CREATE TABLE、ADD COLUMN 和 DROP COLUMN 更改列数据类型，并支持 renaming a column。但是，不支持 DROP TABLE、RENAME TABLE 和对其他属性 (例如列默认值、列可为 null 值、字符集等) 进行更新。

- 对于源上的分区表，当您为目标表准备模式设置为“在目标上删除表”时，将 AWS DMS 创建一个在 MySQL 目标上没有任何分区的简单表。要将分区表迁移到目标上的分区表，请预先在目标 MySQL 数据库创建分区表。
- 不支持使用 `ALTER TABLE table_name ADD COLUMN column_name` 语句将列添加到表的开头 (FIRST) 或中间 (AFTER)。列始终添加到表的末尾。
- 当表名称包含大写和小写字符且源引擎托管于包含不区分大小写的文件名的操作系统上时，CDC 不受支持。一个示例是使用 HFS+ 的 Microsoft Windows 或 OS X。
- 您可以使用兼容 Aurora MySQL 的 Edition Serverless v1 进行满载，但不能将其用于 CDC。这是因为您无法为 MySQL 启用先决条件。有关更多信息，请参阅[参数组和 Aurora Serverless v1](#)。

兼容 Aurora MySQL 的版本无服务器 v2 支持 CDC。

- 列上的 AUTO\_INCREMENT 属性不会迁移到目标数据库列。
- 当二进制日志未存储在标准块存储上时，不支持捕获更改。例如，当二进制日志存储在 Amazon S3 中时，CDC 不会工作。
- AWS DMS 默认使用 InnoDB 存储引擎创建目标表。如果您需要使用 InnoDB 之外的存储引擎，则必须手动创建该表并使用 [Do nothing \(不执行任何操作\)](#) 模式迁移到该表。
- 除非您的 DMS 迁移任务模式为“迁移现有数据 — 仅限满载”，AWS DMS 否则您不能使用 Aurora MySQL 副本作为源。
- 如果与 MySQL 兼容的源在完全加载期间停止，AWS DMS 任务不会停止，但会出现错误。该任务将成功结束，但目标可能与源不同步。如果发生这种情况，请重新启动该任务或重新加载受影响的表。
- 在列值的一部分上创建的索引不会迁移。例如，索引 `CREATE INDEX first_ten_chars ON customer (名称 (10))` 不在目标上创建。
- 在某些情况下，任务配置为不复制 LOB (任务设置中的 SupportLobs “” 为 false 或者在任务控制台中选择了“不包含 LOB 列”)。在这些情况下，AWS DMS 不会将任何 MEDIUMBLOB、LONGBLOB、MEDIUMTEXT 和 LONGTEXT 列迁移到目标。

BLOB、TINYBLOB、TEXT 和 TINYTEXT 列不会受到影响且将迁移到目标。

- MariaDB 源数据库和目标数据库不支持临时数据表或系统版本控制表。
- 如果在两个 Amazon RDS Aurora MySQL 集群之间迁移，则 RDS Aurora MySQL 源端点必须是读/写实例，而不是副本实例。
- AWS DMS 目前不支持 MariaDB 的视图迁移。
- AWS DMS 不支持 MySQL 分区表的 DDL 更改。要跳过 CDC 期间因分区 DDL 更改而暂停表的情况，请将 `skipTableSuspensionForPartitionDdl` 设置为 true。

- AWS DMS 仅支持 3.5.0 及更高版本中的 XA 事务。以前的版本不支持 XA 事务。AWS DMS 不支持 MariaDB 版本 10.6 中的 XA 事务。有关更多信息，请参阅下面的[the section called “支持 XA 事务”](#)。
- AWS DMS 不使用 GTID 进行复制，即使源数据包含 GTID 也是如此。
- AWS DMS 不支持二进制日志事务压缩。
- AWS DMS 不会在使用 InnoDB 存储引擎的 MySQL 数据库的“删除级联”和“更新时级联”事件进行传播。对于这些事件，MySQL 不会生成二进制日志事件来反映子表上的级联操作。因此，AWS DMS 无法将相应的更改复制到子表。有关更多信息，请参阅[索引、外键或级联更新内容或删除内容未迁移](#)。
- AWS DMS 不会捕获对计算 ( VIRTUAL和GENERATED ALWAYS ) 列的更改。要解决此限制，请执行以下操作之一：
  - 在目标数据库中预先创建目标表，然后使用 DO\_NOTHING 或 TRUNCATE\_BEFORE\_LOAD 完全加载任务设置创建 AWS DMS 任务。
  - 添加转换规则以从任务范围中移除计算列。有关转换规则的信息，请参阅[转换规则和操作](#)。

## 支持 XA 事务

扩展架构 ( XA ) 事务用于将来自多个事务资源的一系列操作，分组为单个可靠的全局事务。XA 事务使用两阶段提交协议。通常，在有未处理的 XA 事务时捕获更改可能会导致数据丢失。如果您的数据库不使用 XA 事务，则可以使用默认值 TRUE 忽略此权限和配置 IgnoreOpenXaTransactionsCheck。要开始从具有 XA 事务的源进行复制，请执行以下操作：

- 确保终端 AWS DMS 端用户具有以下权限：

```
grant XA_RECOVER_ADMIN on *.* to 'userName'@'%';
```

- 将端点设置 IgnoreOpenXaTransactionsCheck 设置为 false。

### Note

AWS DMS 不支持 MariaDB 源数据库 10.6 版本上的 XA 事务。

## 使用 MySQL 作为来源时的端点设置 AWS DMS

您可以使用端点设置来配置 MySQL 源数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建源端点时指定设置，或者使用中带有 `--my-sql-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` [AWS CLI](#) 命令来指定设置。

下表显示了将 MySQL 用作源时您可以使用的端点设置。

名称	描述
EventsPollInterval	<p>指定在数据库处于空闲状态时在二进制日志中查看新更改/事件的频率。</p> <p>默认值：5</p> <p>有效值：1-60</p> <p>例如：<code>--my-sql-settings '{"EventsPollInterval": 5}'</code></p> <p>在示例中，每五秒钟 AWS DMS 检查一次二进制日志中的更改。</p>
ExecuteTimeout	<p>对于 3.4.7 及更高 AWS DMS 版本，设置 MySQL 源端点的客户端语句超时时间，以秒为单位。</p> <p>默认值：60</p> <p>例如：<code>--my-sql-settings '{"ExecuteTimeout": 1500}'</code></p>
ServerTimezone	<p>指定源 MySQL 数据库的时区。</p> <p>例如：<code>--my-sql-settings '{"ServerTimezone": "US/Pacific"}</code></p>
AfterConnectScript	<p>指定要在 AWS DMS 连接到端点后立即运行的脚本。无论 SQL 语句是否成功，迁移任务都会继续运行。</p> <p>有效值：一个或多个有效的 SQL 语句 (用分号分隔)。</p>

名称	描述
CleanSrcMetadataOnMismatch	<p>例如：<code>--my-sql-settings '{"AfterConnectScript": "ALTER SESSION SET CURRENT_SCHEMA=system"}'</code></p> <p>出现不匹配情况时在复制实例上清除并重新创建表元数据信息。例如，对表运行更改 DDL 可能会产生有关在复制实例中缓存的表的不同信息。布尔值。</p> <p>默认值：<code>false</code></p> <p>例如：<code>--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'</code></p>
skipTableSuspensionForPartitionDdl	<p>AWS DMS 不支持 MySQL 分区表的 DDL 更改。对于 3.4.6 及更高 AWS DMS 版本，将此设置为在 CDC 期间更改分区 DDL 时会 <code>true</code> 跳过暂停表。AWS DMS 忽略 <code>partitioned-table-related</code> DDL，并继续处理进一步的二进制日志更改。</p> <p>默认值：<code>false</code></p> <p>例如：<code>--my-sql-settings '{"skipTableSuspensionForPartitionDdl": true}'</code></p>
IgnoreOpenXaTransactionsCheck	<p>对于 3.5.0 及更高 AWS DMS 版本，指定任务在启动时是否应忽略打开的 XA 事务。如果您的源有 XA 事务，则将此项设置为 <code>false</code>。</p> <p>默认值：<code>true</code></p> <p>例如：<code>--my-sql-settings '{"IgnoreOpenXaTransactionsCheck": false}'</code></p>

## MySQL 的源数据类型

下表显示了使用时支持的 MySQL 数据库源数据类型 AWS DMS 以及 AWS DMS 数据类型的默认映射。

有关如何查看目标中映射的数据类型的信息，请参阅有关所使用的目标终端节点的部分。

有关 AWS DMS 数据类型的更多信息，请参见[AWS Database Migration Service 的数据类型](#)。

MySQL 数据类型	AWS DMS 数据类型
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)
LONGBLOB	BLOB
MEDIUMBLOB	BLOB
TINYBLOB	BYTES(255)

MySQL 数据类型	AWS DMS 数据类型
DATE	DATE
DATETIME	DATETIME  不带括号值的 DATETIME 复制时不包括毫秒。括号值为 1 到 5 的 DATETIME ( 例如 DATETIME(5) ) 在复制时包括毫秒。  复制 DATETIME 列时，时间在目标上保持不变。它不会转换为 UTC。
TIME	string
TIMESTAMP	DATETIME  复制 TIMESTAMP 列时，目标上的时间会转换为 UTC。
YEAR	INT2
DOUBLE	REAL8
FLOAT	REAL(DOUBLE)  如果 FLOAT 值不在以下范围内，请使用转换将 FLOAT 映射到 STRING。有关转换的更多信息，请参阅 <a href="#">转换规则和操作</a> 。  支持的 FLOAT 范围是 -1.79E+308 到 -2.23E-308、0 以及 2.23E-308 到 1.79E+308
VARCHAR(45)	WSTRING (45)
VARCHAR(2000)	WSTRING (2000)
VARCHAR(4000)	WSTRING (4000)
VARBINARY (4000)	BYTES (4000)

MySQL 数据类型	AWS DMS 数据类型
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 (##) 此处的##是 ENUM 中最长值的长度。
SET	WSTRING (##) 此处的##是 SET 中所有值的总长度，包括逗号。
JSON	CLOB



**Note**

在某些情况下，您可以使用“零”值（即 0000-00-00）来指定 DATETIME 和 TIMESTAMP 数据类型。如果这样做，请确保复制任务中的目标数据库支持 DATETIME 和 TIMESTAMP 数据类型的“零”值。否则，这些值在目标上将被记录为 null。

## 将 SAP ASE 数据库作为 AWS DMS 源

您可以使用 AWS DMS 从 SAP Adaptive Server Enterprise (ASE) 数据库（以前称为 Sybase）中迁移数据。使用 SAP ASE 数据库作为源，您可以将数据迁移到其他受支持的任何 AWS DMS 目标数据库。

有关 AWS DMS 支持作为源的 SAP ASE 版本的信息，请参阅 [的来源 AWS DMS](#)。

有关使用 SAP ASE 数据库和 AWS DMS 的其他详细信息，请参阅以下章节。

### 主题

- [将 SAP ASE 数据库作为 AWS DMS 源的先决条件](#)
- [将 SAP ASE 作为 AWS DMS 源的限制](#)
- [将 SAP ASE 作为 AWS DMS 源所需的权限](#)
- [删除截断点](#)
- [使用 SAP ASE 作为 AWS DMS 源时的端点设置](#)
- [SAP ASE 的源数据类型](#)

## 将 SAP ASE 数据库作为 AWS DMS 源的先决条件

要将 SAP ASE 数据库作为 AWS DMS 源，请执行以下操作：

- 使用 `sp_setreptable` 命令为表启用 SAP ASE 复制。有关更多信息，请参阅 [Sybase Infocenter 存档](#)。
- 在 SAP ASE 数据库上禁用 RepAgent。有关更多信息，请参阅 [在主数据库中停止和禁用 RepAgent 线程](#)。
- 要复制到为非拉丁字符（例如，中文）配置的 Windows EC2 实例中的 SAP ASE 15.7 版本，请在目标计算机上安装 SAP ASE 15.7 SP121。

**Note**

对于持续的更改数据捕获 (CDC) 复制，DMS 会运行 `dbcc logtransfer` 和 `dbcc log`，以从事务日志中读取数据。

## 将 SAP ASE 作为 AWS DMS 源的限制

将 SAP ASE 数据库作为 AWS DMS 源时，存在以下限制：

- 对于每个 SAP ASE 数据库，您只能运行一个带有持续复制或 CDC 的 AWS DMS 任务。您可以并行运行多个仅完全加载的任务。
- 您不能重命名表。例如，下面的命令将失败。

```
sp_rename 'Sales.SalesRegion', 'SalesReg;
```

- 您不能重命名列。例如，下面的命令将失败。

```
sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';
```

- 在复制到目标数据库时，将截断位于二进制数据类型字符串末尾的零值。例如，源表中的 `0x00000000000000000000000000000001000001000000000` 在目标表中变为 `0x00000000000000000000000000000001000001`。
- 如果数据库默认设置为不允许 NULL 值，则 AWS DMS 使用不允许 NULL 值的列创建目标表。因此，如果完整加载或 CDC 复制任务包含空值，则 AWS DMS 将引发错误。您可以执行以下命令以允许在源数据库中包含 NULL 值，从而防止出现这些错误。

```
sp_dboption database_name, 'allow nulls by default', 'true'  
go  
use database_name  
CHECKPOINT  
go
```

- 不支持 `reorg rebuild` 索引命令。
- AWS DMS 不支持集群或使用 MSA (多站点可用性) /温备用作为源。
- 在映射规则中使用 `AR_H_TIMESTAMP` 转换标头表达式时，不会捕获已添加列的毫秒数。
- 在 CDC 期间运行合并操作将导致不可恢复的错误。要使目标恢复同步，请运行完全加载。
- 使用数据行锁定方案的表不支持回滚触发器事件。

- 从源 SAP 数据库中删除任务范围内的表后，AWS DMS 无法恢复复制任务。如果 DMS 复制任务已停止并执行了任何 DML 操作（INSERT、UPDATE、DELETE），然后删除了表，则必须重新启动复制任务。

## 将 SAP ASE 作为 AWS DMS 源所需的权限

要在 AWS DMS 任务中使用 SAP ASE 数据库作为源，您需要授予权限。为 AWS DMS 数据库定义中指定的用户帐户授予 SAP ASE 数据库中的以下权限：

- sa\_role
- replication\_role
- sybase\_ts\_role
- 默认情况下，如果需要拥有运行 sp\_setreptable 存储过程的权限，AWS DMS 会启用 SAP ASE 复制选项。如果要直接从数据库端点而不是通过 AWS DMS 自身在表上运行 sp\_setreptable，则可以使用 enableReplication 额外连接属性。有关更多信息，请参阅[使用 SAP ASE 作为 AWS DMS 源时的端点设置](#)。

## 删除截断点

当任务开始后，AWS DMS 在 syslogshold 系统视图中建立 \$replication\_truncation\_point 条目，指示复制过程正在进行。在 AWS DMS 工作的同时，它会根据已经复制到目标的数据量，按固定间隔放置复制截断点。

建立 \$replication\_truncation\_point 条目之后，AWS DMS 使任务保持运行状态，以防止数据库日志变得过大。如果要永久停止 AWS DMS 任务，请通过发布以下命令来删除复制截断点：

```
dbcc settrunc('ltm','ignore')
```

删除截断点之后，将无法继续 AWS DMS 任务。日志将继续在检查点自动截断（在设置了自动截断时）。

## 使用 SAP ASE 作为 AWS DMS 源时的端点设置

您可以使用端点设置来配置 SAP ASE 源数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建源端点时指定端点设置，或者在 [AWS CLI](#) 中使用符合 `--sybase-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令指定端点设置。

下表显示了将 SAP ASE 用作源时您可以使用的端点设置。

名称	描述
Charset	<p>将此属性设置为与国际字符集相对应的 SAP ASE 名称。</p> <p>默认值：iso_1</p> <p>示例：<code>--sybase-settings '{"Charset": "utf8"}'</code></p> <p>有效值：</p> <ul style="list-style-type: none"><li>• acsii_8</li><li>• big5hk</li><li>• cp437</li><li>• cp850</li><li>• cp852</li><li>• cp852</li><li>• cp855</li><li>• cp857</li><li>• cp858</li><li>• cp860</li><li>• cp864</li><li>• cp866</li><li>• cp869</li><li>• cp874</li><li>• cp932</li><li>• cp936</li><li>• cp950</li><li>• cp1250</li><li>• cp1251</li><li>• cp1252</li><li>• cp1253</li></ul>

名称	描述
	<ul style="list-style-type: none"><li>• cp1254</li><li>• cp1255</li><li>• cp1256</li><li>• cp1257</li><li>• cp1258</li><li>• deckanji</li><li>• euccns</li><li>• eucgb</li><li>• eucjis</li><li>• eucksc</li><li>• gb18030</li><li>• greek8</li><li>• iso_1</li><li>• iso88592</li><li>• iso88595</li><li>• iso88596</li><li>• iso88597</li><li>• iso88598</li><li>• iso88599</li><li>• iso15</li><li>• kz1048</li><li>• koi8</li><li>• roman8</li><li>• iso88599</li><li>• sjis</li><li>• tis620</li><li>• turkish8</li><li>• utf8</li></ul>

名称	描述
	<p>有关 SAP ASE 数据库中支持的字符集的任何其他问题，请参阅 <a href="#">Adaptive Server Enterprise：支持的字符集</a>。</p>
EnableReplication	<p>如果要从数据库端而不是通过 AWS DMS 在表上启用 <code>sp_setreptable</code>，请设置此属性。</p> <p>默认值：<code>true</code></p> <p>有效值：<code>true</code> 或 <code>false</code></p> <p>示例：<code>--sybase-settings '{"Enable Replication": false}'</code></p>
EncryptPassword	<p>如果您在源数据库中启用了 <code>"net password encryption reqd"</code>，请设置此属性。</p> <p>默认值：<code>0</code></p> <p>有效值：<code>0</code>、<code>1</code> 或 <code>2</code></p> <p>示例：<code>--sybase-settings '{"EncryptPassword": 1}'</code></p> <p>有关这些参数值的更多信息，请参阅 <a href="#">Adaptive Server Enterprise：使用 EncryptPassword 连接字符串属性</a>。</p>
Provider	<p>如果要为 ASE 15.7 及更高版本使用传输层安全性协议 (TLS) 1.2，请设置此属性。请注意，AWS 需要 TLS 1.2 或更高版本，建议使用 1.3 版本。</p> <p>默认值：<code>Adaptive Server Enterprise</code></p> <p>有效值：<code>Adaptive Server Enterprise 16.03.06</code></p> <p>示例：<code>--sybase-settings '{"Provider": "Adaptive Server Enterprise 16.03.06}"</code></p>

## SAP ASE 的源数据类型

有关使用 AWS DMS 时支持的 SAP ASE 源数据类型以及来自 AWS DMS 数据类型的默认映射的列表，请参阅下表。AWS DMS 不支持具有用户定义的类型 (UDT) 数据类型的列的 SAP ASE 源表。具有此数据类型的复制列将创建为 NULL。

有关如何查看目标中映射的数据类型的信息，请参阅[数据迁移的目标](#)有关您的目标终端节点的部分。

有关 AWS DMS 数据类型的其他信息，请参阅[AWS Database Migration Service 的数据类型](#)。

SAP ASE 数据类型	AWS DMS 数据类型
BIGINT	INT8
UNSIGNED BIGINT	UINT8
INT	INT4
UNSIGNED INT	UINT4
SMALLINT	INT2
UNSIGNED SMALLINT	UINT2
TINYINT	UINT1
DECIMAL	NUMERIC
NUMERIC	NUMERIC
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
MONEY	NUMERIC
SMALLMONEY	NUMERIC
DATETIME	DATETIME

SAP ASE 数据类型	AWS DMS 数据类型
BIGDATETIME	DATETIME(6)
SMALLDATETIME	DATETIME
DATE	DATE
TIME	TIME
BIGTIME	TIME
CHAR	STRING
UNICHAR	WSTRING
NCHAR	WSTRING
VARCHAR	STRING
UNIVARCHAR	WSTRING
NVARCHAR	WSTRING
BINARY	BYTES
VARBINARY	BYTES
BIT	BOOLEAN
TEXT	CLOB
UNITEXT	NCLOB
IMAGE	BLOB

## 将 MongoDB 作为 AWS DMS 源

有关 AWS DMS 支持作为源的 MongoDB 版本的信息，请参阅 [的来源 AWS DMS](#)。

请注意以下有关 MongoDB 版本支持的信息：



- AWS DMS 版本 3.4.5 及更高版本支持 MongoDB 版本 4.2 和 4.4。
- AWS DMS 版本 3.4.5 及更高版本支持 MongoDB 4.2 及更高版本和分布式事务。有关 MongoDB 分布式事务的更多信息，请参阅 [MongoDB 文档](#) 中的 [事务](#)。
- AWS DMS 版本 3.5.0 及更高版本不支持 3.6 版本之前的 MongoDB。
- AWS DMS 版本 3.5.1 及更高版本支持 MongoDB 版本 5.0。
- AWS DMS 版本 3.5.2 及更高版本支持 MongoDB 版本 6.0。

如果您是初次使用 MongoDB，请注意以下关于 MongoDB 数据库的重要概念：

- MongoDB 中的记录是一个文档，它是由字段和值对构成的数据结构。字段值可以包含其他文档、数组和文档数组。文档大致相当于关系数据库表中的行。
- MongoDB 中的集合是一组文档，并且大致相当于关系数据库表。
- MongoDB 中的数据库是一组集合，并且大致相当于关系数据库中的架构。
- 在内部，MongoDB 文档以压缩格式存储为二进制 JSON (BSON) 文件，包含文档中每个字段的类型。每个文档都有唯一的 ID。

将 MongoDB 作为源时，AWS DMS 支持两种迁移模式：Document 模式或表模式。您可以在创建 MongoDB 端点时指定要使用的迁移模式，或者通过从 AWS DMS 控制台设置元数据模式参数来指定。或者，您可以创建名为 `_id` 的第二列，通过在端点配置面板中选中将 `_id` 作为单独的列复选标记，将其作为主键。

您选择的迁移模式将影响目标数据的结果格式，如下所述。

### 文档模式

在文档模式下，MongoDB 文档按“原样”迁移，这意味着文档数据将并入目标表中一个名为 `_doc` 的列中。文档模式是您将 MongoDB 作为源终端节点时的默认设置。

例如，请考虑名为 `myCollection` 的 MongoDB 集合中的以下文档。

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

在使用文档模式将数据迁移到关系数据库表后，数据结构如下所示。MongoDB 文档中的数据字段将并入 `_doc` 列。

oid_id	_doc
5a94815f40bd44d1b02bdfe0	{ "a" : 1, "b" : 2, "c" : 3 }
5a94815f40bd44d1b02bdfe1	{ "a" : 4, "b" : 5, "c" : 6 }

您可以选择将额外连接属性 `extractDocID` 设置为 `true`，以创建第二个名为 `"_id"` 的列作为主键。如果要使用 CDC，请将此参数设置为 `true`。

在文档模式中，AWS DMS 按如下方式管理集合的创建和重命名：

- 如果您将一个新集合添加到源数据库，则 AWS DMS 将为该集合创建一个新的目标表并复制所有文档。
- 如果您重命名源数据库上的现有集合，AWS DMS 不会重命名目标表。

如果目标端点是 Amazon DocumentDB，请在文档模式下运行迁移。

## 表模式

在表模式中，AWS DMS 将 MongoDB 文档中的每个顶级字段转换为目标表中的一个列。如果已嵌套字段，则 AWS DMS 会将嵌套值平展到单个列中。随后，AWS DMS 将关键字段和数据类型添加到目标表的列集。

对于每个 MongoDB 文档，AWS DMS 将每个键和类型添加到目标表的列集中。例如，通过使用表模式，AWS DMS 将上一个示例迁移到下表中。

oid_id	a	b	c
5a94815f40bd44d1b02bdfe0	1	2	3
5a94815f40bd44d1b02bdfe1	4	5	6

嵌套值平展到包含键名 (以句点分隔) 的列中。该列的名称是由句点分隔的平展字段名的联接。例如，AWS DMS 将具有嵌套值字段 (如 `{"a" : {"b" : {"c" : 1}}}`) 的 JSON 文档迁移到名为 `a.b.c.` 的列中

为了创建目标列，AWS DMS 将扫描指定数量的 MongoDB 文档并创建包含所有字段及其类型的集。随后，AWS DMS 使用此集创建目标表的列。如果您使用控制台创建或修改 MongoDB 源终端节点，则可指定要扫描的文档的数量。默认值为 1000 个文档。如果您使用 AWS CLI，则可使用额外连接属性 `docsToInvestigate`。

在表模式中，AWS DMS 按如下方式管理文档和集合：

- 当您将一个文档添加到现有集合时，将复制该文档。如果某些字段在目标中不存在，则不会复制这些字段。
- 当您更新文档时，复制更新后的文档。如果某些字段在目标中不存在，则不会复制这些字段。
- 完全支持文档删除。
- 在 CDC 任务期间，添加新集合不会导致在目标上生成新的表。
- 在更改数据捕获 (CDC) 阶段，AWS DMS 不支持对集合重命名。

## 主题

- [将 MongoDB 作为 AWS DMS 的源时所需的权限](#)
- [为 CDC 配置 MongoDB 副本集](#)
- [将 MongoDB 作为 AWS DMS 源时的安全要求](#)
- [对 MongoDB 集合进行分段并以并行方式迁移](#)
- [使用 MongoDB 作为 AWS DMS 的源时迁移多个数据库](#)
- [将 MongoDB 作为 AWS DMS 源时的限制](#)
- [使用 MongoDB 作为 AWS DMS 的源时的端点配置设置](#)
- [MongoDB 的源数据类型](#)

## 将 MongoDB 作为 AWS DMS 的源时所需的权限

对于使用 MongoDB 源的 AWS DMS 迁移，您可以创建具有根权限的用户账户，也可以仅在要迁移的数据库上创建具有权限的用户。

以下代码创建将作为根账户的用户。

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
```

```
}  
)
```

对于 MongoDB 3.x 源，以下代码在要迁移的数据库上创建具有最低权限的用户。

```
use database_to_migrate  
db.createUser(  
  {  
    user: "dms-user",  
    pwd: "password",  
    roles: [ { role: "read", db: "local" }, "read"]  
  })
```

对于 MongoDB 4.x 源，以下代码创建具有最低权限的用户。

```
{ resource: { db: "", collection: "" }, actions: [ "find", "changeStream" ] }
```

例如，在“admin”数据库中创建以下角色。

```
use admin  
db.createRole(  
  {  
    role: "changestreamrole",  
    privileges: [  
      { resource: { db: "", collection: "" }, actions: [ "find","changeStream" ] }  
    ],  
    roles: []  
  }  
)
```

创建角色后，在要迁移的数据库中创建用户。

```
> use test  
> db.createUser(  
  {  
    user: "dms-user12345",  
    pwd: "password",  
    roles: [ { role: "changestreamrole", db: "admin" }, "read"]  
  })
```

## 为 CDC 配置 MongoDB 副本集

要将持续复制或 CDC 用于 MongoDB，AWS DMS 需要对 MongoDB 操作日志 (oplog) 的访问权限。要创建 oplog，您需要部署一个副本集 (如果没有副本集)。有关更多信息，请参阅 [MongoDB 文档](#)。

您可以将 CDC 用于作为源终端节点的 MongoDB 副本集的主要或次要节点。

### 将独立实例转换为副本集

1. 使用命令行，连接到 mongo。

```
mongo localhost
```

2. 停止 mongod 服务。

```
service mongod stop
```

3. 使用以下命令重新启动 mongod：

```
mongod --replSet "rs0" --auth -port port_number
```

4. 使用以下命令测试与副本集的连接：

```
mongo -u root -p password --host rs0/localhost:port_number  
--authenticationDatabase "admin"
```

如果您打算执行文档模式迁移，请在创建 MongoDB 终端节点时选择选项 `_id as a separate column`。通过选择此选项，将创建另一个名为 `_id` 的列以作为主键。AWS DMS 需要第二列以支持数据操作语言 (DML) 操作。

#### Note

AWS DMS 使用操作日志 (oplog) 捕获持续复制期间的更改。如果 MongoDB 在 AWS DMS 读取记录之前将其从 oplog 中清除，则您的任务将失败。我们建议调整 oplog 的大小，将更改至少保留 24 小时。

## 将 MongoDB 作为 AWS DMS 源时的安全要求

AWS DMS 支持 MongoDB 的两种身份验证方法。这两种身份验证方法用于加密密码，因此它们仅在将 `authType` 参数设置为 `PASSWORD` 时使用。

MongoDB 身份验证方法如下：

- MONGODB-CR – 提供向后兼容性
- SCRAM-SHA-1 – 使用 MongoDB 版本 3.x 和 4.0 时的默认值

如果未指定身份验证方法，AWS DMS 将使用 MongoDB 源版本的默认方法。

## 对 MongoDB 集合进行分段并以并行方式迁移

为了提高迁移任务的性能，MongoDB 源端点支持表映射中的并行完全加载的两个选项。

换句话说，在 JSON 设置中，对于并行完全加载，您可以使用表映射的自动分段或范围分段来迁移集合。使用自动分段，您可以为 AWS DMS 指定标准，以自动在每个线程中对源分段用于迁移。使用范围分段，您可以告知 AWS DMS 每个区段的特定范围，以便 DMS 在每个线程中进行迁移。有关这些设置的更多信息，请参阅[表和集合设置规则和操作](#)。

### 使用自动分段范围并行迁移 MongoDB 数据库

您可以并行迁移文档，方法是为 AWS DMS 指定标准，以便为每个线程自动分区（分段）数据。具体而言，您指定每个线程要迁移的文档数量。使用这种方法，AWS DMS 尝试优化区段边界，以最大限度地提高每个线程的性能。

您可以在表映射中，使用以下表设置选项来指定分段标准：

表设置选项	描述
"type"	(必需) 对于将 MongoDB 作为源，设置为 "partitions-auto"。
"number-of-partitions"	(可选) 迁移使用的分区（区段）总数。默认值是 16。
"collection-count-from-meta-data"	(可选) 如果此选项设置为 true，则 AWS DMS 使用估计的集合并数来确定分区数。如果此选项

表设置选项	描述
	<p>设置为 <code>false</code>，则 AWS DMS 使用实际的集计数。默认值为 <code>true</code>。</p>
<code>"max-records-skip-per-page"</code>	<p>( 可选 ) 确定每个分区的边界时，要同时跳过的记录数。AWS DMS 采用分页跳过方法来确定分区的最小边界。默认值是 10000。</p> <p>设置一个相对较大的值，会导致游标超时和任务失败。设置一个相对较低的值，会导致每页的操作次数增加，完全加载速度降低。</p>
<code>"batch-size"</code>	<p>( 可选 ) 限制一个批处理返回的文档数量。每个批处理需要往返一次服务器。如果批处理大小为零 (0)，则游标使用服务器定义的最大批处理大小。默认值是 0。</p>

以下示例显示了自动分段的表映射。

```
{
  "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
    }
  ]
}
```

自动分段有以下限制。每个区段的迁移分别提取集合计数和集合的最小 `_id`。然后，它使用分页跳过来计算该区段的最小边界。

因此，请确保每个集合的最小 `_id` 值保持不变，直到计算出集合中的所有区段边界。如果在区段边界计算期间更改集合的最小 `_id` 值，则会导致数据丢失或重复行错误。

### 使用范围分段并行迁移 MongoDB 数据库

您可以通过为线程中的每个区段指定范围来并行迁移文档。使用这种方法，您可以根据为每个线程选择的文档范围，告知 AWS DMS 要在每个线程中迁移的特定文档。

下图显示了包含七个项目并将 `_id` 作为主键的 MongoDB 集合。



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

要将集合拆分为三个特定区段供 AWS DMS 并行迁移，您可以在迁移任务中添加表映射规则。此方法如以下 JSON 示例所示。

```
{ // 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

```

该表映射定义将源集合拆分为三个区段以并行迁移。以下是分段边界。

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)
```

迁移任务完成后，您可以从任务日志中验证表是否并行加载，如以下示例所示。您还可以验证用于从源表中卸载每个区段的 MongoDB find 子句。

```
[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.
```

目前，AWS DMS 支持将以下 MongoDB 数据类型作为区段键列：

- Double
- String
- ObjectId
- 32 位整数
- 64 位整数

## 使用 MongoDB 作为 AWS DMS 的源时迁移多个数据库

对于所有支持的 MongoDB 版本，AWS DMS 版本 3.4.5 及更高版本支持在单个任务中迁移多个数据库。如果您要迁移多个数据库，请采取以下步骤：

1. 在创建 MongoDB 源端点时，请执行下列操作之一：
  - 在 DMS 控制台的创建端点页面上，确保端点配置下的数据库名称为空。
  - 使用 AWS CLI `CreateEndpoint` 命令，向 `MongoDBSettings` 中的 `DatabaseName` 参数分配一个空字符串值。
2. 对于要从 MongoDB 源迁移的每个数据库，请在任务的表映射中指定数据库名称作为架构名称。您可以在控制台中使用引导式输入或直接在 JSON 中执行此操作。有关引导式输入的更多信息，请参阅 [通过控制台指定表选择和转换规则](#)。有关 JSON 的更多信息，请参阅 [选择规则和操作](#)。

例如，您可以指定以下 JSON 以迁移三个 MongoDB 数据库。

Example 迁移架构中的所有表

以下 JSON 会将源端点中 `Customers`、`Orders` 和 `Suppliers` 数据库的所有表迁移到目标端点。

```
{
  "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": []
  }
]
}

```

## 将 MongoDB 作为 AWS DMS 源时的限制

将 MongoDB 作为 AWS DMS 源时，存在以下限制：

- 在表模式下，集合中的文档的数据类型，必须与所用于的同一字段中值的数据类型一致。例如，如果集合中的文档包含 '{ a:{ b:value ... }'，则集合中引用 a.b 字段 *value* 的所有文档都必须为 *value* 使用相同的数据类型，无论该文档出现在集合中的哪个位置。
- 当 `_id` 选项设置为单独一列时，ID 字符串不能超过 200 个字符。
- 在表模式下，对象 ID 和数组类型键将转换为具有 `oid` 和 `array` 前缀的列。

将使用具有前缀的名称在内部引用这些列。如果您在 AWS DMS 中使用引用了这些列的转换规则，请确保指定具有前缀的列。例如，指定 `#{oid__id}` 而不是 `#{_id}`，或者指定 `#{array__addresses}` 而不是 `#{_addresses}`。

- 集合名称和键名不能包含美元符号 (\$)。
- 在使用 RDBMS 的表模式下，AWS DMS 不支持包含具有不同大小写 (大写、小写) 的相同字段的集合。例如，AWS DMS 不支持拥有两个名为 Field1 和 field1 的集合。
- 表模式和文档模式具有前面所述的限制。
- 使用自动分段进行并行迁移时，存在上述限制。
- MongoDB 不支持源筛选器。
- AWS DMS 不支持嵌套级别大于 97 的文档。
- AWS DMS 不支持 MongoDB 5.0 版本的以下功能：
  - 实时重新分片
  - 客户端字段级加密 (CSFLE)
  - 时间序列集合迁移

#### Note

由于 DocumentDB 不支持时间序列集合，因此在完全加载阶段中迁移的时间序列集合将转换为 Amazon DocumentDB 中的普通集合。

## 使用 MongoDB 作为 AWS DMS 的源时的端点配置设置

在设置 MongoDB 源端点时，您可以使用 AWS DMS 控制台指定多个端点配置设置。

下表介绍了将 MongoDB 数据库作为 AWS DMS 的源时的可用配置设置。

设置 (属性)	有效值	默认值和描述
身份验证模式	"none" "password"	值 "password" 提示输入用户名和密码。如果指定 "none"，则不使用用户名和密码参数。
身份验证源	有效的 MongoDB 数据库名称。	您为身份验证所用凭证进行验证时使用的 MongoDB 数据库的名称。默认值为 "admin"。
身份验证机制	"default" "mongodb_cr"	身份验证机制。"default" 值为 "scram_sha_1"。当 authType 设为 "no" 时，不使用此设置。

设置 ( 属性 )	有效值	默认值和描述
	"scram_sha_1"	
元数据模式	文档和表	选择文档模式或表模式。
要扫描的文档数 (docsToInvestigate)	大于 0 的正整数。	仅在表模式下使用此选项来定义目标表定义。
将 _id 作为单独的列	复选框中的选中标记	可选的复选标记框，用于创建另一个名为 _id 的列作为主键。
socketTimeoutMS	NUMBER 仅限额外连接属性 (ECA)。	此设置以毫秒为单位，用于配置 MongoDB 客户端的连接超时。如果该值小于或等于零，则使用 MongoDB 客户端默认值。
UseUpdateLookUp	布尔值 true false	如果为 true，则在 CDC 更新事件期间，AWS DMS 将整个更新的文档复制到目标。设置为 false 时，AWS DMS 使用 MongoDB 更新命令，仅更新目标上文档中已修改的字段。
ReplicateShardCollections	布尔值 true false	<p>如果为 true，AWS DMS 将数据复制到分片集合。仅当目标端点是 DocumentDB 弹性集群时，AWS DMS 才使用此设置。</p> <p>当此项设置为 true 时，请注意以下几点：</p> <ul style="list-style-type: none"> <li>• 必须将 TargetTablePrepMode 设置为 nothing。</li> <li>• AWS DMS 自动将 useUpdateLookup 设置为 false。</li> </ul>

如果您选择文档作为元数据模式，则可以使用不同的选项。

如果目标端点是 DocumentDB，请确保在文档模式下运行迁移。此外，修改您的源端点并选择选项将 \_id 作为单独的列。如果您的源 MongoDB 工作负载涉及事务，则这是必需的先决条件。

## MongoDB 的源数据类型

将 MongoDB 作为 AWS DMS 源的数据迁移支持大多数 MongoDB 数据类型。在下表中，您可以找到使用 AWS DMS 时支持的 MongoDB 源数据类型以及来自 AWS DMS 数据类型的默认映射。有关 MongoDB 数据类型的更多信息，请参阅 MongoDB 文档中的 [BSON 类型](#)。

有关如何查看目标中映射的数据类型的信息，请参阅有关所使用的目标终端节点的部分。

有关 AWS DMS 数据类型的其他信息，请参阅 [AWS Database Migration Service 的数据类型](#)。

MongoDB 数据类型	AWS DMS 数据类型
布尔值	布尔型
二元	BLOB
Date	Date
时间戳	Date
Int	INT4
长整型	INT8
Double	REAL8
字符串 (UTF-8)	CLOB
数组	CLOB
OID	String
REGEX	CLOB
CODE	CLOB

## 使用亚马逊 DocumentDB ( 兼容 MongoDB ) 作为来源 AWS DMS

有关 AWS DMS 支持将哪些版本的 Amazon DocumentDB ( 与 MongoDB 兼容 ) 作为源的信息，请参阅 [的来源 AWS DMS](#)。



使用 Amazon DocumentDB 作为源，您可以将数据从一个 Amazon DocumentDB 集群迁移到另一个 Amazon DocumentDB 集群。您也可以将数据从 Amazon DocumentDB 集群迁移到支持的其他目标终端节点之一。AWS DMS

如果您不熟悉 Amazon DocumentDB，请注意 Amazon DocumentDB 的以下重要概念：

- Amazon DocumentDB 中的记录是一个文档，它是由字段和值对构成的数据结构。字段值可以包含其他文档、数组和文档数组。文档大致相当于关系数据库表中的行。
- Amazon DocumentDB 中的集合是一组文档，并且大致相当于关系数据库表。
- Amazon DocumentDB 中的数据库是一组集合，并且大致相当于关系数据库中的架构。

AWS DMS 使用 Amazon DocumentDB 作为源代码时，支持两种迁移模式：文档模式和表格模式。在 AWS DMS 控制台中创建 Amazon DocumentDB 源终端节点时，您可以使用元数据模式选项或额外连接属性来指定迁移模式。nestingLevel在下文中，您可以找到有关迁移模式的选择如何影响所生成的目标数据格式的说明。

## 文档模式

在文档模式下，JSON 文档按原样迁移。这意味着文档数据将合并到两个项目之一中。当您使用关系数据库作为目标时，数据是目标表中名为 `_doc` 的单个列。当您使用非关系数据库作为目标时，数据是单个 JSON 文档。默认模式为文档模式，我们建议在迁移到 Amazon DocumentDB 目标时使用这种模式。

例如，请考虑名为 `myCollection` 的 Amazon DocumentDB 集合中的以下文档。

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

在使用文档模式将数据迁移到关系数据库表后，数据结构如下所示。文档中的数据字段将整合到 `_doc` 列中。

oid_id	_doc
5a94815f40bd44d1b02bdfe0	{ "a" : 1, "b" : 2, "c" : 3 }
5a94815f40bd44d1b02bdfe1	{ "a" : 4, "b" : 5, "c" : 6 }

您可以选择将额外连接属性 `extractDocID` 设置为 `true`，以创建第二个名为 `"_id"` 的列作为主键。如果您要使用更改数据捕获 (CDC)，请将此参数设置为 `true`，除非使用 Amazon DocumentDB 作为目标。

### Note

如果向源数据库添加新集合，则会为该集合 AWS DMS 创建一个新的目标表并复制所有文档。

## 表模式

在表模式中，AWS DMS 将 Amazon DocumentDB 文档中的每个顶级字段转换为目标表中的一个列。如果字段是嵌套的，则将嵌套值 AWS DMS 展平为一列。AWS DMS 然后将关键字段和数据类型添加到目标表的列集中。

对于每个 Amazon DocumentDB 文档，AWS DMS 将每个键和类型添加到目标表的列集中。例如，使用表格模式，将前面的示例 AWS DMS 迁移到下表中。

oid_id	a	b	c
5a94815f4 0bd44d1b02bdfe0	1	2	3
5a94815f4 0bd44d1b02bdfe1	4	5	6

嵌套值平展到包含键名 (以句点分隔) 的列中。该列通过联接平展字段名 (由句点分隔) 来命名。例如，AWS DMS 将包含嵌套值字段 (例如嵌套值) 的 JSON 文档迁移 `{"a" : {"b" : {"c" : 1}}}` 到名为的列中 `a.b.c`。

要创建目标列，请 AWS DMS 扫描指定数量的 Amazon DocumentDB 文档，然后创建一组包含所有字段及其类型。AWS DMS 然后使用此集合来创建目标表的列。如果您使用控制台创建或修改 Amazon DocumentDB 源端点，则可指定要扫描的文档的数量。默认值为 1000 个文档。如果使用 AWS CLI，则可以使用额外的连接属性 `docsToInvestigate`。

在表格模式下，像这样 AWS DMS 管理文档和集合：

- 当您将一个文档添加到现有集合时，将复制该文档。如果某些字段在目标中不存在，则不会复制这些字段。
- 当您更新文档时，复制更新后的文档。如果某些字段在目标中不存在，则不会复制这些字段。
- 完全支持文档删除。
- 在 CDC 任务期间，添加新集合不会导致在目标上生成新的表。
- 在变更数据捕获 (CDC) 阶段，AWS DMS 不支持重命名集合。

## 主题

- [设置权限以使用 Amazon DocumentDB 作为源](#)
- [为 Amazon DocumentDB 集群配置 CDC](#)
- [使用 TLS 连接到 Amazon DocumentDB](#)
- [创建 Amazon DocumentDB 源端点](#)
- [对 Amazon DocumentDB 集合进行分段并以并行方式迁移](#)
- [使用 Amazon DocumentDB 作为源时迁移多个数据库 AWS DMS](#)
- [使用亚马逊 DocumentDB 作为来源时的限制 AWS DMS](#)
- [在以 Amazon DocumentDB 为源时使用端点设置](#)
- [Amazon DocumentDB 的源数据类型](#)

## 设置权限以使用 Amazon DocumentDB 作为源

使用 Amazon DocumentDB 源代码进行 AWS DMS 迁移时，您可以创建具有根权限的用户账户。或者，您可以创建一个仅具有待迁移数据库权限的用户。

以下代码创建作为根账户的用户。

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  })
```

对于 Amazon DocumentDB 3.6，以下代码创建在要迁移的数据库上具有最低权限的用户。

```
use database_to_migrate
db.createUser(
{
  user: "dms-user",
  pwd: "password",
  roles: [ { role: "read", db: "db_name" }, "read" ]
})
```

对于 Amazon DocumentDB 4.0 及更高版本，AWS DMS 使用部署范围内的更改流。此处，以下代码创建具有最小权限的用户。

```
db.createUser(
{
  user: "dms-user",
  pwd: "password",
  roles: [ { role: "readAnyDatabase", db: "admin" } ]
})
```

## 为 Amazon DocumentDB 集群配置 CDC

要将持续复制或 CDC 与 Amazon DocumentDB 配合使用，AWS DMS 需要访问亚马逊文档数据库集群的变更流。有关集群的集合和数据库中按时间排列的更新事件序列的描述，请参阅《Amazon DocumentDB 开发人员指南》中的[使用更改流](#)。

使用 MongoDB Shell 对您的 Amazon DocumentDB 集群进行身份验证。然后运行以下命令以启用更改流。

```
db.adminCommand({modifyChangeStreams: 1,
  database: "DB_NAME",
  collection: "",
  enable: true});
```

这种方法可以为您数据库中的所有集合启用更改流。启用更改流后，您可以创建一个迁移任务，该任务可以迁移现有数据，同时复制正在进行的更改。AWS DMS 即使加载了批量数据，仍会继续捕获和应用更改。最终，源数据库和目标数据库将保持同步，尽可能减少迁移的停机时间。

**Note**

AWS DMS 使用操作日志 (oplog) 捕获正在进行的复制期间的更改。如果 Amazon DocumentDB 在 AWS DMS 读取操作日志之前将其从操作日志中清除，则您的任务将失败。我们建议调整 oplog 的大小，将更改至少保留 24 小时。

## 使用 TLS 连接到 Amazon DocumentDB

默认情况下，新创建的 Amazon DocumentDB 集群仅接受使用传输层安全性协议 ( TLS ) 的安全连接。当 TLS 启用时，每个到 Amazon DocumentDB 的连接都需要公有密钥。

您可以通过 `rds-combined-ca-bundle.pem` 从托管的 Amazon AWS S3 存储桶下载文件来检索 Amazon DocumentDB 的公钥。有关下载此文件的更多信息，请参阅《Amazon DocumentDB 开发人员指南》中的 [使用 TLS 加密连接](#)。

下载 `rds-combined-ca-bundle.pem` 文件后，可以将其包含的公钥导入到 AWS DMS。以下步骤说明了如何完成此操作。

使用 AWS DMS 控制台导入您的公钥

1. 登录 AWS Management Console 并选择 AWS DMS。
2. 在导航窗格中，选择证书。
3. 选择导入证书。此时显示导入新 CA 证书页面。
4. 在证书配置部分中，执行下列操作之一：
  - 对于证书标识符，请输入证书的唯一名称，例如 `docdb-cert`。
  - 选择选择文件，导航到您保存 `rds-combined-ca-bundle.pem` 文件的位置，然后选择该文件。
5. 选择 Add new CA certificate (添加新 CA 证书)。

AWS CLI 以下示例使用 AWS DMS `import-certificate` 命令导入公钥 `rds-combined-ca-bundle.pem` 文件。

```
aws dms import-certificate \  
  --certificate-identifier docdb-cert \  
  --certificate-pem file:///./rds-combined-ca-bundle.pem
```

## 创建 Amazon DocumentDB 源端点

您可以使用控制台或 AWS CLI 创建 Amazon DocumentDB 源端点。通过控制台使用以下过程。

### 使用控制台配置 Amazon DocumentDB 源终端节点 AWS DMS

1. 登录 AWS Management Console 并选择 AWS DMS。
2. 在导航窗格中，选择端点，然后选择创建端点。
3. 对于端点标识符，请提供一个可以帮助您轻松识别端点的名称，例如 docdb-source。
4. 对于源引擎，请选择 Amazon DocumentDB (与 MongoDB 兼容)。
5. 对于服务器名称，请输入您的 Amazon DocumentDB 数据库端点所在的服务器的名称。例如，您可以输入 Amazon EC2 实例的公有 DNS 名称，如 democluster.cluster-cjf6q8nxfefi.us-east-2.docdb.amazonaws.com。
6. 对于端口，输入 27017。
7. 对于 SSL mode (SSL 模式)，请选择 verify-full。如果您已经在 Amazon DocumentDB 集群上禁用 SSL，则可以跳过此步骤。
8. 对于 CA 证书，请选择 Amazon DocumentDB 证书 rds-combined-ca-bundle.pem。有关添加此证书的说明，请参阅[使用 TLS 连接到 Amazon DocumentDB](#)。
9. 对于数据库名称，请输入要迁移的数据库的名称。

通过 CLI 使用以下过程。

要配置 Amazon DocumentDB 源端点，请使用 AWS CLI

- 运行以下 AWS DMS create-endpoint 命令来配置 Amazon DocumentDB 源终端节点，使用您自己的值替换占位符。

```
aws dms create-endpoint \  
    --endpoint-identifier a_memorable_name \  
    --endpoint-type source \  
    --engine-name docdb \  
    --username value \  
    --password value \  
    --server-name servername_where_database_endpoint_resides \  
    --port 27017 \  
    --
```

```
--database-name name_of_endpoint_database
```

## 对 Amazon DocumentDB 集合进行分段并以并行方式迁移

为了提高迁移任务的性能，Amazon DocumentDB 源端点支持表映射中的并行完全加载功能的两个选项。换句话说，在 JSON 设置中，对于并行完全加载，您可以使用表映射的自动分段或范围分段选项来迁移集合。自动分段选项允许您指定在每个话题中自动分段源 AWS DMS 以进行迁移的标准。范围分割选项允许您告知 DMS 要 AWS DMS 在每个线程中迁移的每个分段的具体范围。有关这些设置的更多信息，请参阅[表和集合设置规则和操作](#)。

### 使用自动分段范围并行迁移 Amazon DocumentDB 数据库

您可以并行迁移文档，方法是 AWS DMS 为每个线程指定自动分区（分段）数据的标准，尤其是每个线程要迁移的文档数量。使用这种方法，AWS DMS 尝试优化区段边界，以最大限度地提高每个线程的性能。

您可以在表映射中使用以下表设置选项来指定分段标准：

表设置选项	描述
"type"	(必需) 对于将 Amazon DocumentDB 作为源，设置为 "partitions-auto"。
"number-of-partitions"	(可选) 迁移使用的分区（区段）总数。默认值是 16。
"collection-count-from-metadata"	(可选) 如果设置为 true，则 AWS DMS 使用估计的收集计数来确定分区数。如果设置为 false，则 AWS DMS 使用实际收集次数。默认值为 true。
"max-records-skip-per-page"	(可选) 在确定每个分区的边界时要同时跳过的记录数。AWS DMS 使用分页跳过方法来确定分区的最小边界。默认值是 10000。设置一个相对较大的值，可能会导致游标超时和任务失败。设置一个相对较低的值，会导致每页的操作次数增加，完全加载速度降低。

表设置选项	描述
"batch-size"	( 可选 ) 限制一个批处理返回的文档数量。每个批处理需要往返一次服务器。如果批处理大小为零 ( 0 ) , 则游标使用服务器定义的最大批处理大小。默认值是 0。

以下示例显示了自动分段的表映射。

```
{
  "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
      }
    }
  ]
}
```



自动分段有以下限制。每个区段的迁移分别提取集合并计数和集合的最小 `_id`。然后，它使用分页跳过来计算该区段的最小边界。因此，请确保每个集合的最小 `_id` 值保持不变，直到计算出集合中的所有区段边界。如果在区段边界计算期间更改集合的最小 `_id` 值，则可能会导致数据丢失或重复行错误。

## 使用特定区段范围并行迁移 Amazon DocumentDB 数据库

以下示例显示了包含七个项目并将 `_id` 作为主键的 Amazon DocumentDB 集合。

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

要将集合拆分为三个区段以并行迁移，可以在迁移任务中添加表映射规则，如以下 JSON 示例所示。

```
{ // 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

```

该表映射定义将源集合拆分为三个区段以并行迁移。以下是分段边界。

```
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)
```

迁移任务完成后，您可以从任务日志中验证表是否并行加载，如以下示例所示。您还可以验证用于从源表中卸载每个区段的 Amazon DocumentDB find 子句。

```
[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.
```

目前，AWS DMS 支持以下 Amazon DocumentDB 数据类型作为区段键列：

- Double
- String
- ObjectId
- 32 位整数
- 64 位整数

## 使用 Amazon DocumentDB 作为源时迁移多个数据库 AWS DMS

AWS DMS 3.4.5 及更高版本仅支持在单个任务中迁移多个数据库，仅适用于 Amazon DocumentDB 版本 4.0 及更高版本。如果要迁移多个数据库，请执行以下操作：

### 1. 创建 Amazon DocumentDB 源端点时：

- 在 AWS Management Console 或 AWS DMS 中，将创建终端节点页面上的“端点配置”下的数据库名称留空。
- 在 AWS Command Line Interface (AWS CLI) 中，为操作指定的 `documentDbSetting.DatabaseNames` 中的参数分配一个空字符串值。CreateEndpoint

### 2. 对于要从此 Amazon DocumentDB 源端点迁移的每个数据库，使用控制台中的指导式输入或直接以 JSON 格式输入，将每个数据库的名称指定为任务的表映射中的架构名称。有关指导式输入的更多信息，请参阅 [通过控制台指定表选择和转换规则](#) 中的描述。有关 JSON 的更多信息，请参阅 [选择规则和操作](#)。

例如，您可以指定以下 JSON 以迁移三个 Amazon DocumentDB 数据库。

## Example 迁移架构中的所有表

以下 JSON 会将源端点中 Customers、Orders 和 Suppliers 数据库的所有表迁移到目标端点。

```
{
  "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"
    }
  ]
}
```

## 使用亚马逊 DocumentDB 作为来源时的限制 AWS DMS

以下是使用亚马逊 DocumentDB 作为来源时的限制：AWS DMS

- 当 `_id` 选项设置为单独一列时，ID 字符串不能超过 200 个字符。
- 在表模式下，对象 ID 和数组类型键将转换为具有 `oid` 和 `array` 前缀的列。

将使用具有前缀的名称在内部引用这些列。如果您在 AWS DMS 引用这些列时使用转换规则，请务必指定带前缀的列。例如，指定 `${oid__id}` 而不是 `${_id}`，或者指定 `${array__addresses}` 而不是 `${_addresses}`。

- 集合名称和键名不能包含美元符号 (\$)。
- 表模式和文档模式具有前面讨论的限制。
- 使用自动分段进行并行迁移时，存在上述限制。

- Amazon DocumentDB (兼容 MongoDB) 源不支持使用特定的时间戳作为更改数据捕获 (CDC) 的起始位置。无论时间戳如何, 持续复制任务都会开始捕获更改。
- 当使用 DocumentDB (兼容 MongoDB) 作为源时, DMS 每秒最多可以处理 250 条记录。
- AWS DMS 不支持嵌套级别大于 97 的文档。
- DocumentDB 不支持源筛选器。
- AWS DMS 在弹性集群模式下, 不支持 DocumentDB 的 CDC (更改数据捕获) 复制作为源。

## 在以 Amazon DocumentDB 为源时使用端点设置

您可以使用端点设置来配置 Amazon DocumentDB 源数据库, 这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建源端点时指定设置, 或者使用中带有 `--doc-db-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` [AWS CLI](#) 命令来指定设置。

下表显示了在将 Amazon DocumentDB 用作源时可以使用的端点设置。

属性名称	有效值	默认值和描述
NestingLevel	"none" "one"	"none" – 指定 "none" 以使用文档模式。指定 "one" 使用表模式。
ExtractDocID	布尔值 true false	false – 当 NestingLevel 设为 "none" 时使用此属性。  如果您的目标数据库是 Amazon DocumentDB, 请设置 <code>'{"ExtractDocID": true}'</code> 。
DocsToInvestigate	大于 0 的正整数。	1000 – 当 NestingLevel 设为 "one" 时使用此属性。
ReplicateShardCollections	布尔值 true false	如果为 true, AWS DMS 则将数据复制到分片集合。 AWS DMS 仅当目标终端节点是 DocumentDB 弹性集群时才使用此设置。  当此项设置为 true 时, 请注意以下几点:

属性名称	有效值	默认值和描述
		<ul style="list-style-type: none"> <li>• 必须将 TargetTablePrepMode 设置为 nothing。</li> <li>• AWS DMS 自动设置 useUpdateLookup 为 false。</li> </ul>

## Amazon DocumentDB 的源数据类型

在下表中，您可以找到使用 AWS DMS 时支持的 Amazon DocumentDB 源数据类型。您还可以在此表中找到 AWS DMS 数据类型的默认映射。有关数据类型的更多信息，请参阅 MongoDB 文档中的 [BSON 类型](#)。

有关如何查看目标中映射的数据类型的信息，请参阅有关所使用的目标终端节点的部分。

有关 AWS DMS 数据类型的更多信息，请参见 [AWS Database Migration Service 的数据类型](#)。

Amazon DocumentDB 数据类型	AWS DMS 数据类型
布尔值	布尔型
二元	BLOB
Date	Date
时间戳	Date
Int	INT4
长整型	INT8
Double	REAL8
字符串 (UTF-8)	CLOB
数组	CLOB
OID	String

## 使用亚马逊 S3 作为来源 AWS DMS

您可以使用从 Amazon S3 存储桶迁移数据 AWS DMS。为此，请提供对包含一个或多个数据文件的 Amazon S3 存储桶的访问。在该 S3 存储桶中，请包含一个 JSON 文件以描述该数据与这些文件中的数据的数据表之间的映射。

在开始完全加载之前，必须在 Amazon S3 存储桶中包含源数据文件。您可以使用 `bucketName` 参数指定存储桶名称。

源数据文件可以采用以下格式：

- 逗号分隔值 (.csv)
- Parquet (DMS 版本 3.5.3 及更高版本)。有关使用 Parquet 格式文件的信息，请参见 [使用 Amazon S3 中的拼木地板格式文件作为来源 AWS DMS](#)

对于逗号分隔值 (.csv) 格式的源数据文件，请使用以下命名约定对其进行命名。在该约定中，*schemaName* 是源架构，*tableName* 是该架构中的表的名称。

```
/schemaName/tableName/LOAD001.csv  
/schemaName/tableName/LOAD002.csv  
/schemaName/tableName/LOAD003.csv  
...
```

例如，假定数据文件位于以下 Amazon S3 路径的 `mybucket` 中。

```
s3://mybucket/hr/employee
```

加载时，AWS DMS 假设源架构名称为 `hr`，源表名为 `employee`。

除了 `bucketName` (必填项) 之外，您还可以选择提供一个 `bucketFolder` 参数来指定 AWS DMS 应在 Amazon S3 存储桶中查找数据文件的位置。继续前面的示例，如果设置 `bucketFolder` 为 `sourcedata`，则 AWS DMS 读取以下路径中的数据文件。

```
s3://mybucket/sourcedata/hr/employee
```

您可以使用额外连接属性指定列分隔符、行分隔符、null 值指示符以及其他参数。有关更多信息，请参见 [Amazon S3 的终端节点设置作为来源 AWS DMS](#)。

您可以使用 `ExpectedBucketOwner` Amazon S3 端点设置指定存储桶所有者并防止狙击，如下所示。然后，当您发出测试连接或执行迁移的请求时，S3 会根据指定参数检查存储桶所有者的账户 ID。



```
--s3-settings='{ "ExpectedBucketOwner": "AWS_Account_ID" }'
```

## 主题

- [将 Amazon S3 的外部表定义为数据源 AWS DMS](#)
- [使用 CDC 并将 Amazon S3 作为 AWS DMS 的源](#)
- [使用 Amazon S3 作为来源的先决条件 AWS DMS](#)
- [使用 Amazon S3 作为来源时的限制 AWS DMS](#)
- [Amazon S3 的终端节点设置作为来源 AWS DMS](#)
- [Amazon S3 的源数据类型](#)
- [使用 Amazon S3 中的拼木地板格式文件作为来源 AWS DMS](#)

## 将 Amazon S3 的外部表定义为数据源 AWS DMS

除了数据文件以外，您还必须提供外部表定义。外部表定义是一个 JSON 文档，它描述了 AWS DMS 应如何解释 Amazon S3 中的数据。该文档的最大大小为 2 MB。如果您使用 AWS DMS 管理控制台创建源端点，则可以直接在表映射框中输入 JSON。如果您使用 AWS Command Line Interface (AWS CLI) 或 AWS DMS API 执行迁移，则可以创建一个 JSON 文件来指定外部表的定义。

假定您具有包含以下内容的文件。

```
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
```

以下是该数据的示例外部表定义。

```
{
  "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"
}
]
}

```

该 JSON 文档中的元素如下所示：

**TableCount** – 源表数。在该示例中，只有一个表。

**Tables** – 一个数组，对于每个源表，它包含一个 JSON 映射。在该示例中，只有一个映射。每个映射包含以下元素：

- **TableName** – 源表的名称。
- **TablePath** – AWS DMS 可以在 Amazon S3 存储桶中找到完整数据加载文件的路径。如果指定了 `bucketFolder` 值，则在路径前面添加其值。
- **TableOwner** – 该表的架构名称。
- **TableColumns** – 包含一个或多个映射的数组，每个映射描述源表中的一个列：

- `ColumnName` – 源表中的列的名称。
- `ColumnType` – 列的数据类型。有关有效的数据类型，请参阅[Amazon S3 的源数据类型](#)。
- `ColumnLength` – 该列中的字节数。由于 S3 源不支持完整 LOB 模式，因此最大列长度限制为 2147483647 字节 (2,047 MegaBytes)。 `ColumnLength` 对以下数据类型有效：
  - `BYTE`
  - `string`
- `ColumnNullable` – 一个布尔值；如果该列可以包含 NULL 值，则此值为 `true` (默认值为 `false`)。
- `ColumnIsPk` – 一个布尔值；如果该列是主键的一部分，则此值为 `true` (默认值为 `false`)。
- `ColumnDateFormat` – 具有 `DATE`、`TIME` 和 `DATETIME` 类型的列的输入日期格式，用于将数据字符串解析为日期对象。可能的值包括：

```
- 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` – 总列数。该数字必须与 `TableColumns` 数组中的元素数相匹配。

如果您未另行指定，则 AWS DMS 假定 `ColumnLength` 为零。

#### Note

在支持的版本中 AWS DMS，S3 源数据还可以包含一个可选的操作列，作为列值之前的第一 `TableName` 列。此操作列标识在完全加载期间用于将数据迁移到 S3 目标终端节点的操作 (`INSERT`)。

如果存在，则此列的值是 `INSERT` 操作关键字的第一个字符 (`I`)。如果指定，则此列通常指示在之前的迁移中 DMS 创建的 S3 源作为 S3 目标。

在 3.4.2 之前的 DMS 版本中，通过以前的 DMS 完全加载创建的 S3 源数据中没有此列。将此列添加到 S3 目标数据可以使得写入 S3 目标的所有行的格式保持一致，不论它们是在完全加载还是 CDC 加载期间写入。有关格式化 S3 目标数据的选项的更多信息，请参阅[指示迁移的 S3 数据中的源数据库操作](#)。

对于 NUMERIC 类型的列，指定精度和小数位数。精度是数字中的总位数，而小数位数是小数点右边的位数。为此，您可以使用 ColumnPrecision 和 ColumnScale 元素，如下所示。

```
...
  {
    "ColumnName": "HourlyRate",
    "ColumnType": "NUMERIC",
    "ColumnPrecision": "5"
    "ColumnScale": "2"
  }
...
```

对于数据包含小数秒的 DATETIME 类型的列，请指定小数位数。小数位数是小数秒的位数，范围可以从 0 到 9。为此，您可以使用 ColumnScale 元素，如下所示。

```
...
  {
    "ColumnName": "HireDate",
    "ColumnType": "DATETIME",
    "ColumnScale": "3"
  }
...
```

如果未另行指定，则 AWS DMS 假定 ColumnScale 为零并截断小数秒。

## 使用 CDC 并将 Amazon S3 作为 AWS DMS 的源

AWS DMS 执行完整数据加载后，它可以选择将数据更改复制到目标端点。为此，您需要将变更数据捕获文件（CDC 文件）上传到您的 Amazon S3 存储桶。AWS DMS 在上传这些 CDC 文件时读取它们，然后将更改应用到目标端点。

CDC 文件是按如下方式命名的：

```
CDC00001.csv
CDC00002.csv
CDC00003.csv
...
```

**Note**

要在更改数据文件夹中成功地复制 CDC 文件，请按词汇（序列）顺序上传它们。例如，在文件 CDC00003.csv 之前上传文件 CDC00002.csv。否则，如果在 CDC00003.csv 之后加载 CDC00002.csv，则会跳过后者而不会复制。但是，如果在 CDC00003.csv 之后加载文件 CDC00004.csv，则该文件复制成功。

要指明在哪里 AWS DMS 可以找到文件，请指定 `cdcPath` 参数。对于前面的示例，如果将 `cdcPath` 设置为 `changedata`，则 AWS DMS 读取以下路径中的 CDC 文件。

```
s3://mybucket/changedata
```

如果将 `cdcPath` 设置为 `changedata`，并将 `bucketFolder` 设置为 `myFolder`，AWS DMS 会读取以下路径中的 CDC 文件。

```
s3://mybucket/myFolder/changedata
```

CDC 文件中的记录格式如下所示：

- 操作 – 要执行的更改操作：INSERT ( I )、UPDATE ( U ) 或 DELETE ( D )。这些关键字和字符值不区分大小写。

**Note**

在支持的 AWS DMS 版本中，AWS DMS 可以通过两种方式识别每个加载记录要执行的操作。AWS DMS 可以根据记录的关键字值（例如 INSERT）或其关键字的初始字符（例如 I）执行此操作。在之前的版本中，只能从完整的关键字值中 AWS DMS 识别加载操作。在的早期版本中 AWS DMS，写入了完整的关键字值来记录 CDC 数据。以前的版本仅使用关键字首字母将操作值写入任何 S3 目标。

无论如何写入操作列 AWS DMS 来创建 S3 源数据，都可以识别这两种格式来处理操作。此方法支持使用 S3 目标数据作为源用于以后的迁移。借助这种方法，您无需更改以后 S3 源的操作列中显示的任意关键字首字母值的格式。

- 表名称 – 源表的名称。
- 架构名称 – 源架构的名称。
- 数据 – 表示要更改的数据的一个或多个列。

以下是名为 `employee` 的表的示例 CDC 文件。

```
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
```

## 使用 Amazon S3 作为来源的先决条件 AWS DMS

要使用 Amazon S3 作为来源 AWS DMS，您的源 S3 存储桶必须与迁移数据的 DMS 复制实例位于同一 AWS 区域。此外，用于迁移的 AWS 账户必须具有源存储桶的读取访问权限。

分配给用于创建迁移任务的用户账户的 AWS Identity and Access Management (IAM) 角色必须具有以下权限集。

```
{
  "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*"
      ]
    }
  ]
}
```

如果在 Amazon S3 存储桶上启用了版本控制，则分配给用于创建迁移任务的用户账户的 AWS Identity and Access Management (IAM) 角色必须具有以下权限集。

```
{
  "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*"
      ]
    }
  ]
}
```

## 使用 Amazon S3 作为来源时的限制 AWS DMS

将 Amazon S3 作为源时存在以下限制：

- 不要为 S3 启用版本控制。如果您需要 S3 版本控制，请使用生命周期策略主动删除旧版本。否则，您可能会遇到由于 S3 list-object 调用超时而导致端点测试连接失败的情况。要为 S3 存储桶创建生命周期策略，请参阅[管理存储生命周期](#)。要删除 S3 对象的某个版本，请参阅[从启用版本控制的存储桶中删除对象版本](#)。
- 3.4.7 及更高版本支持启用了 VPC ( 网关 VPC ) 的 S3 存储桶。
- MySQL 将 time 数据类型转换为 string。要在 MySQL 中查看 time 数据类型值，请将目标表中的列定义为 string，然后将任务的目标表准备模式设置设置为 Truncate。
- AWS DMS 在内部对两种 BYTE 数据类型 BYTE 和数据类型都使用该 BYTES 数据类型。
- S3 源端点不支持 DMS 表重新加载功能。
- AWS DMS 不支持以 Amazon S3 作为源的完整 LOB 模式。

在 Amazon S3 中使用 Parquet 格式的文件作为源文件时，存在以下限制：

- S3 Parquet DDMMYYYY t Source 日期分区功能不支持或不支持日期。MMYYYYDD

## Amazon S3 的终端节点设置作为来源 AWS DMS

您可以使用端点设置来配置 Amazon S3 源数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建源端点时指定设置，或者使用中带有 `--s3-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` [AWS CLI](#) 命令来指定设置。

下表显示了将 Amazon S3 作为源时可以使用的端点设置。

选项	描述
BucketFolder	(可选) S3 存储桶中的文件夹名称。如果提供该属性，则分别从 <code>s3://myBucket/bucketFolder /schemaName /tableName /</code> 和 <code>s3://myBucket/bucketFolder /</code> 路径中读取源数据文件和 CDC 文件。如果未指定此属性，则使用的路径为 <code>schemaName /tableName /</code> 。  '{"BucketFolder": " <i>sourceData</i> "}'
BucketName	S3 桶的名称。  '{"BucketName": " <i>myBucket</i> "}'
CdcPath	CDC 文件的位置。如果任务捕获更改数据，则该属性是必需的，否则，该属性是可选的。如果 CdcPath 存在，则从此路径 AWS DMS 读取 CDC 文件并将数据更改复制到目标端点。有关更多信息，请参阅 <a href="#">使用 CDC 并将 Amazon S3 作为 AWS DMS 的源</a> 。  '{"CdcPath": " <i>changeData</i> "}'
CsvDelimiter	源文件中用于分隔列的分隔符。默认值为逗号。下面是一个示例。  '{"CsvDelimiter": ","}'
CsvNullValue	用户定义的字符串，AWS DMS 从源读取时将其视为 null。默认值是空字符串。如果未设置此参数，则 AWS DMS 将空字符串视为空值。如果将此



选项	描述
	参数设置为诸如“\N”之类的字符串，则 AWS DMS 会将此字符串视为空值，并将空字符串视为空字符串值。
CsvRowDelimiter	源文件中用于分隔行的分隔符。默认值为换行符 (\n)。  '{"CsvRowDelimiter": "\n"}'
DataFormat	将此值设置 Parquet 为可读取 Parquet 格式的数据。  '{"DataFormat": "Parquet"}'
IgnoreHeaderRows	当此值设置为 1 时，将 AWS DMS 忽略.csv 文件中的第一行标题。值为 1 可启用该功能，值为 0 将禁用该功能。  默认值是 0。  '{"IgnoreHeaderRows": 1}'
Rfc4180	当此值设置为 true 或 y 时，每个前导双引号必须后跟一个结束双引号。此格式符合 RFC 4180。当此值设置为 false 或 n 时，字符串文字将按原样复制到目标。在这种情况下，分隔符（行或列）表示字段的末尾。因此，您不能使用分隔符作为字符串的一部分，因为它表示值的末尾。  默认值为 true。  有效值：true、false、y、n  '{"Rfc4180": false}'

## Amazon S3 的源数据类型

使用 Amazon S3 作为数据源的数据迁移 AWS DMS 需要将数据从 Amazon S3 映射到 AWS DMS 数据类型。有关更多信息，请参阅 [将 Amazon S3 的外部表定义为数据源 AWS DMS](#)。

有关如何查看目标中映射的数据类型的信息，请参阅有关所使用的目标终端节点的部分。

有关 AWS DMS 数据类型的更多信息，请参见 [AWS Database Migration Service 的数据类型](#)。

以下 AWS DMS 数据类型用于 Amazon S3 作为数据源：

- BYTE – 需要 ColumnLength。有关更多信息，请参阅 [将 Amazon S3 的外部表定义为数据源 AWS DMS](#)。
- DATE
- TIME
- DATETIME – 有关更多信息和示例，请参阅 [将 Amazon S3 的外部表定义为数据源 AWS DMS](#) 中的 DATETIME 类型示例。
- INT1
- INT2
- INT4
- INT8
- 数字-需要ColumnPrecision和ColumnScale。AWS DMS 支持以下最大值：
  - ColumnPrecision: 38
  - ColumnScale: 31

有关更多信息和示例，请参阅 [将 Amazon S3 的外部表定义为数据源 AWS DMS](#) 中的 NUMERIC 类型示例。

- REAL4
- REAL8
- STRING – 需要 ColumnLength。有关更多信息，请参阅 [将 Amazon S3 的外部表定义为数据源 AWS DMS](#)。
- UINT1
- UINT2
- UINT4
- UINT8
- BLOB
- CLOB
- BOOLEAN

## 使用 Amazon S3 中的拼木地板格式文件作为来源 AWS DMS

在 3.5.3 及更高 AWS DMS 版本中，您可以使用 S3 存储桶中的 Parquet 格式文件作为满载或 CDC 复制的来源。

DMS 仅支持 Parquet 格式的文件作为 DMS 通过将数据迁移到 S3 目标端点而生成的源。文件名必须采用支持的格式，否则 DMS 不会将其包含在迁移中。

对于 Parquet 格式的源数据文件，它们必须位于以下文件夹和命名约定中。

```
schema/table1/LOAD00001.parquet
schema/table2/LOAD00002.parquet
schema/table2/LOAD00003.parquet
```

对于 Parquet 格式的 CDC 数据的源数据文件，请使用以下文件夹和命名约定对其进行命名和存储。

```
schema/table/20230405-094615814.parquet
schema/table/20230405-094615853.parquet
schema/table/20230405-094615922.parquet
```

要访问 Parquet 格式的文件，请设置以下端点设置：

- 将 DataFormat 设置为 Parquet。
- 请勿设置该cdcPath设置。确保在指定的 schema/ 表文件夹中创建 Parquet 格式的文件。

有关 S3 终端节点设置的更多信息，请参阅 AWS Database Migration Service API 参考[中的 S3 Settings](#)。

### Parquet 格式文件支持的数据类型

AWS DMS 从 Parquet 格式文件迁移数据时，支持以下源和目标数据类型。迁移之前，请确保目标表中包含正确数据类型的列。

源数据类型	目标数据类型
BYTE	BINARY
DATE	DATE32
TIME	TIME32
DATETIME	TIMESTAMP
INT1	INT8

源数据类型	目标数据类型
INT2	INT16
INT4	INT32
INT8	INT64
NUMERIC	DECIMAL
REAL4	FLOAT
REAL8	DOUBLE
STRING	STRING
UINT1	UINT8
UINT2	UINT16
UINT4	UINT32
UINT8	UINT
WSTRING	STRING
BLOB	BINARY
NCLOB	STRING
CLOB	STRING
BOOLEAN	BOOL

## 使用适用于 Linux、Unix、Windows 的 IBM Db2 和亚马逊 RDS 数据库 (Db2 LUW) 作为来源 AWS DMS

您可以使用 () 将数据从适用于 Linux、Unix、Windows 和 Amazon RDS (Db2 LUW) 的 IBM Db AWS Database Migration Service 2 数据库迁移到任何支持的目标数据库。AWS DMS

有关 AWS DMS 支持作为源的 Linux、Unix、Windows 和 RDS 上的 Db2 版本的信息，请参阅[的来源 AWS DMS](#)。

可以使用安全套接字层 (SSL) 加密 Db2 LUW 端点与复制实例之间的连接。有关将 SSL 用于 Db2 LUW 终端节点的更多信息，请参阅[将 SSL 与 AWS Database Migration Service](#)。

## 使用 Db2 LUW 作为源时的先决条件 AWS DMS

需要先满足以下先决条件，然后才能将 Db2 LUW 数据库作为源。

要启用持续复制（也称为更改数据捕获 (CDC)），请执行以下操作：

- 将数据库设置为可恢复，这 AWS DMS 需要捕获更改。如果数据库配置参数 LOGARCHMETH1 和/或 LOGARCHMETH2 设置为 ON，则数据库是可恢复的。

如果您的数据库是可恢复的，则 AWS DMS 可以在 ARCHIVE LOG 需要时访问 Db2。

- 确保 DB2 事务日志可用，并有足够的保留期供 AWS DMS 其处理。
- DB2 需要 SYSADM 或 DBADM 授权才能提取事务日志记录。向用户账户授予以下权限：
  - SYSADM 或 DBADM
  - DATAACCESS

### Note

对于仅完全加载任务，DMS 用户账户需要 DATAACCESS 权限。

- 使用 IBM DB2 for LUW 版本 9.7 作为源时，请按如下方式设置额外连接属性（ECA）：`CurrentLSN`

`CurrentLSN=LSN`，这里的 *LSN* 指定要在其中启动复制的日志序列号（LSN）。或者 `CurrentLSN=scan`。

## 使用 Db2 LUW 作为源代码时的限制 AWS DMS

AWS DMS 不支持群集数据库。但是，您可以为集群的每个终端节点定义一个单独的 Db2 LUW。例如，您可以使用集群中的任何一个节点创建完全加载迁移任务，然后从每个节点创建单独的任务。

AWS DMS 不支持您的源 Db2 LUW 数据库中的 BOOLEAN 数据类型。

在使用持续复制 (CDC) 时，以下限制将适用：

- 当包含多个分区的表被截断时，AWS DMS 控制台中显示的 DDL 事件数等于分区数。这是因为 Db2 LUW 为每个分区记录一个单独的 DDL。
- 已分区表上不支持以下 DDL 操作：
  - ALTER TABLE ADD PARTITION
  - ALTER TABLE DETACH PARTITION
  - ALTER TABLE ATTACH PARTITION
- AWS DMS 不支持从 DB2 高可用性灾难恢复 (HADR) 备用实例进行持续复制迁移。备用实例无法访问。
- 不支持 DECFLOAT 数据类型。因此，在持续复制期间，将忽略对 DECFLOAT 列所做的更改。
- 不支持 RENAME COLUMN 语句。
- 对多维聚类 (MDC) 表执行更新时，每次更新都会在 AWS DMS 控制台中显示为 INSERT + DELETE。
- 如果未启用任务设置 Include LOB columns in replication (在复制中包含 LOB 列)，则将在持续复制期间暂停任何具有 LOB 列的表。
- 对于 Db2 LUW 版本 10.5 及更高版本，将忽略包含存储数据的可变长度字符串列。out-of-row 此限制仅适用于对于具有 VARCHAR 和 VARGRAPHIC 等数据类型的列，那些使用扩展行大小创建的表。要解决此限制，请将表移至具有较大页面大小的表格空间。有关更多信息，请参阅[如果我想更改 DB2 表空间的页面大小该怎么办](#)。
- 对于正在进行的复制，DMS 不支持对 DB2 LOAD 实用程序在页面级别加载的数据进行迁移。相反，请使用 IMPORT 实用程序，该实用程序使用 SQL 插入。有关更多信息，请参阅[IMPORT 和 LOAD 实用程序之间的区别](#)。
- 当复制任务正在运行时，只有在使用 DATA CAPTURE CHANGE 属性创建了表时，DMS 才捕获 CREATE TABLE DDL。
- 使用 Db2 数据库分区功能 (DPF) 时，DMS 存在以下限制：
  - 在 DPF 环境中，DMS 无法在 Db2 节点之间协调事务。这是由于 IBM DB2READLOG API 接口中的限制造成的。在 DPF 中，事务可能跨越多个 Db2 节点，具体取决于 DB2 对数据进行分区的方式。因此，您的 DMS 解决方案必须独立地从每个 Db2 节点捕获事务。
  - 通过在多个 DMS 源端点 1 上设置 connectNode 为，DMS 可以从 DPF 集群中的每个 Db2 节点捕获本地事务。此配置对应于 DB2 服务器配置文件 db2nodes.cfg 中定义的逻辑节点号。
  - 单个 Db2 节点上的本地事务可能是更大的全局事务的一部分。DMS 在目标上独立应用每个本地事务，无需与其他 Db2 节点上的事务进行协调。这种独立的处理可能会导致复杂性，尤其是在分区之间移动行时。

- 当 DMS 从多个 Db2 节点复制时，无法保证目标上的操作顺序是正确的，因为 DMS 会独立地为每个 Db2 节点应用操作。您必须确保独立于每个 Db2 节点捕获本地事务适用于您的特定用例。
- 从 DPF 环境迁移时，我们建议先运行不带缓存事件的满载任务，然后运行仅限 CDC 的任务。我们建议每个 Db2 节点运行一个任务，从您使用端点设置设置的满载开始时间戳或 LRI ( 日志记录标识符 ) 开始。StartFromContext 有关确定复制起点的信息，请参阅 IBM Support 文档中的[查找复制开始的 LSN 或 LRI 值](#)。
- 对于持续复制 ( CDC )，如果您计划从特定时间戳开始复制，则必须将 StartFromContext 连接属性设置为所需的时间戳。
- 目前，DMS 不支持 Db2 pureScale 功能，这是 DB2 LUW 的扩展，可以用来扩展数据库解决方案。
- AWS DMS 使用适用于 Amazon RDS 的 Db2 作为来源时不支持 CDC。

## 使用 Db2 LUW 作为源时的端点设置 AWS DMS

您可以使用端点设置来配置 Db2 LUW 源数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建源端点时指定设置，或者使用中带有 `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` [AWS CLI](#) 命令来指定设置。

下表显示了将 Db2 LUW 用作源时可以使用的端点设置。

名称	描述
CurrentLSN	对于持续复制 (CDC)，请使用 CurrentLSN 指定要在其中启动复制的日志序列号 (LSN)。
MaxKBytesPerRead	每次读取的最大字节数，以 NUMBER 值表示。默认值为 64 KB。
SetDataCaptureChanges	启用持续复制 (CDC) 作为 BOOLEAN 值。默认值为 true。
StartFromContext	对于持续复制 ( CDC )，请使用 StartFromContext 指定日志的下限，复制将从此位置开始。StartFromContext 接受不同格式的值。有效值包括： <ul style="list-style-type: none"> <li>• timestamp ( UTC )。例如：</li> </ul>

名称	描述
	<pre data-bbox="721 212 1507 327">'{"StartFromContext": "timestamp:2021-09-21T13:00:00"}'</pre> <ul data-bbox="688 401 781 432" style="list-style-type: none"> <li>• NOW</li> </ul> <p data-bbox="721 478 1463 604">对于 IBM DB2 LUW 版本 10.5 及更高版本，NOW 与 CurrentLSN: scan 结合使用，从最新的 LSO 启动任务。例如：</p> <pre data-bbox="721 646 1507 762">'{"CurrentLSN": "scan", "StartFromContext": "NOW"}'</pre> <ul data-bbox="688 835 959 867" style="list-style-type: none"> <li>• 特定 LRI。例如：</li> </ul> <pre data-bbox="721 909 1507 1024">'{"StartFromContext": "0100000000000022CC000000000004FB13"}'</pre> <p data-bbox="688 1094 1495 1178">要确定日志文件的 LRI/LSN 范围，请按以下示例所示运行 db2f1sn 命令。</p> <pre data-bbox="688 1220 1507 1293">db2f1sn -db <i>SAMPLE</i> -lrange 2</pre> <p data-bbox="688 1329 1117 1367">该示例的输出类似于以下内容。</p> <pre data-bbox="688 1409 1507 1644">S0000002.LOG: has LRI range 0000000000000010000000000000004F9A6 to 000000000000001000000000000022CC00000000004FB13</pre> <p data-bbox="688 1682 1451 1766">在该输出中，日志文件为 S0000002 .LOG，StartFrom ContextLRI 值为范围末尾的 34 字节。</p>



名称	描述
	010000000000000022CC000000000004FB13

## IBM Db2 LUW 的源数据类型

使用 Db2 LUW 作为源的数据迁移 AWS DMS 支持大多数 Db2 LUW 数据类型。下表显示了使用时支持的 Db2 LUW 源数据类型 AWS DMS 以及 AWS DMS 数据类型的默认映射。有关 Db2 LUW 数据类型的更多信息，请参阅 [Db2 LUW 文档](#)。

有关如何查看目标中映射的数据类型的信息，请参阅使用的目标终端节点对应的部分。

有关 AWS DMS 数据类型的更多信息，请参见 [AWS Database Migration Service 的数据类型](#)。

Db2 LUW 数据类型	AWS DMS 数据类型
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s)
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	如果精度为 16，则为 REAL8；如果精度为 34，则为 STRING
GRAPHIC (n)	WSTRING，用于长度大于 0 且小于或等于 127 个双字节字符的固定长度图形字符串
VARGRAPHIC (n)	WSTRING，用于长度大于 0 且小于或等于 16352 个双字节字符的可变长度图形字符串

Db2 LUW 数据类型	AWS DMS 数据类型
LONG VARCHAR(n)	CLOB，用于长度大于 0 且小于或等于 16352 个双字节字符的可变长度图形字符串
CHARACTER(n)	STRING，用于长度大于 0 且小于或等于 255 个双字节字符的固定长度字符串
VARCHAR(n)	STRING，用于长度大于 0 且小于或等于 32704 个双字节字符的可变长度字符串
LONG VARCHAR(n)	CLOB，用于长度大于 0 且小于或等于 32704 个双字节字符的可变长度字符串
CHAR(n) FOR BIT DATA	BYTES
VARCHAR(n) FOR BIT DATA	BYTES
LONG VARCHAR FOR BIT DATA	BYTES
DATE	DATE
TIME	TIME
TIMESTAMP	DATETIME
BLOB(n)	BLOB 最大长度为 2147483647 个字节
CLOB(n)	CLOB 最大长度为 2147483647 个字节
DBCLOB(n)	CLOB 最大长度为 1073741824 个双字节字符
XML	CLOB

## 将 IBM Db2 for z/OS 数据库用作 AWS DMS 的源

您可以使用 AWS Database Migration Service (AWS DMS) 将数据从 IBM for z/OS 迁移到任何受支持的目标数据库。

有关 AWS DMS 支持作为源的 Db2 for z/OS 版本的信息，请参阅 [的来源 AWS DMS](#)。

### 将 Db2 for z/OS 作为 AWS DMS 源时的先决条件

要在 AWS DMS 中使用 IBM Db2 for z/OS 数据库作为源，请向在源端点连接设置中指定的 Db2 for z/OS 用户授予以下权限。

```
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
```

还要授予 `SELECT ON user defined` 源表权限。

AWS DMS IBM Db2 for z/OS 源端点依靠 IBM Data Server Driver for ODBC 来访问数据。数据库服务器必须具有有效的 IBM ODBC Connect 许可证，DMS 才能连接到此端点。

### 将 Db2 for z/OS 作为 AWS DMS 源时的限制

将 Db2 for z/OS 数据库作为 AWS DMS 的源时，存在以下限制：

- 仅支持完全加载复制任务。不支持更改数据捕获 (CDC)。
- 不支持并行加载。
- 不支持对视图进行数据验证。
- 在表映射中，对于列/表级转换和行级选择筛选器，必须以大写形式指定架构、表和列的名称。

### IBM Db2 for z/OS 的源数据类型

使用 Db2 for z/OS 作为 AWS DMS 的源进行数据迁移时支持大部分 Db2 for z/OS 数据类型。下表显示了使用 AWS DMS 时支持的 Db2 for z/OS 源数据类型，以及来自 AWS DMS 数据类型的默认映射。

有关 Db2 for z/OS 数据类型的更多信息，请参阅 [IBM Db2 for z/OS 文档](#)。

有关如何查看目标中映射的数据类型的信息，请参阅使用的目标终端节点对应的部分。

有关 AWS DMS 数据类型的其他信息，请参阅[AWS Database Migration Service 的数据类型](#)。

Db2 for z/OS 数据类型	AWS DMS 数据类型
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s)  如果在 DB2 配置中将小数点设置为逗号 (,)，请将复制配置为支持 DB2 设置。
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	如果精度为 16，则为 REAL8；如果精度为 34，则为 STRING
GRAPHIC (n)	如果 $n \geq 127$ ，则为 WSTRING，用于双字节字符的固定长度图形字符串，长度大于 0 且小于或等于 127
VARGRAPHIC (n)	WSTRING，用于长度大于 0 且小于或等于 16352 个双字节字符的可变长度图形字符串
LONG VARGRAPHIC (n)	CLOB，用于长度大于 0 且小于或等于 16352 个双字节字符的可变长度图形字符串
CHARACTER (n)	STRING，用于长度大于 0 且小于或等于 255 个双字节字符的固定长度字符串
VARCHAR (n)	STRING，用于长度大于 0 且小于或等于 32704 个双字节字符的可变长度字符串

Db2 for z/OS 数据类型	AWS DMS 数据类型
LONG VARCHAR (n)	CLOB , 用于长度大于 0 且小于或等于 32704 个双字节字符的可变长度字符串
CHAR (n) FOR BIT DATA	BYTES
VARCHAR (n) FOR BIT DATA	BYTES
LONG VARCHAR FOR BIT DATA	BYTES
DATE	DATE
TIME	TIME
TIMESTAMP	DATETIME
BLOB (n)	BLOB 最大长度为 2147483647 个字节
CLOB (n)	CLOB 最大长度为 2147483647 个字节
DBCLOB (n)	CLOB 最大长度为 1073741824 个双字节字符
XML	CLOB
BINARY	BYTES
VARBINARY	BYTES
ROWID	BYTES。有关使用 ROWID 的更多信息，请参阅以下内容。
TIMESTAMP ( 有时区 )	不支持。

当任务的目标表准备模式设置为 `DROP_AND_CREATE` (默认) 时, 默认情况下会迁移 `ROWID` 列。数据验证会忽略这些列, 因为这些行在特定的数据库和表之外毫无意义。要禁止迁移这些列, 您可以执行下列准备步骤之一:

- 预先创建不带这些列的目标表。然后, 将任务的目标表格准备模式设置为 `DO_NOTHING` 或 `TRUNCATE_BEFORE_LOAD`。您可以使用 AWS Schema Conversion Tool (AWS SCT) 预先创建不带这些列的目标表。
- 在任务中添加表映射规则筛选掉这些列, 使其被忽略。有关更多信息, 请参阅 [转换规则和操作](#)。

## PostgreSQL for AWS Mainframe Modernization 服务中的 EBCDIC 排序规则

AWS Mainframe Modernization 服务可帮助您将大型机应用程序现代化为 AWS 托管运行时系统环境。它提供了工具和资源来帮助您规划和实施迁移与现代化项目。有关大型机现代化和迁移的更多信息, 请参阅 [AWS 上的 Mainframe Modernization](#)。

某些 IBM Db2 for z/OS 数据集在编码时采用广义二进制编码的十进制交换码 (EBCDIC) 字符集。这是一种在 ASCII (American Standard Code for Information Interchange, 美国信息交换标准代码) 普遍使用之前开发的字符集。代码页将每个文本字符映射到字符集中的字符。传统的代码页包含代码点和字符 ID 之间的映射信息。字符 ID 是一个 8 字节的字符数据字符串。代码点是代表一个字符的 8 位二进制数。代码点通常以其二进制值的十六进制表示形式显示。

如果您目前使用 Mainframe Modernization 服务的 Micro Focus 或 BluAge 组件, 则必须告知 AWS DMS 要转换 (翻译) 某些代码点。您可以使用 AWS DMS 任务设置来执行转换。以下示例说明如何使用 AWS DMS `CharacterSetSettings` 操作在 DMS 任务设置中映射转换。

```
"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"}
  ]
}
```

PostgreSQL 已有一些 EBCDIC 排序规则可以理解所需的转换。系统支持几种不同的代码页。以下各节提供了一些 JSON 示例，说明在所有支持的代码页中必须转换的内容。您只需复制并粘贴 DMS 任务中所需要的 JSON 即可。

### Micro Focus 特定的 EBCDIC 排序规则

对于 Micro Focus，请根据需要对以下排序规则的字符子集进行转换。

```
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 各排序规则的 Micro Focus 数据转换：

en\_us\_cp1140m

代码转换：

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

AWS DMS 任务的相应输入映射：

```
{ "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

代码转换：

0000	0180
00B8	0160
00BC	0161
00BD	017D
00BE	017E
00A8	0152
00B4	0153
00A6	0178

AWS DMS 任务的相应输入映射：

```
{ "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

代码转换：

0000	0180
00A6	0160
00B8	0161
00BC	017D
00BD	017E
00BE	0152
00A8	0153
00B4	0178



## AWS DMS 任务的相应输入映射：

```
{ "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

## 代码转换：

0000	0180
00B8	0160
00BC	0161
00BD	017D
00BE	017E
00A8	0152
00B4	0153
00A6	0178

## AWS DMS 任务的相应输入映射：

```
{ "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

## 代码转换：

```

0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178

```

AWS DMS 任务的相应输入映射：

```

{"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

代码转换：

```

0000    0180
00A6    0160
00B8    0161
00A8    017D
00BC    017E
00BD    0152
00BE    0153
00B4    0178

```

AWS DMS 任务的相应输入映射：

```

{"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

代码转换：

0000	0180
00A6	0160
00B8	0161
00BC	017D
00BD	017E
00BE	0152
00A8	0153
00B4	0178

AWS DMS 任务的相应输入映射：

```
{"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

代码转换：

0000	0180
00B8	0160
00A8	0161
00BC	017D
00BD	017E

00BE	0152
00B4	0153
00A6	0178

### AWS DMS 任务的相应输入映射：

```
{
  "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

### 代码转换：

0000	0180
00A6	0160
00B8	0161
00BC	017D
00BD	017E
00BE	0152
00A8	0153
00B4	0178

### AWS DMS 任务的相应输入映射：

```
{
  "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"
}
```

## BluAge 特定的 EBCDIC 排序规则

对于 BluAge，请根据需要转换以下所有低值和高值。这些排序规则只应用于支持 Mainframe Migration BluAge 服务。

```
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 BluAge 数据转换：

da-DK-cp277b 和 da-DK-cp1142b

代码转换：

```
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

```

### AWS DMS 任务的相应输入映射：

```

{"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 和 de-DE-1141b

代码转换：

```
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

```

#### AWS DMS 任务的相应输入映射：

```

{"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 和 en-GB-1146b

代码转换：

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
```

#### AWS DMS 任务的相应输入映射：

```
{"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 和 en-us-1140b

代码转换：

```
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

AWS DMS 任务的相应输入映射：



```
{"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 和 es-ES-1145b

### 代码转换：

```
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

```

### AWS DMS 任务的相应输入映射：

```

{"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 和 fi-FI-1143b

### 代码转换：

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

```

#### AWS DMS 任务的相应输入映射：

```

{"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 和 fr-FR-1147b

## 代码转换：

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
```

#### AWS DMS 任务的相应输入映射：

```
{"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 和 it-IT-1144b

### 代码转换：

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
```

### AWS DMS 任务的相应输入映射：

```
{ "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 和 nl-BE-1148b

### 代码转换：

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

## AWS DMS 任务的相应输入映射：

```
{ "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"}
```

## 数据迁移的目标

AWS Database Migration Service (AWS DMS) 可以将多种应用最广泛的数据库作为数据复制的目标。目标可能位于 Amazon Elastic Compute Cloud (Amazon EC2) 实例和 Amazon Relational Database Service (Amazon RDS) 实例上，也可能是本地数据库。

有关有效目标的完整列表，请参阅 [AWS DMS 的目标](#)。

### Note

AWS DMS 不支持对以下目标端点类型实施跨 AWS 区域迁移：

- Amazon DynamoDB
- Amazon OpenSearch Service
- Amazon Kinesis Data Streams

## 主题

- [将 Oracle 数据库作为 AWS Database Migration Service 的目标](#)
- [将 Microsoft SQL Server 数据库作为 AWS Database Migration Service 的目标](#)
- [将 PostgreSQL 数据库作为 AWS Database Migration Service 的目标](#)
- [将与 MySQL 兼容的数据库作为 AWS Database Migration Service 的目标](#)
- [将 Amazon Redshift 数据库作为 AWS Database Migration Service 目标](#)
- [将 SAP ASE 数据库作为 AWS Database Migration Service 的目标](#)
- [将 Amazon S3 作为 AWS Database Migration Service 的目标](#)
- [将 Amazon DynamoDB 数据库作为 AWS Database Migration Service 的目标](#)
- [使用 Amazon Kinesis Data Streams 作为目标 AWS Database Migration Service](#)
- [使用 Apache Kafka 作为目标 AWS Database Migration Service](#)
- [使用 Amazon OpenSearch Service 集群作为 AWS Database Migration Service 的目标](#)
- [将 Amazon DocumentDB 作为 AWS Database Migration Service 的目标](#)
- [将 Amazon Neptune 作为 AWS Database Migration Service 的目标](#)
- [将 Redis 作为 AWS Database Migration Service 的目标](#)
- [使用 Babelfish 作为 AWS Database Migration Service 的目标](#)
- [将 Amazon Timestream 作为 AWS Database Migration Service 的目标](#)
- [将 Amazon RDS for Db2 和 IBM Db2 LUW 作为 AWS DMS 的目标](#)

## 将 Oracle 数据库作为 AWS Database Migration Service 的目标

您可以使用从其他 Oracle 数据库或其他支持的数据库之一将数据迁移到 Oracle 数据库目标。AWS DMS 可以使用安全套接字层 (SSL) 加密 Oracle 终端节点与复制实例之间的连接。有关在 Oracle 终端节点上使用 SSL 的更多信息，请参阅[将 SSL 与 AWS Database Migration Service](#)。AWS DMS 还支持使用 Oracle 透明数据加密 (TDE) 来加密目标数据库中的静态数据，因为 Oracle TDE 不需要加密密钥或密码即可写入数据库。

有关 AWS DMS 支持作为目标的 Oracle 版本的信息，请参见[目标 AWS DMS](#)。

将 Oracle 作为目标时，我们假定数据将迁移到目标连接所使用的架构或用户。如果您要将数据迁移到其他架构，请使用架构转换来完成此操作。例如，假设您的目标终端节点连接到用户 RDSMASTER 并且您希望从用户 PERFDATA1 迁移到 PERFDATA2。在这种情况下，请创建一个转换，如下所示。

```
{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "rename",
  "rule-target": "schema",
  "object-locator": {
    "schema-name": "PERFDATA1"
  },
  "value": "PERFDATA2"
}
```

使用 Oracle 作为目标时，会将所有表和索引迁移到目标中的默认表和索引表空间。如果要将表和索引迁移到不同的表和索引表空间，请使用表空间转换来执行此操作。例如，假设您在 INVENTORY 架构中有一组已分配给 Oracle 源中的某些表空间的表。对于迁移，您希望将所有这些表分配给目标中的单个 INVENTORYSPACE 表空间。在这种情况下，请创建一个转换，如下所示。

```
{
  "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"
}
```

有关转换的更多信息，请参阅[使用 JSON 指定表选择和转换规则](#)。

如果 Oracle 同时作为源和目标，则可以通过设置 Oracle 源额外连接属性 `enableHomogenousTablespace=true` 来保留现有的表或索引表空间分配。有关更多信息，请参阅 [使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。

有关使用 Oracle 数据库作为目标的更多详细信息 AWS DMS，请参阅以下各节：

## 主题

- [将 Oracle 作为目标的局限性 AWS Database Migration Service](#)
- [将 Oracle 作为目标所需的用户账户权限](#)
- [将 Oracle 数据库配置为目标 AWS Database Migration Service](#)
- [使用 Oracle 作为目标时的终端节点设置 AWS DMS](#)
- [Oracle 的目标数据类型](#)

## 将 Oracle 作为目标的局限性 AWS Database Migration Service

使用 Oracle 作为数据迁移目标的限制如下所示：

- AWS DMS 不会在目标 Oracle 数据库上创建架构。您必须在目标 Oracle 数据库上创建任意所需的架构。Oracle 目标中很可能已存在该架构名称。源架构中的表将导入用户或架构，该用户或架构 AWS DMS 用于连接到目标实例。要迁移多个架构，可以创建多个复制任务。您也可以将数据迁移到目标上的不同架构。为此，您需要对 AWS DMS 表映射使用架构转换规则。
- AWS DMS 不支持带有 INDEXTYPE 上下文的表的 `Use direct path full load` 选项。作为解决方法，您可以使用数组加载。
- 利用批量优化应用选项，加载到净更改表中时使用直接路径，这不支持 XML 类型。作为解决方法，您可以使用事务应用模式。
- Oracle 目标可以按不同方式处理从源数据库迁移的空字符串（例如，转换为一个空格字符串）。这可能会导致 AWS DMS 验证报告不匹配。
- 您可以使用以下公式表示在批量优化应用模式下支持的每个表的总列数：

$$2 * \text{columns\_in\_original\_table} + \text{columns\_in\_primary\_key} \leq 999$$

例如，如果原始表有 25 列，其主键由 5 列组成，则总列数为 55。如果表超过支持的列数，则所有更改都将在 `one-by-one` 模式下应用。

- AWS DMS 不支持 Oracle 云基础架构 (OCI) 上的自治数据库。

## 将 Oracle 作为目标所需的用户账户权限

要在 AWS Database Migration Service 任务中使用 Oracle 目标，请在 Oracle 数据库中授予以下权限。您可向 AWS DMS 的 Oracle 数据库定义中指定的用户账户授予这些权限。

- SELECT ANY TRANSACTION
- 针对 V\$NLS\_PARAMETERS 的 SELECT
- 针对 V\$TIMEZONE\_NAMES 的 SELECT
- 针对 ALL\_INDEXES 的 SELECT
- 针对 ALL\_OBJECTS 的 SELECT
- 针对 DBA\_OBJECTS 的 SELECT
- 针对 ALL\_TABLES 的 SELECT
- 针对 ALL\_USERS 的 SELECT
- 针对 ALL\_CATALOG 的 SELECT
- 针对 ALL\_CONSTRAINTS 的 SELECT
- 针对 ALL\_CONS\_COLUMNS 的 SELECT
- 针对 ALL\_TAB\_COLS 的 SELECT
- 针对 ALL\_IND\_COLUMNS 的 SELECT
- 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

对于下列要求，授予这些附加权限：

- 要使用特定表列表，请对任意复制表授予 SELECT 权限以及 ALTER 权限。
- 要允许用户在默认表空间中创建表，请授予权限 GRANT UNLIMITED TABLESPACE。
- 对于登录，请授予权限 CREATE SESSION。
- 如果您使用直接路径（完全加载时的默认路径），则 GRANT LOCK ANY TABLE to *dms\_user*;
- 如果使用“DROP and CREATE”表格准备模式时架构不同，则 GRANT CREATE ANY INDEX to *dms\_user*;
- 对于一些完全加载场景，您已选择“DROP and CREATE table (对表执行 DROP 或 CREATE 操作)”或“TRUNCATE before loading (加载前执行 TRUNCATE 操作)”选项，其中目标表架构与 DMS 用户的不同。在此情况下，授予 DROP ANY TABLE 权限。
- 要将更改存储在更改表或审计表（其中目标表架构与 DMS 用户的不同）中，请授予 CREATE ANY TABLE 和 CREATE ANY INDEX 权限。

目标数据库所需的 AWS Database Migration Service 读取权限

必须向 AWS DMS 用户帐户授予以下 DBA 表的读取权限：

- 针对 DBA\_USERS 的 SELECT
- 针对 DBA\_TAB\_PRIVS 的 SELECT
- 针对 DBA\_OBJECTS 的 SELECT
- 针对 DBA\_SYNONYMS 的 SELECT
- 针对 DBA\_SEQUENCES 的 SELECT
- 针对 DBA\_TYPES 的 SELECT
- 针对 DBA\_INDEXES 的 SELECT
- 针对 DBA\_TABLES 的 SELECT
- 针对 DBA\_TRIGGERS 的 SELECT
- SYS.DBA\_REGISTRY 上的 SELECT

如果无法将任意所需权限授予 V\$xxx，则将其授予 V\_\$xxx。

## 移民前评估

要在 Oracle 作为目标[甲骨文评估](#)的情况下使用中列出的预迁移评估，必须向目标dms\_user数据库的数据库用户添加以下权限：

```
GRANT SELECT ON V_$INSTANCE TO dms_user;
```

## 将 Oracle 数据库配置为目标 AWS Database Migration Service

在使用 Oracle 数据库作为数据迁移目标之前，必须向提供一个 Oracle 用户帐户 AWS DMS。该用户帐户必须具有 Oracle 数据库的读/写权限，如[将 Oracle 作为目标所需的用户帐户权限](#)中所指定。

## 使用 Oracle 作为目标时的终端节点设置 AWS DMS

您可以使用端点设置来配置 Oracle 目标数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者使用中的create-endpoint命令和 --oracle-settings '{"EndpointSetting": "value", ...}' JSON 语法。[AWS CLI](#)

下表显示了您可以将 Oracle 作为目标的端点设置。

名称	描述
EscapeCharacter	<p>将此属性设置为转义字符。此转义字符允许您在表映射表达式中将单个通配符用作一个普通字符。有关更多信息，请参阅<a href="#">表映射中的通配符</a>。</p> <p>默认值：Null</p> <p>有效值：除通配符之外的任何字符</p> <p>例如：--oracle-settings '{"Escape Character": "#"}'</p>
UseDirectPathFullLoad	<p>设置为时Y，AWS DMS 使用直接路径满载。指定此值以在 Oracle Call Interface (OCI) 中启用直接路径协议。此 OCI 协议允许在完全加载期间批量加载 Oracle 目标表。</p> <p>默认值：true</p> <p>有效值：true/false</p>



名称	描述
DirectPathParallelLoad	<p>例如：<code>--oracle-settings '{"UseDirectPathFullLoad": false}'</code></p> <p>如果设置为 <code>true</code>，则此属性在 <code>UseDirectPathFullLoad</code> 设置为 <code>Y</code> 时指定并行加载。此属性也仅在您使用 <code>parallel</code> 加载 AWS DMS 功能时适用。有关更多信息，请参阅 <a href="#">表和集合设置规则和操作</a> 中 <code>parallel-load</code> 操作的描述。</p> <p>有关指定此并行加载设置的限制是目标表不能具有任何约束或索引。有关此限制的更多信息，请参阅<a href="#">在并行直接路径加载后启用约束</a>。如果启用约束或索引，则将此属性设置为 <code>true</code> 将不起作用。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true/false</code></p> <p>例如：<code>--oracle-settings '{"DirectPathParallelLoad": true}'</code></p>
DirectPathNoLog	<p>如果设置为 <code>true</code>，此属性可以通过直接写入表而不向数据库日志写入跟踪来提高 Oracle 目标数据库的提交率。有关更多信息，请参阅<a href="#">直接加载 INSERT</a>。此属性仅在您将 <code>UseDirectPathFullLoad</code> 设置为 <code>Y</code> 时才适用。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true/false</code></p> <p>例如：<code>--oracle-settings '{"DirectPathNoLog": true}'</code></p>

名称	描述
CharLengthSemantics	<p>指定字符列的长度是以字节数为单位还是以字符数为单位。要指示字符列长度以字符数为单位，请将此属性设置为 CHAR。否则，字符列长度以字节数为单位。</p> <p>默认值：未设置为 CHAR</p> <p>有效值：CHAR</p> <p>例如：<code>--oracle-settings '{"CharLengthSemantics": "CHAR"}'</code></p>
AlwaysReplaceEmptyString	<p>AWS DMS 在迁移到 Oracle 目标时，会添加一个额外的空间来复制空字符串。通常，Oracle 没有表示空字符串的符号。在 varchar2 上插入空字符串时，会将空字符串加载为 NULL。如果要在 Oracle 上将数据作为 NULL 插入，请将此属性设置为 FALSE。</p> <p>默认值：true</p> <p>有效值：true/false</p> <p>例如：<code>--oracle-settings '{"AlwaysReplaceEmptyString": false}'</code></p>

## Oracle 的目标数据类型

与一起使用的目标 Oracle 数据库 AWS DMS 支持大多数 Oracle 数据类型。下表显示了使用时支持的 Oracle 目标数据类型 AWS DMS 以及 AWS DMS 数据类型的默认映射。有关如何查看从源映射的数据类型的更多信息，请参阅有关所使用的源的部分。

AWS DMS 数据类型	Oracle 数据类型
BOOLEAN	NUMBER (1)
BYTES	RAW (length)
DATE	DATETIME

AWS DMS 数据类型	Oracle 数据类型
TIME	TIMESTAMP (0)
DATETIME	TIMESTAMP (scale)
INT1	NUMBER (3)
INT2	NUMBER (5)
INT4	NUMBER (10)
INT8	NUMBER (19)
NUMERIC	NUMBER (p,s)
REAL4	FLOAT
REAL8	FLOAT
STRING	<p>带日期指示 : DATE</p> <p>带时间指示 : TIMESTAMP</p> <p>带时间戳指示 : TIMESTAMP</p> <p>带 timestamp_with_timezone 指示 : TIMESTAMP WITH TIMEZONE</p> <p>带 timestamp_with_local_timezone 指示 : TIMESTAMP WITH LOCAL TIMEZONE 带 interval_year_to_month 指示 : INTERVAL YEAR TO MONTH</p> <p>带 interval_day_to_second 指示 : INTERVAL DAY TO SECOND</p> <p>如果 length &gt; 4000 : CLOB</p> <p>在所有其他情况下 : VARCHAR2 (length)</p>
UINT1	NUMBER (3)
UINT2	NUMBER (5)

AWS DMS 数据类型	Oracle 数据类型
UINT4	NUMBER (10)
UINT8	NUMBER (19)
WSTRING	<p>如果 length &gt; 2000 : NCLOB</p> <p>在所有其他情况下 : NVARCHAR2 (length)</p>
BLOB	<p>BLOB</p> <p>要将此数据类型与一起使用 AWS DMS，必须为特定任务启用 BLOB。仅包含主键的表中支持 BLOB 数据类型</p>
CLOB	<p>CLOB</p> <p>要将此数据类型与一起使用 AWS DMS，必须为特定任务启用 CLOB。在更改数据捕获 (CDC) 期间，仅包含主键的表支持 CLOB 数据类型。</p> <p>string</p> <p>源上声明的大小大于 4000 字节的 Oracle VARCHAR2 数据类型通过 AWS DMS CLOB 映射到 Oracle 目标上的字符串。</p>
NCLOB	<p>NCLOB</p> <p>要将此数据类型与一起使用 AWS DMS，必须允许将 nCLOB 用于特定任务。在 CDC 期间，仅包含主键的表中支持 NCLOB 数据类型。</p> <p>WSTRING</p> <p>源上声明的大小大于 4000 字节的 Oracle VARCHAR2 数据类型通过 AWS DMS NCLOB 映射到 Oracle 目标上的 WSTRING。</p>
XMLTYPE	<p>XMLTYPE 目标数据类型仅在 Oracle 到 Oracle 复制任务中相关。</p> <p>当源数据库是 Oracle 时，源数据类型将按“原样”复制到 Oracle 目标。例如，源上的 XMLTYPE 数据类型将创建为目标上的 XMLTYPE 数据类型。</p>

# 将 Microsoft SQL Server 数据库作为 AWS Database Migration Service 的目标

您可以使用 AWS DMS 将数据迁移到 Microsoft SQL Server 数据库。将 SQL Server 数据库作为目标时，您可以从另一个 SQL Server 数据库或其他支持的数据库之一迁移数据。

有关 AWS DMS 支持作为目标的 SQL Server 版本的信息，请参阅[的目标 AWS DMS](#)。

AWS DMS 支持 Enterprise、Standard、Workgroup 和 Developer 版的本地和 Amazon RDS 版本。

有关使用 AWS DMS 和 SQL Server 目标数据库的其他详细信息，请参阅以下内容。

## 主题

- [将 SQL Server 作为 AWS Database Migration Service 目标的限制](#)
- [将 SQL Server 作为 AWS Database Migration Service 目标时的安全要求](#)
- [使用 SQL Server 作为 AWS DMS 的目标时的端点设置](#)
- [Microsoft SQL Server 的目标数据类型](#)

## 将 SQL Server 作为 AWS Database Migration Service 目标的限制

将 SQL Server 数据库作为 AWS DMS 目标时，存在以下限制：

- 在您手动创建带有计算列的 SQL Server 目标表时，如果使用 BCP 批量复制实用程序，则不支持完全加载复制。要使用完全加载复制，请通过在端点上设置额外连接属性 (ECA) 'useBCPFullLoad=false' 来禁用 BCP 加载。有关在端点上设置 ECA 的信息，请参阅[创建源和目标终端节点](#)。有关使用 BCP 的更多信息，请参阅[Microsoft SQL Server 文档](#)。
- 在复制带有 SQL Server 空间数据类型 ( GEOMETRY 和 GEOGRAPHY ) 的表时，AWS DMS 使用默认空间引用标识符 (SRID) 替换所有您可能插入的 SRID。对于 GEOMETRY，默认 SRID 为 0，对于 GEOGRAPHY，则为 4326。
- 不支持临时表。在事务性应用模式下，如果在目标中手动创建这些表，则迁移临时表可能适用于仅复制任务。
- 目前，PostgreSQL 源中的 boolean 数据类型会作为具有不一致值的 bit 数据类型迁移到 SQLServer 目标。

作为解决方法，请执行下列操作：

- 使用该列的 VARCHAR(1) 数据类型预先创建表 ( 或者让 AWS DMS 创建表 )。然后，让下游处理将“F”视为 False，将“T”视为 True。

- 为避免出现必须更改下游处理的情况，请在任务中添加转换规则，将“F”值更改为“0”，将“T”值更改为 1，并将它们存储为 SQL Server 位数据类型。
- AWS DMS 不支持通过更改处理来设置列的可空性（使用带 ALTER COLUMN [SET|DROP] NOT NULL 子句的 ALTER TABLE 语句）。
- 不支持 Windows 身份验证。

## 将 SQL Server 作为 AWS Database Migration Service 目标时的安全要求

下面介绍将 AWS DMS 和 Microsoft SQL Server 目标结合使用时的安全要求：

- AWS DMS 用户账户必须在要连接到的 SQL Server 数据库上至少具有 db\_owner 用户角色。
- SQL Server 系统管理员必须向所有 AWS DMS 用户账户提供此权限。

## 使用 SQL Server 作为 AWS DMS 的目标时的端点设置

您可以使用端点设置来配置 SQL Server 目标数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者在 [AWS CLI](#) 中使用符合 `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令。

下表显示了将 SQL Server 作为目标时您可以使用的端点设置。

名称	描述
ControlTablesFileGroup	<p>为 AWS DMS 内部表指定文件组。当复制任务开始时，将在指定的文件组上创建所有内部 AWS DMS 控制表（awsdms_apply_exception、awsdms_apply、awsdms_changes）。</p> <p>默认值：无</p> <p>有效值：字符串</p> <p>例如：<code>--microsoft-sql-server-settings '{"ControlTablesFileGroup": "filegroup1"}'</code></p> <p>以下是用于创建文件组的命令的示例。</p>

名称	描述
	<pre>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>使用此额外连接属性 (ECA) 设置 SQL Server 实例的客户端语句超时时间 (以秒为单位)。默认值为 60 秒。</p> <p>例如: '{"ExecuteTimeout": 100}'</p>
UseBCPFullLoad	<p>使用该属性为使用 BCP 的完全加载操作传输数据。如果目标表包含的身份列在源表中不存在, 必须禁用使用 BCP 加载表选项。</p> <p>默认值: true</p> <p>有效值: true/false</p> <p>例如: --microsoft-sql-server-settings '{"UseBCPFullLoad": false}'</p>

## Microsoft SQL Server 的目标数据类型

下表显示了使用 AWS DMS 时支持的 Microsoft SQL Server 目标数据类型以及来自 AWS DMS 数据类型的默认映射。有关 AWS DMS 数据类型的其他信息, 请参阅 [AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	SQL Server 数据类型
BOOLEAN	TINYINT
BYTES	VARBINARY(length)
DATE	<p>对于 SQL Server 2008 及更高版本，请使用 DATE。</p> <p>对于早期版本，如果小数位数等于或小于 3，请使用 DATETIME。在所有其他情况下，请使用 VARCHAR (37)。</p>
TIME	<p>对于 SQL Server 2008 及更高版本，请使用 DATETIME2 (%d)。</p> <p>对于早期版本，如果小数位数等于或小于 3，请使用 DATETIME。在所有其他情况下，请使用 VARCHAR (37)。</p>
DATETIME	<p>对于 SQL Server 2008 及更高版本，请使用 DATETIME2 (精度)。</p> <p>对于早期版本，如果小数位数等于或小于 3，请使用 DATETIME。在所有其他情况下，请使用 VARCHAR (37)。</p>
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	FLOAT
STRING	<p>如果列是日期或时间列，请执行以下操作：</p> <ul style="list-style-type: none"> <li>对于 SQL Server 2008 及更高版本，请使用 DATETIME2。</li> <li>对于早期版本，如果小数位数等于或小于 3，请使用 DATETIME。在所有其他情况下，请使用 VARCHAR (37)。</li> </ul>



AWS DMS 数据类型	SQL Server 数据类型
	如果列不是日期或时间列，请使用 VARCHAR (length)。
UINT1	TINYINT
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR (length)
BLOB	VARBINARY(max)  IMAGE  要将此数据类型用于 AWS DMS，必须允许对特定任务使用 BLOB。AWS DMS 仅在包含主键的表中支持 BLOB 数据类型。
CLOB	VARCHAR(max)  要将此数据类型用于 AWS DMS，必须允许对特定任务使用 CLOB。在更改数据捕获 (CDC) 期间，AWS DMS 仅在包含主键的表中支持 CLOB 数据类型。
NCLOB	NVARCHAR(max)  要将此数据类型用于 AWS DMS，必须允许对特定任务使用 NCLOB。在 CDC 期间，AWS DMS 仅在包含主键的表中支持 NCLOB 数据类型。

## 将 PostgreSQL 数据库作为 AWS Database Migration Service 的目标

您可以使用将数据从另一个 PostgreSQL AWS DMS L 数据库或其他支持的数据库迁移到 PostgreSQL 数据库。

有关支持作为目标的 PostgreSQL 版本的信息 AWS DMS，请参阅 [的目标 AWS DMS](#)

**Note**

- 亚马逊 Aurora Serverless 可用作兼容 PostgreSQL 的亚马逊 Aurora 的目标。有关亚马逊 Aurora Serverless 的更多信息，请参阅亚马逊 Aurora 用户指南中的[使用亚马逊 Aurora Serverless v2](#)。
- Aurora Serverless 数据库集群只能从 Amazon VPC 访问，不能使用[公有 IP 地址](#)。因此，如果您要在不同于 Aurora PostgreSQL Serverless 的其他区域创建复制实例，则必须配置[VPC 对等连接](#)。否则，请检查 Aurora PostgreSQL Serverless [区域](#)的可用性，并决定将其中一个区域用于 Aurora PostgreSQL Serverless 和您的复制实例。
- Babelfish 功能内置在 Amazon Aurora 中，无需支付额外费用。有关更多信息，请参阅[将适用于 Aurora PostgreSQL 的 Babelfish 作为 AWS Database Migration Service 的目标](#)。

AWS DMS 在满载阶段将数据从源迁移到目标时 table-by-table 采用了一种方法。无法在完全加载阶段保证表顺序。在完全加载阶段以及应用各个表的缓存事务时，表将不同步。因此，活动引用完整性约束可能会导致任务在完全加载阶段失败。

在 PostgreSQL 中，将使用触发器实施外键 (引用完整性约束)。在满载阶段，每次 AWS DMS 加载一个表。强烈建议您在完全加载期间使用以下方法之一禁用外键约束：

- 从实例中临时禁用所有触发器并完成完全加载。
- 在 PostgreSQL 中使用 `session_replication_role` 参数。

在任何给定时间，触发器可能处于以下状态之一：`origin`、`replica`、`always` 或 `disabled`。在将 `session_replication_role` 参数设置为 `replica` 时，只有处于 `replica` 状态的触发器是活动触发器，并且它们在被调用时触发。否则，触发器保持非活动状态。

PostgreSQL 具有故障保护机制以防止截断表，即使设置了 `session_replication_role`。您可以将该机制作为禁用触发器的替代方法，以帮助完成完全加载运行。为此，请将目标表准备模式设置为 `DO_NOTHING`。否则，在具有外键约束时，`DROP` 和 `TRUNCATE` 操作将失败。

在 Amazon RDS 中，可以使用参数组控制该参数的设置。对于在 Amazon EC2 上运行的 PostgreSQL 实例，可以直接设置该参数。

有关使用 PostgreSQL 数据库作为目标的更多详细信息，AWS DMS 请参阅以下各节：

## 主题

- [使用 PostgreSQL 作为目标的限制 AWS Database Migration Service](#)
- [使用 PostgreSQL 数据库作为目标时的安全要求 AWS Database Migration Service](#)
- [使用 PostgreSQL 作为目标时的端点设置和额外连接属性 \(ECA\) AWS DMS](#)
- [PostgreSQL 的目标数据类型](#)
- [使用 Babelfish for Aurora PostgreSQL 作为目标 AWS Database Migration Service](#)

## 使用 PostgreSQL 作为目标的限制 AWS Database Migration Service

将 PostgreSQL 数据库作为 AWS DMS 的目标时，存在以下限制：

- 对于异构迁移，JSON 数据类型将在内部转换为原生 CLOB 数据类型。
- 在 Oracle 到 PostgreSQL 的迁移中，如果 Oracle 中的一列包含空字符（十六进制值 U+0000）AWS DMS，则将空字符转换为空格（十六进制值 U+0020）。这是出于 PostgreSQL 限制的原因。
- AWS DMS 不支持复制到使用合并函数创建的具有唯一索引的表。
- 如果您的表使用序列，则在停止从源数据库 NEXTVAL 进行复制后，请更新目标数据库中每个序列的值。AWS DMS 复制源数据库中的数据，但在正在进行的复制过程中不会将序列迁移到目标数据库。

## 使用 PostgreSQL 数据库作为目标时的安全要求 AWS Database Migration Service

出于安全目的，用于数据迁移的用户账户，必须是您作为目标的任意 PostgreSQL 数据库中的注册用户。

您的 PostgreSQL 目标端点需要最低的用户权限才能运行迁移，请参阅以下示 AWS DMS 例。

```
CREATE USER newuser WITH PASSWORD 'your-password';
ALTER SCHEMA schema_name OWNER TO newuser;
```

或者，

```
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;
```

## 使用 PostgreSQL 作为目标时的端点设置和额外连接属性 (ECA) AWS DMS

您可以使用端点设置和额外连接属性 (ECA) 来配置您的 PostgreSQL 目标数据库。

您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者使用中的 `create-endpoint` 命令和 `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON 语法。[AWS CLI](#)

您可以使用终端节点的 `ExtraConnectionAttributes` 参数指定 ECA。

下表显示了您可以使用 PostgreSQL 作为目标的端点设置。

名称	描述
MaxFileSize	<p>指定用于将数据传输到 PostgreSQL 的任何 .csv 文件的最大大小 (以 KB 为单位)。</p> <p>默认值 : 32768 KB ( 32 MB )</p> <p>有效值 : 1-1048576 KB ( 最大 1.1 GB )</p> <p>例如 : <code>--postgre-sql-settings '{"MaxFileSize": 512}'</code></p>
ExecuteTimeout	<p>设置 PostgreSQL 实例的客户端语句超时 (以秒为单位)。默认值为 60 秒。</p> <p>例如 : <code>--postgre-sql-settings '{"ExecuteTimeout": 100}'</code></p>
AfterConnectScript= SET session_replication_role = replica	<p>此属性具有 AWS DMS 绕过外键和用户触发器，可缩短批量加载数据所需的时间。</p>
MapUnboundedNumericAsString	<p>此参数将具有无界 NUMERIC 数据类型的列视为 STRING，以便在不损失数值精度的前提下成功迁移。此</p>

名称	描述
	<p>参数仅用于从 PostgreSQL 源复制到 PostgreSQL 目标或与 PostgreSQL 兼容的数据库。</p> <p>默认值：false</p> <p>有效值：false/true</p> <p>例如：<code>--postgre-sql-settings '{"MapUnboundedNumericAsString": "true"}'</code></p> <p>使用此参数可能会导致某些复制性能下降，因为需要从数字转换为字符串，然后再转换回数字。DMS 版本 3.4.4 及更高版本支持使用此参数</p> <div data-bbox="688 800 1507 1352" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>只能同时在 PostgreSQL 源端点和目标端点中使用 <code>MapUnboundedNumericAsString</code>。在 CDC 期间，在源 PostgreSQL 端点上使用 <code>MapUnboundedNumericAsString</code> 会将精度限制在 28 以内。在目标端点上使用 <code>MapUnboundedNumericAsString</code> 时，使用精度 28、小数位数 6 迁移数据。请勿将 <code>MapUnboundedNumericAsString</code> 与非 PostgreSQL 目标一起使用。</p> </div>
loadUsingCSV	<p>使用此额外连接属性 (ECA) 通过 \COPY 命令传输用于满载操作的数据。</p> <p>默认值：true</p> <p>有效值：true/false</p> <p>ECA 示例：<code>loadUsingCSV=true;</code></p> <p>注意：由于直接执行 INSERT，将此 ECA 设置为 false 可能会导致某些复制性能下降。</p>

名称	描述
DatabaseMode	<p>使用此属性可以更改复制过程在处理需要一些额外配置的 Postgresql 兼容端点（例如 Babelfish 端点）时的默认行为。</p> <p>默认值：DEFAULT</p> <p>有效值：DEFAULT、BABELFISH</p> <p>例如：DatabaseMode=default;</p>
BabelfishDatabaseName	<p>使用此属性指定要迁移到的目标 Babelfish T-SQL 数据库的名称。如果 DatabaseMode 设置为 Babelfish，则此项为必填。这不是保留的 babelfish_db 数据库。</p> <p>例如：BabelfishDatabaseName=TargetDb;</p>

## PostgreSQL 的目标数据类型

的 PostgreSQL 数据库端点 AWS DMS 支持大多数 PostgreSQL 数据库数据类型。下表显示了 AWS DMS 使用时支持的 PostgreSQL 数据库目标数据类型以及数据类型的默认映射 AWS DMS。

有关 AWS DMS 数据类型的更多信息，请参见[AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	PostgreSQL 数据类型
BOOLEAN	BOOLEAN
BLOB	BYTEA
BYTES	BYTEA
DATE	DATE
TIME	TIME
DATETIME	<p>如果小数位数介于 0 和 6 之间，请使用 TIMESTAMP。</p> <p>如果小数位数介于 7 和 9 之间，请使用 VARCHAR (37)。</p>

AWS DMS 数据类型	PostgreSQL 数据类型
INT1	SMALLINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (P,S)
REAL4	FLOAT4
REAL8	FLOAT8
string	<p>如果长度介于 1 和 21845 之间，请使用 VARCHAR (以字节为单位的长度)。</p> <p>如果长度介于 21846 和 2147483647 之间，请使用 VARCHAR (65535)。</p>
UINT1	SMALLINT
UINT2	INTEGER
UINT4	BIGINT
UINT8	BIGINT
WSTRING	<p>如果长度介于 1 和 21845 之间，请使用 VARCHAR (以字节为单位的长度)。</p> <p>如果长度介于 21846 和 2147483647 之间，请使用 VARCHAR (65535)。</p>
NCLOB	TEXT
CLOB	TEXT

**Note**

从 PostgreSQL 源进行复制时 AWS DMS，除了具有用户定义数据类型的列之外，会为所有列创建具有相同数据类型的目标表。在此类情况下，在目标中创建数据类型作为“可变字符”。

## 使用 Babelfish for Aurora PostgreSQL 作为目标 AWS Database Migration Service

可以使用 AWS Database Migration Service 将 SQL Server 源表迁移到适用于 Amazon Aurora PostgreSQL 的 Babelfish 目标。有了 Babelfish，Aurora PostgreSQL 可以理解 Microsoft SQL Server 的专有 SQL 语言 T-SQL，并支持相同的通信协议。因此，为 SQL Server 编写的应用程序现在只需更改更少的代码，即可与 Aurora 配合使用。Babelfish 功能内置在 Amazon Aurora 中，无需支付额外费用。可以通过 Amazon RDS 控制台在 Amazon Aurora 集群上激活 Babelfish。

使用 AWS DMS 控制台、API 或 CLI 命令创建 AWS DMS 目标终端节点时，将目标引擎指定为 Amazon Aurora PostgreSQL，并将数据库命名为 `babelfish_db`。在端点设置部分，添加相关设置以将 `DatabaseMode` 设置为 `Babelfish`，并将 `BabelfishDatabaseName` 设置为 Babelfish T-SQL 数据库的名称。

### 向迁移任务添加转换规则

在为 Babelfish 目标定义迁移任务时，需要包含转换规则，以确保 DMS 使用目标数据库中预先创建的 T-SQL Babelfish 表。

首先，在迁移任务中添加一个转换规则，使所有表名都变为小写。Babelfish 将您使用 T-SQL 创建的表的名称以小写形式存储在 PostgreSQL `pg_class` 目录中。但是，当您的 SQL Server 表具有大小写混合的名称时，DMS 会使用 PostgreSQL 原生数据类型而不是与 T-SQL 兼容的数据类型来创建表。因此，请务必添加一个转换规则，使所有表名都变为小写。请注意，不应将列名称转换为小写。

接下来，如果您在定义集群时使用了多数据库迁移模式，请添加重命名原始 SQL Server 架构的转换规则。务必重命名 SQL Server 架构名称，使之包含 T-SQL 数据库的名称。例如，如果原始 SQL Server 架构名称为 `dbo`，而您的 T-SQL 数据库名称为 `mydb`，则使用转换规则将架构重命名为 `mydb_dbo`。

如果您使用单数据库模式，则不需要转换规则即可重命名架构名称。架构名称与 Babelfish one-to-one 模式中的目标 T-SQL 数据库存在映射。

以下转换规则示例将所有表名设置为小写，并将原始 SQL Server 架构名称从 `dbo` 重命名为 `mydb_dbo`。

```
{
```



```
"rules": [  
  {  
    "rule-type": "transformation",  
    "rule-id": "566251737",  
    "rule-name": "566251737",  
    "rule-target": "schema",  
    "object-locator": {  
      "schema-name": "dbo"  
    },  
    "rule-action": "rename",  
    "value": "mydb_dbo",  
    "old-value": null  
  },  
  {  
    "rule-type": "transformation",  
    "rule-id": "566139410",  
    "rule-name": "566139410",  
    "rule-target": "table",  
    "object-locator": {  
      "schema-name": "%",  
      "table-name": "%"  
    },  
    "rule-action": "convert-lowercase",  
    "value": null,  
    "old-value": null  
  },  
  {  
    "rule-type": "selection",  
    "rule-id": "566111704",  
    "rule-name": "566111704",  
    "object-locator": {  
      "schema-name": "dbo",  
      "table-name": "%"  
    },  
    "rule-action": "include",  
    "filters": []  
  }  
]  
}
```

## 使用包含 Babelfish 表的 PostgreSQL 目标端点的限制

使用包含 Babelfish 表的 PostgreSQL 目标端点时，存在以下限制：

- 对于目标表准备模式，仅使用不执行任何操作或截断模式。不要使用删除目标中的表模式。在该模式下，DMS 会将这些表创建为 T-SQL 可能无法识别的 PostgreSQL 表。
- AWS DMS 不支持 `sql_variant` 数据类型。
- Babelfish 不支持 `HEIRARCHYID`、`GEOMETRY` 和 `GEOGRAPHY` 数据类型。要迁移这些数据类型，您可以添加转换规则以将数据类型转换为 `wstring(250)`。
- Babelfish 仅支持使用 `BYTEA` 数据类型迁移 `BINARY`、`VARBINARY` 和 `IMAGE` 数据类型。对于早期版本的 Aurora PostgreSQL，可以使用 DMS 将这些表迁移到 [Babelfish 目标端点](#)。您不必为 `BYTEA` 数据类型指定长度，如以下示例所示。

```
[Picture] [VARBINARY](max) NULL
```

将前面的 T-SQL 数据类型更改为 T-SQL 支持的 `BYTEA` 数据类型。

```
[Picture] BYTEA NULL
```

- 对于早期版本的 Aurora PostgreSQL Babelfish，如果您使用 PostgreSQL 目标端点创建用于从 SQL Server 持续复制到 Babelfish 的迁移任务，则需要为任何使用 `IDENTITY` 列的表分配 `SERIAL` 数据类型。从 Aurora PostgreSQL（版本 15.3/14.8 及更高版本）和 Babelfish（版本 3.2.0 及更高版本）开始，支持标识列，并且不再需要分配 `SERIAL` 数据类型。有关更多信息，请参阅《SQL Server 到 Aurora PostgreSQL 迁移行动手册》的“序列和标识”部分中的 [SERIAL 使用情况](#)。然后，当您在 Babelfish 中创建表时，需修改如下的列定义。

```
[IDCol] [INT] IDENTITY(1,1) NOT NULL PRIMARY KEY
```

将前面的内容改为以下内容。

```
[IDCol] SERIAL PRIMARY KEY
```

兼容 Babelfish 的 Aurora PostgreSQL 使用默认配置创建序列并向该列添加 `NOT NULL` 约束。新创建的序列的行为类似于常规序列（以 1 为增量），并且没有复合 `SERIAL` 选项。

- 使用包含 `IDENTITY` 列或 `SERIAL` 数据类型的表迁移数据后，根据列的最大值重置基于 PostgreSQL 的序列对象。对表执行完全加载后，使用以下 T-SQL 查询生成语句以便设置关联序列对象的种子。

```
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(%)';

```

该查询会生成一系列 SELECT 语句，您可以执行这些语句来更新 IDENTITY 和 SERIAL 的最大值。

- 对于 3.2 之前的 Babelfish 版本，完整 LOB 模式可能会导致表错误。如果发生这种情况，请为加载失败的表创建一个单独任务。然后，使用受限 LOB 模式为最大 LOB 大小 (KB) 指定相应的值。另一种选择是设置 SQL Server 端点连接属性设置 ForceFullLob=True。
- 对于 3.2 之前的 Babelfish 版本，如果对不使用基于整数的主键的 Babelfish 表执行数据验证，则会生成一条消息，指出找不到合适的唯一键。从 Aurora PostgreSQL (版本 15.3/14.8 及更高版本) 和 Babelfish (版本 3.2.0 及更高版本) 开始，支持对非整数主键进行数据验证。
- 由于秒的小数位数存在精度差异，DMS 会报告使用 DATETIME 数据类型的 Babelfish 表的数据验证失败。为避免出现这类失败，可以为 DATETIME 数据类型添加以下验证规则类型。

```

{
    "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"  
}
```

## 将与 MySQL 兼容的数据库作为 AWS Database Migration Service 的目标

您可以使用任何支持的源数据引擎将数据迁移到任何与 MySQL 兼容的数据库。AWS DMS 如果您要迁移到与 MySQL 兼容的本地数据库，则 AWS DMS 需要您的源引擎位于生态系统中。AWS 引擎可以安装在 AWS 托管服务上，例如亚马逊 RDS、Amazon Aurora 或 Amazon S3。或者，该引擎也可以位于 Amazon EC2 上的自管理数据库中。

您可以使用 SSL 来加密 MySQL 兼容终端节点与复制实例之间的连接。有关将 SSL 用于 MySQL 兼容终端节点的更多信息，请参阅[将 SSL 与 AWS Database Migration Service](#)。

有关 AWS DMS 支持作为目标的 MySQL 版本的信息，请参阅[的目标 AWS DMS](#)。

您可以使用以下与 MySQL 兼容的数据库作为目标：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

无论使用什么源存储引擎（MyISAM、MEMORY 等），默认情况下，AWS DMS 都会创建与 MySQL 兼容的目标表作为 InnoDB 表。

如果您需要 InnoDB 之外的存储引擎中的表，则可以在与 MySQL 兼容的目标上手动创建表，然后使用 Do nothing (不执行任何操作) 选项来迁移表。有关更多信息，请参阅[完全加载任务设置](#)。

有关将与 MySQL 兼容的数据库作为 AWS DMS 的目标的更多详细信息，请参阅以下章节。

## 主题

- [使用任何与 MySQL 兼容的数据库作为目标 AWS Database Migration Service](#)
- [使用与 MySQL 兼容的数据库作为目标的限制 AWS Database Migration Service](#)
- [使用与 MySQL 兼容的数据库作为目标时的终端节点设置 AWS DMS](#)
- [MySQL 的目标数据类型](#)

## 使用任何与 MySQL 兼容的数据库作为目标 AWS Database Migration Service

在使用与 MySQL 兼容的数据库作为 AWS DMS 的目标之前，请确保完成以下先决条件：

- 为其提供对兼容 MySQL 的数据库具有读/写权限的用户帐户。要创建所需的权限，请运行以下命令。

```
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>'@'%' ;
```

- 在完全加载迁移阶段，您必须禁用目标表上的外键。要在满负荷期间对兼容 MySQL 的数据库禁用外键检查，可以在目标端点的 AWS DMS 控制台的“额外连接属性”部分中添加以下命令。

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

- 设置数据库参数 `local_infile = 1` 以启用 AWS DMS，以便将数据加载到目标数据库。

## 使用与 MySQL 兼容的数据库作为目标的限制 AWS Database Migration Service

使用 MySQL 数据库作为目标时，AWS DMS 不支持以下内容：

- 数据定义语言 (DDL) 语句 TRUNCATE PARTITION、DROP TABLE 和 RENAME TABLE。
- 使用 ALTER TABLE *table\_name* ADD COLUMN *column\_name* 语句将列添加到表的开头或中间。

- 在满载任务中将数据加载到兼容 MySQL 的目标时，AWS DMS 不会报告任务日志中的限制导致的错误，这些错误可能会导致重复的键错误或与记录数量不匹配。这是由 MySQL 使用 LOAD DATA 命令处理本地数据的方式造成的。务必在完全加载阶段执行以下操作：
  - 禁用限制
  - 使用 AWS DMS 验证来确保数据的一致性。
- 当您将列值更新为其现有值时，与 MySQL 兼容的数据库会返回 0 rows affected 警告。尽管此行为在技术上没有错误，但它与其他数据库引擎应对这种情况的方式不同。例如，Oracle 执行一个行的更新。对于与 MySQL 兼容的数据库，在 awsdms\_apply\_exceptions 控制表中 AWS DMS 生成一个条目并记录以下警告。

Some changes from the source database had no impact when applied to the target database. See awsdms\_apply\_exceptions table for details.

- Aurora Serverless 可用作 Amazon Aurora 版本 2 的目标，且与 MySQL 版本 5.7 兼容。（选择 Aurora MySQL 版本 2.07.1 以便能够使用 Aurora Serverless，且与 MySQL 5.7 兼容。）有关 Aurora Serverless 的更多信息，请参阅亚马逊 [Aurora 用户指南中的使用 Aurora Serverless v2](#)。
- AWS DMS 不支持使用适用于 Aurora 或 Amazon RDS 的读取器终端节点，除非实例处于可写模式，即 read\_only 和 innodb\_read\_only 参数设置为 0 或 OFF。有关使用 Amazon RDS 和 Aurora 作为目标的更多信息，请参见以下内容：
  - [确定您连接到的数据库实例](#)
  - [使用 MySQL 更新只读副本](#)

## 使用与 MySQL 兼容的数据库作为目标时的终端节点设置 AWS DMS

您可以使用端点设置来配置与 MySQL 兼容的目标数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者使用中的 create-endpoint 命令和 --my-sql-settings '{"EndpointSetting": "value", ...}' JSON 语法。 [AWS CLI](#)

下表显示了您可以配合使用 MySQL 作为目标的端点设置。

名称	描述
TargetDbType	指定将目标上的源表迁移到的位置 (单个数据库或多个数据库)。如果指定 SPECIFIC_DATABASE，则需要

名称	描述
	<p>使用 AWS CLI 或时指定数据库名称 AWS Management Console。</p> <p>默认值：MULTIPLE_DATABASES</p> <p>有效值：{SPECIFIC_DATABASE、MULTIPLE_DATABASES }</p> <p>例如：--my-sql-settings '{"TargetDbType": "MULTIPLE_DATABASES"}'</p>
ParallelLoadThreads	<p>改进了将数据加载到与 MySQL 兼容的目标数据库时的性能。指定用于将数据加载到与 MySQL 兼容的目标数据库中的线程的数目。设置大量线程可能会对数据库性能产生不利影响，因为每个线程均需要一个单独的连接。</p> <p>默认值：1</p> <p>有效值：1-5</p> <p>例如：--my-sql-settings '{"ParallelLoadThreads": 1}'</p>
AfterConnectScript	<p>指定脚本在 AWS DMS 连接到终端节点后立即运行。</p> <p>例如，您可以指定与 MySQL 兼容的目标应将收到的语句转换为 latin1 字符集，该字符集是数据库的默认编入字符集。此参数通常可在从 UTF8 客户端转换时提高性能。</p> <p>例如：--my-sql-settings '{"AfterConnectScript": "SET character_set_connection='latin1'"}'</p>

名称	描述
MaxFileSize	<p>指定用于将数据传输到与 MySQL 兼容的数据库的任何 .csv 文件的最大大小 ( 以 KB 为单位 )。</p> <p>默认值 : 32768 KB ( 32 MB )</p> <p>有效值 : 1-1048576</p> <p><code>--my-sql-settings '{"MaxFileSize": 512}'</code></p>
CleanSrcMetadataOnMismatch	<p>出现不匹配情况时在复制实例上清除并重新创建表元数据信息。例如，对表运行更改 DDL 语句可能会产生有关在复制实例中缓存的表的不同信息。布尔值。</p> <p>默认值 : false</p> <p>例如 : <code>--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'</code></p>

您也可以使用额外的连接属性配置与 MySQL 兼容的目标数据库。

下表显示了将 MySQL 作为目标时可以使用的额外连接属性。

名称	描述
Initstmt=SET FOREIGN_KEY_CHECKS=0;	<p>禁用外键检查。</p> <p>例如 : <code>--extra-connection-attributes "Initstmt=SET FOREIGN_KEY_CHECKS=0;"</code></p>
Initstmt=SET time_zone	<p>指定与 MySQL 兼容的目标数据库的时区。</p> <p>默认值 : UTC</p> <p>有效值 : 目标 MySQL 数据库中可用的时区名称。</p> <p>例如 : <code>--extra-connection-attributes "Initstmt=SET time_zone= <i>US/Pacific</i> ;"</code></p>



或者，您可以使用 `--my-sql-settings` 命令的 `AfterConnectScript` 参数来禁用外键检查并为数据库指定时区。

## MySQL 的目标数据类型

下表显示了使用时支持的 MySQL 数据库目标数据类型 AWS DMS 以及 AWS DMS 数据类型的默认映射。

有关 AWS DMS 数据类型的更多信息，请参见 [AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	MySQL 数据类型
BOOLEAN	BOOLEAN
BYTES	如果长度介于 1 和 65535 之间，请使用 VARBINARY (length)。  如果长度介于 65536 和 2147483647 之间，请使用 LONGLOB。
DATE	DATE
TIME	TIME
TIMESTAMP	“如果小数位数 => 0 且 <= 6，请使用 DATETIME (Scale)  如果小数位数 => 7 且 <= 9，请使用 VARCHAR (37)”
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (p,s)
REAL4	FLOAT

AWS DMS 数据类型	MySQL 数据类型
REAL8	DOUBLE PRECISION
string	<p>如果长度介于 1 和 21845 之间，请使用 VARCHAR (length)。</p> <p>如果长度介于 21846 和 2147483647 之间，请使用 LONGTEXT。</p>
UINT1	UNSIGNED TINYINT
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	<p>如果长度介于 1 和 32767 之间，请使用 VARCHAR (length)。</p> <p>如果长度介于 32768 和 2147483647 之间，请使用 LONGTEXT。</p>
BLOB	<p>如果长度介于 1 和 65535 之间，请使用 BLOB。</p> <p>如果长度介于 65536 和 2147483647 之间，请使用 LONGBLOB。</p> <p>如果长度为 0，请使用 LONGBLOB (完全 LOB 支持)。</p>

AWS DMS 数据类型	MySQL 数据类型
NCLOB	<p>如果长度介于 1 和 65535 之间，请使用 TEXT。</p> <p>如果长度介于 65536 和 2147483647 之间，请将 LONGTEXT 与 ucs2 一起用于 CHARACTER SET。</p> <p>如果长度为 0，请将 LONGTEXT (完全 LOB 支持) 与 ucs2 一起用于 CHARACTER SET。</p>
CLOB	<p>如果长度介于 1 和 65535 之间，请使用 TEXT。</p> <p>如果长度介于 65,536 和 2147483647 之间，请使用 LONGTEXT。</p> <p>如果长度为 0，请使用 LONGTEXT (完全 LOB 支持)。</p>

## 将 Amazon Redshift 数据库作为 AWS Database Migration Service 目标

您可以使用 AWS Database Migration Service 将数据迁移到 Amazon Redshift 数据库。Amazon Redshift 是 Cloud 中的一种完全托管的 PB 级数据仓库服务。使用 Amazon Redshift 数据库作为目标，您可以从所有其他支持的源数据库迁移数据。

您可以将 Amazon Redshift Serverless 作为 AWS DMS 的目标。有关更多信息，请参阅下面的[使用 AWS DMS 并将 Amazon Redshift Serverless 作为目标](#)。

Amazon Redshift 集群必须与复制实例处于相同的 AWS 账户和相同的 AWS 区域中。

在将数据库迁移到 Amazon Redshift 期间，先将数据移动到 Amazon S3 存储桶。当文件位于 Amazon S3 存储桶中时，AWS DMS 随之将其传输到 Amazon Redshift 数据仓库中正确的表。AWS DMS 在与 Amazon Redshift 数据库相同的 AWS 区域中创建 S3 存储桶。AWS DMS 复制实例必须位于同一个 AWS 区域中。

如果使用 AWS CLI 或 DMS API 将数据迁移到 Amazon Redshift，请设置 AWS Identity and Access Management (IAM) 角色以允许 S3 访问。有关创建此 IAM 角色的更多信息，请参阅 [创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

Amazon Redshift 端点为以下过程提供完全的自动化：

- 架构生成和数据类型映射
- 源数据库表的完全加载
- 增量加载对源表进行的更改
- 应用对源表的数据定义语言 (DDL) 的架构更改
- 完全加载和更改数据捕获 (CDC) 流程之间的同步。

AWS Database Migration Service 支持完全加载和更改处理操作。AWS DMS 从源数据库读取数据并创建一系列逗号分隔值 (.csv) 文件。对于完全加载操作，AWS DMS 为每个表创建文件。然后，AWS DMS 将各个表的表文件复制到 Amazon S3 中的单独文件夹中。当文件上传到 Amazon S3 之后，AWS DMS 发送复制命令，将文件中的数据复制到 Amazon Redshift。对于更改处理操作，AWS DMS 将净更改复制到 .csv 文件。随后，AWS DMS 将净更改文件上传到 Amazon S3 并将数据复制到 Amazon Redshift。

有关将 Amazon Redshift 作为 AWS DMS 目标的其他详细信息，请参阅以下章节：

## 主题

- [将 Amazon Redshift 数据库作为 AWS Database Migration Service 目标的先决条件](#)
- [将 Redshift 作为目标所需的权限](#)
- [将 Amazon Redshift 作为 AWS Database Migration Service 目标时的限制](#)
- [将 Amazon Redshift 数据库配置为 AWS Database Migration Service 的目标](#)
- [将 Amazon Redshift 作为 AWS Database Migration Service 目标时使用增强型 VPC 路由](#)
- [创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密](#)
- [使用 Amazon Redshift 作为 AWS DMS 的目标时的端点设置](#)
- [使用数据加密密钥，并将 Amazon S3 存储桶作为中间存储](#)
- [Amazon Redshift 的多线程任务设置](#)
- [Amazon Redshift 的目标数据类型](#)
- [使用 AWS DMS 并将 Amazon Redshift Serverless 作为目标](#)

## 将 Amazon Redshift 数据库作为 AWS Database Migration Service 目标的先决条件

下面的列表介绍将 Amazon Redshift 作为数据迁移目标时所需的先决条件：

- 使用 AWS 管理控制台启动 Amazon Redshift 集群。请记下有关您的 AWS 账户和 Amazon Redshift 集群的基本信息，例如密码、用户名和数据库名称。在创建 Amazon Redshift 目标端点时，您需要这些值。
- Amazon Redshift 集群必须与复制实例处于相同的 AWS 账户和相同的 AWS 区域中。
- AWS DMS 复制实例需要与您集群所使用的 Amazon Redshift 端点（主机名和端口）的网络连接。
- AWS DMS 使用 Amazon S3 存储桶将数据传输到 Amazon Redshift 数据库。对于 AWS DMS 创建存储桶，控制台使用 IAM 角色 `dms-access-for-endpoint`。如果使用 AWS CLI 或 DMS API 创建数据库迁移并将 Amazon Redshift 作为目标数据库，则必须创建此 IAM 角色。有关创建此角色的更多信息，请参阅 [创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。
- AWS DMS 将 BLOB、CLOB 和 NCLOB 转换为目标 Amazon Redshift 实例上的 VARCHAR。Amazon Redshift 不支持大于 64 KB 的 VARCHAR 数据类型，因此您无法在 Amazon Redshift 上存储传统 LOB。
- 将目标元数据任务设置 `BatchApplyEnabled` 设为 `true`，以供 AWS DMS 在 CDC 期间处理对 Amazon Redshift 目标表的更改。同时需要源表和目标表的主键。如果没有主键，则会逐条语句应用更改。这会导致目标延迟并影响集群提交队列，从而对 CDC 期间的任务性能产生负面影响。

## 将 Redshift 作为目标所需的权限

使用 GRANT 命令为用户或用户组定义访问权限。权限包括各种访问选项，例如读取表和视图中的数据、写入数据和创建表的能力。有关为 Amazon Redshift 使用 GRANT 的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 [GRANT](#)。

以下是授予 Amazon Redshift 表和视图上的表、数据库、架构、函数、过程或语言级权限的语法。

```
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 } [, ...]

```

以下是 Amazon Redshift 表和视图上的列级权限的语法。

```

GRANT { { SELECT | UPDATE } ( column_name [, ...] ) [, ...] | ALL [ PRIVILEGES ]
  ( column_name [, ...] ) }
  ON { [ TABLE ] table_name [, ...] }
  TO { username | GROUP group_name | PUBLIC } [, ...]

```

以下是授予具有指定角色的用户和组的 ASSUMEROLE 权限的语法。

```

GRANT ASSUMEROLE
  ON { 'iam_role' [, ...] | ALL }
  TO { username | GROUP group_name | PUBLIC } [, ...]
  FOR { ALL | COPY | UNLOAD } [, ...]

```

## 将 Amazon Redshift 作为 AWS Database Migration Service 目标时的限制

将 Amazon Redshift 数据库作为目标时，存在以下限制：

- 不要为用作 Amazon Redshift 目标的中间存储的 S3 存储桶启用版本控制。如果您需要 S3 版本控制，请使用生命周期策略主动删除旧版本。否则，您可能会遇到由于 S3 list-object 调用超时而导致端点测试连接失败的情况。要为 S3 存储桶创建生命周期策略，请参阅[管理存储生命周期](#)。要删除 S3 对象的某个版本，请参阅[从启用版本控制的存储桶中删除对象版本](#)。
- 不支持以下 DDL：

```
ALTER TABLE table name MODIFY COLUMN column name data type;
```

- AWS DMS 无法迁移或复制对名称以下划线 ( \_ ) 开头的架构所做的更改。如果您具有名称以下划线开头的架构，请使用映射转换重命名目标上的架构。
- Amazon Redshift 不支持大于 64 KB 的 VARCHAR。传统数据库中的 LOB 不能存储在 Amazon Redshift 中。
- 当任意主键列名称使用了保留字时，不支持对具有多列主键的表应用 DELETE 语句。转到[此处](#)查看 Amazon Redshift 保留词的列表。
- 如果源系统对源表的主键执行 UPDATE 操作，则可能会遇到性能问题。这些性能问题会在对目标应用更改时出现。这是因为 UPDATE ( 和 DELETE ) 操作依赖于主键值来标识目标行。如果您更新源表的主键，则任务日志将包含类似如下的消息：

```
Update on table 1 changes PK to a PK that was previously updated in the same bulk update.
```

- 在为 Redshift 集群配置端点时，DMS 不支持自定义 DNS 名称，您需要使用 Amazon 提供的 DNS 名称。由于 Amazon Redshift 集群必须与复制实例位于相同的 AWS 账户和区域中，因此如果您使用自定义 DNS 端点，则验证将失败。
- Amazon Redshift 的默认空闲会话超时时间为 4 小时。当 DMS 复制任务中没有任何活动时，Redshift 会在 4 小时后断开会话连接。由于 DMS 无法连接并可能需要重新启动，因此会导致出现错误。一个临时解决方法是为 DMS 复制用户设置大于 4 小时的会话超时限制。或者，请参阅《Amazon Redshift 数据库开发人员指南》中的 [ALTER USER](#) 描述。
- 当 AWS DMS 在没有主键或唯一键的情况下复制源表数据时，CDC 延迟可能会很高，导致性能水平不可接受。

## 将 Amazon Redshift 数据库配置为 AWS Database Migration Service 的目标

AWS Database Migration Service 必须配置为使用 Amazon Redshift 实例。下表列出可用于 Amazon Redshift 端点的配置属性。

属性	描述
服务器	所使用的 Amazon Redshift 集群的名称。

属性	描述
port	Amazon Redshift 的端口号。默认值是 5439。
username	已注册用户的 Amazon Redshift 用户名。
password	在 username 属性中指定的用户的密码。
database	所使用的 Amazon Redshift 数据仓库 ( 服务 ) 的名称。

如果要添加额外的连接字符串属性到 Amazon Redshift 端点，可以指定 `maxFileSize` 和 `fileTransferUploadStreams` 属性。有关这些属性的更多信息，请参阅[使用 Amazon Redshift 作为 AWS DMS 的目标时的端点设置](#)。

## 将 Amazon Redshift 作为 AWS Database Migration Service 目标时使用增强型 VPC 路由

如果您将增强型 VPC 路由用于 Amazon Redshift 目标，则您的 Amazon Redshift 集群和数据存储库之间的所有 COPY 流量将流经您的 VPC。由于增强型 VPC 路由会影响 Amazon Redshift 访问其他资源的方式，因此如果您未正确配置 VPC，COPY 命令可能会失败。

AWS DMS 会使用 COPY 命令将 S3 中的数据移动到 Amazon Redshift 集群，因此可能会受此行为的影响。

AWS DMS 将数据加载到 Amazon Redshift 目标时采用的步骤如下：

1. AWS DMS 将数据从源复制到复制服务器上的 .csv 文件。
2. AWS DMS 使用 AWS 开发工具包将 .csv 文件复制到您账户上的 S3 存储桶中。
3. 然后，AWS DMS 在 Amazon Redshift 中使用 COPY 命令将数据从 S3 中的 CSV 文件复制到 Amazon Redshift 中相应的表。

如果未启用增强型 VPC 路由，则 Amazon Redshift 会通过互联网路由流量，包括至 AWS 网络中的其他服务的流量。如果未启用该功能，您不必配置网络路径。如果启用此功能，您必须专门创建您集群的 VPC 与数据源之间的网络路径。有关所需配置的更多信息，请参阅 Amazon Redshift 文档中的[增强型 VPC 路由](#)。



## 创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密

您可以先加密推送到 Amazon S3 的目标数据，然后再将其复制到 Amazon Redshift。为此，您可以创建并使用自定义 AWS KMS 密钥。您可以在创建 Amazon Redshift 目标端点时使用所创建的密钥通过下列机制之一加密目标数据：

- 在使用 AWS CLI 运行 `create-endpoint` 命令时使用以下选项。

```
--redshift-settings '{"EncryptionMode": "SSE_KMS", "ServerSideEncryptionKmsKeyId":  
  "your-kms-key-ARN"}'
```

此处，*your-kms-key-ARN* 是您的 KMS 密钥的 Amazon 资源名称 (ARN)。有关更多信息，请参阅[使用数据加密密钥](#)，并将 [Amazon S3 存储桶](#) 作为中间存储。

- 将额外连接属性 `encryptionMode` 设置为值 `SSE_KMS`，并将额外连接属性 `serverSideEncryptionKmsKeyId` 设置为您的 KMS 密钥的 ARN。有关更多信息，请参阅[使用 Amazon Redshift 作为 AWS DMS 的目标时的端点设置](#)。

要使用 KMS 密钥加密 Amazon Redshift 目标数据，您需要一个有权访问 Amazon Redshift 数据的 AWS Identity and Access Management (IAM) 角色。随后，在附加到您创建的加密密钥的政策（密钥策略）中访问此 IAM 角色。您可以通过创建以下项在 IAM 控制台中执行此操作：

- 一个带 AWS 托管策略的 IAM 角色。
- 一个带引用此角色的密钥政策的 KMS 密钥。

以下过程介绍了如何执行此操作。


### 创建带所需 AWS 托管策略的 IAM 角色

1. 通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择角色。Roles (角色) 页面随即打开。
3. 选择创建角色。Create role (创建角色) 页面随即打开。
4. 选择 AWS 服务作为可信实体后，选择 DMS 作为要使用该角色的服务。
5. 选择下一步: 权限。此时将显示 Attach permissions policies (附加权限策略) 页面。
6. 查找并选择 AmazonDMSRedshiftS3Role 策略。
7. 选择下一步：标签。此时将显示 Add tags (添加标签) 页面。您可以在此处添加所需的任何标签。
8. 选择下一步: 审核并检查您的结果。

9. 如果这些设置是您所需的，请输入角色的名称（例如，DMS-Redshift-endpoint-access-role）及任何其他描述，然后选择 Create role（创建角色）。Roles（角色）页面随即打开，并显示一条指示您的角色已创建的消息。


现在，您已创建新角色来访问 Amazon Redshift 资源以使用指定名称（例如，DMS-Redshift-endpoint-access-role）进行加密。

创建带引用您的 IAM 角色的密钥政策的 AWS KMS 加密密钥

 Note

有关如何将 AWS DMS 与 AWS KMS 加密密钥结合使用的更多信息，请参阅 [设置加密密钥并指定 AWS KMS 权限](#)。

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS Key Management Service (AWS KMS) 控制台：<https://console.aws.amazon.com/kms>。
2. 要更改 AWS 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择 Create key。此时会打开 Configure key（配置密钥）页面。
5. 对于 Key type（密钥类型），选择 Symmetric（对称）。

 Note

创建此密钥时，您只能创建对称密钥，因为所有 AWS 服务（如 Amazon Redshift）仅使用对称加密密钥。

6. 选择 Advanced Options。对于 Key material origin（密钥材料来源），请确保选择了 KMS，然后选择 Next（下一步）。此时会打开 Add labels（添加标签）页面。
7. 对于 Create alias and description（创建别名和描述），输入密钥的别名（例如 DMS-Redshift-endpoint-encryption-key）和任何其他描述。
8. 对于 Tags（标签），添加要用于帮助标识密钥并跟踪其使用情况的任何标签，然后选择 Next（下一步）。此时会打开 Define key administrative permissions（定义密钥管理权限）页面，其中显示了可供选择的用户和角色的列表。
9. 添加要管理密钥的用户和角色。确保这些用户和角色具有管理密钥所需的权限。

10. 对于 Key deletion (密钥删除), 选择密钥管理员是否能删除密钥, 然后选择 Next (下一步)。此时会打开 Define Key Usage Permissions (定义密钥使用权限) 页面, 其中显示了可供选择的用户和角色的附加列表。
11. 对于此账户, 选择要对 Amazon Redshift 目标执行加密操作的可用用户。此外, 还应选择您之前在角色中创建的角色以启用访问权限来加密 Amazon Redshift 目标对象, 例如 DMS-Redshift-endpoint-access-role。
12. 如果要添加未列出的其他账户以获得相同访问权限, 请为其他 AWS 账户选择添加其他 AWS 账户, 然后选择下一步。此时会打开 Review and edit key policy (查看和编辑密钥策略) 页面, 其中显示了密钥策略的 JSON, 您可以通过键入到现有 JSON 来查看和编辑该策略。在此处, 您可以看到密钥策略引用您在上一步中选择的角色和用户 (例如, Admin 和 User1) 的位置。您还可以查看不同的委托人 (用户和角色) 所允许的不同的密钥操作, 如以下示例所示。

```
{
  "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. 选择完成。加密密钥页面随即打开，并显示一条指示您的 AWS KMS key 已创建的消息。

现在，您已创建一个具有指定别名（例如，DMS-Redshift-endpoint-encryption-key）的新 KMS 密钥。此密钥允许 AWS DMS 加密 Amazon Redshift 目标数据。

### 使用 Amazon Redshift 作为 AWS DMS 的目标时的端点设置

您可以使用端点设置来配置 Amazon Redshift 目标，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者在 [AWS CLI](#) 中使用符合 `--redshift-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令。

下表显示了将 Amazon Redshift 作为目标时您可以使用的端点设置。

名称	描述
MaxFileSize	<p>指定用于将数据传输到 Amazon Redshift 的任何 CSV 文件的最大大小 (以 KB 为单位)。</p> <p>默认值：32768 KB (32 MB)</p> <p>有效值：1-1048576</p> <p>例如：<code>--redshift-settings '{"MaxFileSize": 512}'</code></p>
FileTransferUploadStreams	<p>指定用于上传单个文件的线程的数目。</p> <p>默认值：10</p> <p>有效值：1-64</p>

名称	描述
	例如： <code>--redshift-settings '{"FileTransferUploadStreams": 20}'</code>
Acceptanydate	<p>指定是否接受任何日期格式，包括无效和日期格式，如 0000-00-00。布尔值。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true   false</code></p> <p>例如：<code>--redshift-settings '{"Acceptanydate": true}'</code></p>
Dateformat	<p>指定日期格式。默认情况下，这是一个字符串输入且为空。默认格式为 YYYY-MM-DD，但您可以将其更改为 DD-MM-YYYY。如果您的日期或时间值使用不同的格式，请结合使用 <code>auto</code> 参数与 <code>Dateformat</code> 参数。<code>auto</code> 参数将识别使用 <code>Dateformat</code> 字符串时不受支持的一些格式。<code>auto</code> 关键字区分大小写。</p> <p>默认值：<code>empty</code></p> <p>有效值：<code>"dateformat_string"</code> 或 <code>auto</code></p> <p>例如：<code>--redshift-settings '{"Dateformat": "auto"}'</code></p>

名称	描述
Timeformat	<p>指定时间格式。默认情况下，这是一个字符串输入且为空。auto 参数将识别使用 Timeformat 字符串时不受支持的一些格式。如果您的日期和时间值使用各不相同的格式，请结合使用 auto 参数与 Timeformat 参数。</p> <p>默认值：10</p> <p>有效值：<i>"Timeformat_string"</i>   "auto"   "epochsecs"   "epochmillisecs"</p> <p>例如：<code>--redshift-settings '{"Timeformat": "auto"}'</code></p>
Emptyasnull	<p>指定 AWS DMS 是否应迁移作为 null 的空 CHAR 和 VARCHAR 字段。值 true 可将空的 CHAR 和 VARCHAR 字段设置为 null。</p> <p>默认值：false</p> <p>有效值：true   false</p> <p>例如：<code>--redshift-settings '{"Emptyasnull": true}'</code></p>
TruncateColumns	<p>将列中的数据截断为合适的字符数以符合列规范。仅适用于具有 VARCHAR 或 CHAR 数据类型的列以及大小为 4 MB 或以下的行。</p> <p>默认值：false</p> <p>有效值：true   false</p> <p>例如：<code>--redshift-settings '{"TruncateColumns": true}'</code></p>

名称	描述
RemoveQuotes	<p>删除传入数据中的字符串周围的引号。将保留引号中的所有字符（包括分隔符）。有关删除对 Amazon Redshift 目标的引用的更多信息，请参阅 <a href="#">Amazon Redshift 数据库开发人员指南</a>。</p> <p>默认值：false</p> <p>有效值：true   false</p> <p>例如：<code>--redshift-settings '{"Remove Quotes": true}'</code></p>
TrimBlanks	<p>删除 VARCHAR 字符串的尾部空格字符。此参数仅适用于具有 VARCHAR 数据类型的列。</p> <p>默认值：false</p> <p>有效值：true   false</p> <p>例如：<code>--redshift-settings '{"TrimBlanks": true}'</code></p>



名称	描述
EncryptionMode	<p>指定在将数据复制到 Amazon Redshift 之前要用于将其推送到 S3 的服务器端加密模式。有效值为 SSE_S3 ( S3 服务器端加密 ) 或 SSE_KMS ( KMS 密钥加密 )。如果您选择 SSE_KMS , 请将 ServerSideEncryptionKmsKeyId 参数设置为要用于加密的 KMS 密钥的 Amazon 资源名称 (ARN)。</p> <div data-bbox="688 541 1507 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>您也可以使用 CLI <code>modify-endpoint</code> 命令将现有端点的 EncryptionMode 设置值从 SSE_KMS 更改为 SSE_S3。但您不能将 EncryptionMode 值从 SSE_S3 更改为 SSE_KMS。</p> </div> <p>默认值 : SSE_S3</p> <p>有效值 : SSE_S3 或 SSE_KMS</p> <p>例如 : <code>--redshift-settings '{"EncryptionMode": "SSE_S3"}'</code></p>
ServerSideEncryptionKmsKeyId	<p>如果您将 EncryptionMode 设置为 SSE_KMS , 请将此参数设置为 KMS 密钥的 ARN。您可以通过在为您的账户创建的 AWS KMS 密钥列表中选择密钥别名来查找此 ARN。在创建密钥时 , 您必须将特定策略和角色与密钥关联。有关更多信息 , 请参阅<a href="#">创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密</a>。</p> <p>例如 : <code>--redshift-settings '{"ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'</code></p>

名称	描述
EnableParallelBatchInMemoryCSVFiles	EnableParallelBatchInMemoryCSVFiles 设置通过让 DMS 写入磁盘而不是内存来提高大型多线程完全加载任务的性能。默认值为 false。
CompressCsvFiles	使用此属性可以压缩迁移期间发送到 Amazon Redshift 目标的数据。默认值为 true，默认情况下压缩处于启用状态。

## 使用数据加密密钥，并将 Amazon S3 存储桶作为中间存储

您可以使用 Amazon Redshift 目标端点设置来配置以下内容：

- 自定义 AWS KMS 数据加密密钥。随后，可以使用此密钥来加密推送到 Amazon S3 的数据，然后再将其复制到 Amazon Redshift。
- 自定义 S3 存储桶作为迁移到 Amazon Redshift 的数据的中间存储。
- 将布尔值映射为来自 PostgreSQL 源的布尔值。默认情况下，BOOLEAN 类型迁移为 varchar(1)。您可以指定 MapBooleanAsBoolean 让 Redshift 目标将布尔类型迁移为布尔值，如以下示例所示。

```
--redshift-settings '{"MapBooleanAsBoolean": true}'
```

请注意，您必须在源端点和目标端点上都设置此设置才能使其生效。

## 用于数据加密的 KMS 密钥设置

以下示例说明如何配置自定义 KMS 密钥以加密推送到 S3 的数据。要开始，您可以使用 AWS CLI 进行以下 create-endpoint 调用。

```
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"}'
```

此处，`--redshift-settings` 选项所指定的 JSON 对象定义了两个参数。一个是带值 `SSE_KMS` 的 `EncryptionMode` 参数。另一个是带值 `arn:aws:kms:us-east-1:111122223333:key/24c3c5a1-f34a-4519-a85b-2debbef226d1` 的 `ServerSideEncryptionKmsKeyId` 参数。此值是自定义 KMS 密钥的 Amazon 资源名称 (ARN)。

默认情况下，使用 S3 服务器端加密进行 S3 数据加密。对于上一示例中的 Amazon Redshift 目标，这也等同于指定其端点设置，如以下示例所示。

```
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"}
```

有关使用 S3 服务器端加密的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的[使用服务器端加密保护数据](#)。

#### Note

您也可以使用 CLI `modify-endpoint` 命令将现有端点的 `EncryptionMode` 参数值从 `SSE_KMS` 更改为 `SSE_S3`。但您不能将 `EncryptionMode` 值从 `SSE_S3` 更改为 `SSE_KMS`。

## Amazon S3 存储桶设置

当您将数据迁移到 Amazon Redshift 目标端点时，在将迁移的数据复制到 Amazon Redshift 之前使用默认 Amazon S3 存储桶作为中间任务存储。例如，给出的有关使用 AWS KMS 数据加密密钥创建 Amazon Redshift 目标端点的示例使用此默认 S3 存储桶（请参阅[用于数据加密的 KMS 密钥设置](#)）。

您可以通过在 AWS CLI `create-endpoint` 命令的 `--redshift-settings` 选项值中包含以下参数来为此中间存储指定自定义 S3 存储桶：

- `BucketName` – 代表您指定的 S3 存储桶存储名称的字符串。如果您的服务访问角色基于 `AmazonDMSRedshiftS3Role` 策略，则此值的前缀必须为 `dms-`，例如 `dms-my-bucket-name`。
- `BucketFolder` – ( 可选 ) 代表指定 S3 存储桶中存储文件夹名称的字符串。
- `ServiceAccessRoleArn` – 允许对 S3 存储桶进行管理访问的 IAM 角色的 ARN。通常，您将根据 `AmazonDMSRedshiftS3Role` 策略创建此角色。有关示例，请参阅 [创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密](#) 中关于创建具有所需 AWS 托管策略的 IAM 角色的过程。

**Note**

如果您使用 `create-endpoint` 命令的 `--service-access-role-arn` 选项指定其他 IAM 角色的 ARN，则此 IAM 角色选项优先。

以下示例显示了如何使用这些参数通过使用 AWS CLI 在以下 `create-endpoint` 调用中指定自定义 Amazon S3 存储桶。

```
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"}'
```

## Amazon Redshift 的多线程任务设置

您可以使用多线程任务设置提高 Amazon Redshift 目标端点的完全加载和更改数据捕获 (CDC) 任务的性能。为此，您可以指定并发线程的数量和要存储在缓冲区中的记录数。

### Amazon Redshift 的多线程完全加载任务设置

要提高完全加载性能，您可以使用以下 `ParallelLoad*` 任务设置：

- `ParallelLoadThreads` – 指定 DMS 在完全加载期间用于将数据记录推送到 Amazon Redshift 目标端点的并发线程数。默认值为零 (0)，最大值为 32。有关更多信息，请参阅[完全加载任务设置](#)。

使用 `ParallelLoadThreads` 任务设置时，可以使用设置为 `false` 的 `enableParallelBatchInMemoryCSVFiles` 属性。该属性通过让 DMS 写入磁盘而不是内存来提高大型多线程完全加载任务的性能。默认值为 `true`。

- `ParallelLoadBufferSize` – 指定使用以 Redshift 为目标且使用并行加载线程时的最大数据记录请求量。默认值是 100，最大值是 1,000。建议您在 `ParallelLoadThreads > 1` (大于 1) 时使用此选项。

**Note**

在 AWS DMS 版本 3.4.5 及更高版本中，支持在完全加载到 Amazon Redshift 目标端点期间使用 `ParallelLoad*` 任务设置。

不支持在更改数据捕获 (CDC) 期间或启用并行加载的完全加载迁移任务期间使用 `ReplaceInvalidChars` Redshift 端点设置。如果未启用并行加载，则支持完全加载迁移。有关更多信息，请参阅《AWS Database Migration Service API 参考》中的 [RedshiftSettings](#)。

## Amazon Redshift 的 CDC 任务设置

要提高 CDC 性能，您可以使用以下 `ParallelApply*` 任务设置：

- `ParallelApplyThreads` – 指定 AWS DMS 在 CDC 加载期间用于将数据记录推送到 Amazon Redshift 目标端点的并发线程数。默认值为零 (0)，最大值为 32。建议的最小值等于集群中的切片数。
- `ParallelApplyBufferSize` – 指定使用以 Redshift 为目标的并行应用线程时的最大数据记录请求量。默认值是 100，最大值是 1,000。建议您在 `ParallelApplyThreads > 1` (大于 1) 时使用此选项。

为了在将 Redshift 作为目标时获得最大收益，我们建议 `ParallelApplyBufferSize` 的值至少为 `ParallelApplyThreads` 数量的两倍 (乘以二或更多)。

**Note**

在 AWS DMS 版本 3.4.3 及更高版本中，支持在 CDC 到 Amazon Redshift 目标端点期间使用 `ParallelApply*` 任务设置。

应用的并行度级别取决于总批量大小与用于传输数据的最大文件大小之间的相关性。将多线程 CDC 任务设置与 Redshift 目标一起使用时，当批量大小相对于最大文件大小而言较大时，可以获得好处。例如，您可以使用以下端点和任务设置组合，进行调整以获得最佳性能。

```
// Redshift endpoint setting  
  
MaxFileSize=250000;
```

```
// Task settings

BatchApplyEnabled=true;
BatchSplitSize =8000;
BatchApplyTimeoutMax =1800;
BatchApplyTimeoutMin =1800;
ParallelApplyThreads=32;
ParallelApplyBufferSize=100;
```

通过使用前一个示例中的设置，事务工作负载繁重的客户会通过批处理缓冲区受益，其中包含 8000 条记录、可在 1800 秒内装满、使用 32 个并行线程且最大文件大小为 250 MB。

有关更多信息，请参阅[更改处理优化设置](#)。

### Note

在持续复制到 Redshift 集群期间运行的 DMS 查询可以与正在运行的其他应用程序查询共享相同的 WLM (工作负载管理) 队列。因此，请考虑正确配置 WLM 属性，以避免其在持续复制到 Redshift 目标期间影响性能。例如，如果正在运行其他并行 ETL 查询，则 DMS 的运行速度会变慢，就会失去性能增益。

## Amazon Redshift 的目标数据类型

AWS DMS 的 Amazon Redshift 端点支持大多数 Amazon Redshift 数据类型。下表显示了使用 AWS DMS 时支持的 Amazon Redshift 目标数据类型以及来自 AWS DMS 数据类型的默认映射。

有关 AWS DMS 数据类型的其他信息，请参阅[AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	Amazon Redshift 数据类型
BOOLEAN	BOOL
BYTES	VARCHAR (Length)
DATE	DATE
TIME	VARCHAR(20)

AWS DMS 数据类型	Amazon Redshift 数据类型
DATETIME	<p>如果小数位数 =&gt; 0 且 =&lt; 6，则视 Redshift 目标列类型而定，为以下之一：</p> <p>TIMESTAMP (s)</p> <p>TIMESTAMPTZ (s) – 如果源时间戳包含时区偏移（例如在 SQL Server 或 Oracle 中），则它会在插入/更新时转换为 UTC。如果它不包含偏移量，则时间已经以 UTC 为单位考虑。</p> <p>如果小数位数 =&gt; 7 且 &lt;= 9，请使用：</p> <p>VARCHAR(37)</p>
INT1	INT2
INT2	INT2
INT4	INT4
INT8	INT8
NUMERIC	<p>如果小数位数 =&gt; 0 且 &lt;= 37，请使用：</p> <p>NUMERIC (p,s)</p> <p>如果小数位数 =&gt; 38 且 &lt;= 127，请使用：</p> <p>VARCHAR (Length)</p>
REAL4	FLOAT4
REAL8	FLOAT8
STRING	<p>如果长度为 1–65,535，请使用 VARCHAR（以字节为单位的长度）</p> <p>如果长度为 65,536–2,147,483,647，请使用 VARCHAR (65535)</p>

AWS DMS 数据类型	Amazon Redshift 数据类型
UINT1	INT2
UINT2	INT2
UINT4	INT4
UINT8	NUMERIC (20,0)
WSTRING	如果长度为 1–65,535，请使用 NVARCHAR (以字节为单位的长度)  如果长度为 65,536–2,147,483,647，请使用 NVARCHAR (65535)
BLOB	VARCHAR (最大 LOB 大小 *2)  最大 LOB 大小不能超过 31 KB。Amazon Redshift 不支持大于 64 KB 的 VARCHAR。
NCLOB	NVARCHAR (最大 LOB 大小)  最大 LOB 大小不能超过 63 KB。Amazon Redshift 不支持大于 64 KB 的 VARCHAR。
CLOB	VARCHAR (最大 LOB 大小)  最大 LOB 大小不能超过 63 KB。Amazon Redshift 不支持大于 64 KB 的 VARCHAR。

## 使用 AWS DMS 并将 Amazon Redshift Serverless 作为目标

AWS DMS 支持使用 Amazon Redshift Serverless 作为目标端点。有关使用 Amazon Redshift Serverless 的信息，请参阅 [Amazon Redshift 管理指南](#) 中的 [Amazon Redshift Serverless](#)。

本主题介绍如何将 Amazon Redshift Serverless 端点与 AWS DMS 配合使用。



**Note**

创建 Amazon Redshift Serverless 端点时，对于 [RedshiftSettings](#) 端点配置的 `DatabaseName` 字段，使用 Amazon Redshift 数据仓库的名称或工作组端点的名称。对于 `ServerName` 字段，使用无服务器集群的工作组页面显示的 `Endpoint` 值（例如，`default-workgroup.093291321484.us-east-1.redshift-serverless.amazonaws.com`）。有关创建端点的信息，请参阅[创建源和目标终端节点](#)。有关工作组端点的信息，请参阅[连接到 Amazon Redshift Serverless](#)。

## 以 Amazon Redshift Serverless 作为目标时的信任策略

使用 Amazon Redshift Serverless 作为目标端点时，您必须在信任策略中添加以下突出显示的部分。该信任策略已附加到 `dms-access-for-endpoint` 角色。

```
{
  "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
}
}
```

有关在 AWS DMS 中使用信任策略的更多信息，请参阅[创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

将 Amazon Redshift Serverless 作为目标时的限制

使用 Redshift Serverless 作为目标存在以下限制：

- AWS DMS 仅在支持 Amazon Redshift Serverless 的地区支持将 Amazon Redshift Serverless 作为端点。有关哪些区域支持 Amazon Redshift Serverless 的信息，请参阅 [AWS 一般参考](#) 中 [Amazon Redshift 端点和配额](#) 主题中的 Redshift Serverless API。
- 使用增强型 VPC 路由时，请确保在与 Redshift Serverless 或 Redshift 预置集群相同的 VPC 中创建 Amazon S3 端点。有关更多信息，请参阅[将 Amazon Redshift 作为 AWS Database Migration Service 目标时使用增强型 VPC 路由](#)。
- AWS DMS Serverless 不支持 Amazon Redshift Serverless 作为端点。

## 将 SAP ASE 数据库作为 AWS Database Migration Service 的目标

您可以使用 AWS DMS，从任意支持的数据源将数据迁移到 SAP Adaptive Server Enterprise (ASE) (以前称为 Sybase) 数据库。

有关 AWS DMS 支持作为目标的 SAP ASE 版本的信息，请参阅[的目标 AWS DMS](#)。

### 将 SAP ASE 数据库作为 AWS Database Migration Service 目标的先决条件

在开始将 SAP ASE 数据库作为 AWS DMS 的目标之前，请确保满足以下先决条件：

- 向 AWS DMS 用户提供 SAP ASE 账户访问权限。此用户必须具有 SAP ASE 数据库中的读/写权限。
- 在某些情况下，您可能会复制到安装在 Microsoft Windows 上、使用非拉丁字符（例如，中文）配置的 Amazon EC2 实例上的 SAP ASE 版本 15.7。在此类情况下，AWS DMS 需要在目标 SAP ASE 计算机上安装 SAP ASE 15.7 SP121。

## 将 SAP ASE 数据库作为 AWS DMS 的目标的限制

将 SAP ASE 数据库作为 AWS DMS 的目标时，存在以下限制：

- AWS DMS 不支持包含以下数据类型的字段的表：具有这些数据类型的复制列显示为 null。
  - 用户定义的类型 (UDT)

## 使用 SAP ASE 作为 AWS DMS 的目标时的端点设置

您可以使用端点设置来配置 SAP ASE 目标数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者在 [AWS CLI](#) 中使用符合 `--sybase-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令。

下表显示了将 SAP ASE 作为目标时您可以使用的端点设置。

名称	描述
Driver	<p>如果要为 ASE 15.7 及更高版本使用 TLS，请设置此属性。</p> <p>默认值：Adaptive Server Enterprise</p> <p>示例：driver=Adaptive Server Enterprise 16.03.06;</p> <p>有效值：Adaptive Server Enterprise 16.03.06</p>
AdditionalConnectionProperties	要指定的任何其他 ODBC 连接参数。

## SAP ASE 的目标数据类型

下表显示了使用 AWS DMS 时支持的 SAP ASE 数据库目标数据类型以及来自 AWS DMS 数据类型的默认映射。

有关 AWS DMS 数据类型的其他信息，请参阅 [AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	SAP ASE 数据类型
BOOLEAN	BIT
BYTES	VARBINARY (Length)
DATE	DATE
TIME	TIME
TIMESTAMP	如果小数位数 => 0 且 <= 6 , 请使用 BIGDATETIME  如果小数位数 => 7 且 <= 9 , 请使用 VARCHAR (37)”
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	DOUBLE PRECISION
STRING	VARCHAR (Length)
UINT1	TINYINT
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	VARCHAR (Length)

AWS DMS 数据类型	SAP ASE 数据类型
BLOB	IMAGE
CLOB	UNITEXT
NCLOB	TEXT

## 将 Amazon S3 作为 AWS Database Migration Service 的目标

您可以使用 AWS DMS 将数据从支持的任意数据库源迁移到 Amazon S3。在使用 Amazon S3 作为 AWS DMS 任务中的目标时，默认情况下，完全加载和更改数据捕获 (CDC) 数据将以逗号分隔值 (.csv) 格式写入。对于更紧凑的存储和更快速的查询选项，您还可以选择将数据写入 Apache Parquet (.parquet) 格式。

AWS DMS 使用增量十六进制计数器命名完全加载期间创建的文件，例如，将 .csv 文件命名为 LOAD00001.csv、LOAD00002...、LOAD00009、LOAD0000A 等。AWS DMS 使用时间戳命名 CDC 文件，例如 20141029-1134010000.csv。对于包含记录的各个源表，AWS DMS 在指定的目标文件夹下创建文件夹（如果源表不为空）。AWS DMS 将所有完全加载和 CDC 文件写入到指定的 Amazon S3 存储桶。您可以使用 [MaxFileSize](#) 端点设置来控制 AWS DMS 所创建文件的大小。

参数 `bucketFolder` 包含 .csv 或 .parquet 文件在上传到 S3 存储桶之前的存储位置。利用 .csv 文件，表数据使用以下格式存储在 S3 存储桶中，与完全加载文件一起显示。

```
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
...
```

您可以使用额外连接属性指定列分隔符、行分隔符以及其他参数。有关额外连接属性的更多信息，请参阅此部分结尾的[使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。

您可以使用 `ExpectedBucketOwner` Amazon S3 端点设置指定存储桶所有者并防止狙击，如下所示。然后，当您发出测试连接或执行迁移的请求时，S3 会根据指定参数检查存储桶所有者的账户 ID。

```
--s3-settings='{ "ExpectedBucketOwner": "AWS_Account_ID" }'
```

在使用 AWS DMS 复制使用 CDC 任务进行的数据更改时，.csv 或 .parquet 输出文件的第一列指示行数据的更改方式，如以下 .csv 文件所示。

```
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
```

对于该示例，假设源数据库中有一个 EMPLOYEE 表。AWS DMS 将数据写入 .csv 或 .parquet 文件中来响应以下事件：

- 14 年 6 月 4 日，纽约办事处聘用了一名新员工 (Bob Smith，员工 ID 101)。在 .csv 或 .parquet 文件中，第一列中的 I 指示在源数据库的 EMPLOYEE 表中插入 (INSERT) 了一个新行。
- 15 年 10 月 8 日，Bob 调到了洛杉矶办事处。在 .csv 或 .parquet 文件中，U 指示更新 (UPDATE) 了 EMPLOYEE 表中的相应行以反映 Bob 的新办公地点。该行的其余部分反映更新 (UPDATE) 后如何显示该 EMPLOYEE 表行。
- 17 年 3 月 13 日，Bob 又调到了达拉斯办事处。在 .csv 或 .parquet 文件中，U 指示再次更新 (UPDATE) 了该行。该行的其余部分反映更新 (UPDATE) 后如何显示该 EMPLOYEE 表行。
- 在达拉斯工作一段时间后，Bob 离开了公司。在 .csv 或 .parquet 文件中，D 指示在源表中删除 DELETE 了该行。该行的其余部分反映在删除之前如何显示该 EMPLOYEE 表行。

请注意，对于 CDC，AWS DMS 会默认存储每个数据库表的行更改，而不考虑事务顺序。如果要根据事务顺序将行更改存储在 CDC 文件中，则需要使用 S3 端点设置来指定此设置以及您希望在 S3 目标上存储 CDC 事务文件的文件夹路径。有关更多信息，请参阅[捕获数据更改 \(CDC\) 时包括 S3 目标上的事务顺序](#)。

要控制在执行数据复制任务期间向 Amazon S3 目标写入的频率，您可以配置 `cdcMaxBatchInterval` 和 `cdcMinFileSize` 额外连接属性。这可以提高分析数据的性能，而无需任何额外的开销操作。有关更多信息，请参阅[使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。

## 主题

- [将 Amazon S3 作为目标的先决条件](#)

- [将 Amazon S3 作为目标的限制](#)
- [安全性](#)
- [使用 Apache Parquet 存储 Amazon S3 对象](#)
- [Amazon S3 对象标记](#)
- [创建 AWS KMS 密钥以加密 Amazon S3 目标对象](#)
- [使用基于日期的文件夹分区](#)
- [使用 Amazon S3 作为 AWS DMS 的目标时并行加载已分区的源](#)
- [使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)
- [将 AWS Glue Data Catalog 与 Amazon S3 目标一起用于 AWS DMS](#)
- [在 Amazon S3 目标上使用数据加密、parquet 文件和 CDC](#)
- [指示迁移的 S3 数据中的源数据库操作](#)
- [S3 Parquet 的目标数据类型](#)

## 将 Amazon S3 作为目标的先决条件

在使用 Amazon S3 作为目标之前，请检查是否满足以下条件：

- 作为目标的 S3 存储桶与用于迁移数据的 DMS 复制实例位于相同的 AWS 区域中。
- 您在迁移中使用的 AWS 账户拥有一个 IAM 角色，此角色具有对作为目标的 S3 存储桶的写入和删除访问权限。
- 此角色已标记访问，以便您可以标记写入目标存储桶的任何 S3 对象。
- IAM 角色已将 DMS (dms.amazonaws.com) 添加为可信实体。

要设置此账户访问权限，请确保向用于创建迁移任务的用户账户分配的角色具有以下一系列权限。

```
{
  "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"
    ]
  }
]
```

有关将 S3 作为目标使用验证的先决条件，请参阅[S3 目标验证先决条件](#)。

## 将 Amazon S3 作为目标的限制

将 Amazon S3 作为目标时存在以下限制：

- 不要为 S3 启用版本控制。如果您需要 S3 版本控制，请使用生命周期策略主动删除旧版本。否则，您可能会遇到由于 S3 list-object 调用超时而导致端点测试连接失败的情况。要为 S3 存储桶创建生命周期策略，请参阅[管理存储生命周期](#)。要删除 S3 对象的某个版本，请参阅[从启用版本控制的存储桶中删除对象版本](#)。
- 3.4.7 及更高版本支持启用了 VPC ( 网关 VPC ) 的 S3 存储桶。
- 支持以下数据定义语言 (DDL) 命令以更改数据捕获 (CDC)：截断表、删除表、创建表、重命名表、添加列、删除列、重命名列、重命名列和更改列数据类型。请注意，在源数据库中添加、删除或重命名列时，不会在目标 S3 存储桶中记录 ALTER 语句，AWS DMS 也不会更改先前创建的记录以匹配新结构。更改后，AWS DMS 使用新的表结构创建所有新记录。

### Note

截断 DDL 操作会删除 S3 存储桶中的所有文件和相应的表文件夹。您可以使用任务设置来禁用该行为，并配置 DMS 在更改数据捕获 (CDC) 期间处理 DDL 行为的方式。有关更多信息，请参阅[更改处理 DDL 处理的任务设置](#)。

- 不支持完整 LOB 模式。
- 不支持在完全加载期间对源表结构进行更改。支持在完全加载期间更改数据。



- 将多个任务从同一源表复制到同一目标 S3 终端节点存储桶将导致这些任务写入到同一文件。如果您的数据源来自同一个表，我们建议您指定不同的目标端点（存储桶）。
- S3 端点不支持 BatchApply。为 S3 目标使用“批应用”（例如，BatchApplyEnabled 目标元数据任务设置）可能会导致数据丢失。
- 您可以将 DatePartitionEnabled 或 addColumnName 与 PreserveTransactions 或 CdcPath 一起使用。
- AWS DMS 不支持使用转换规则将多个源表重命名到同一个目标文件夹中。
- 如果在完全加载阶段对源表进行大量写入，则 DMS 可能会将重复的记录写入 S3 存储桶或缓存的更改。
- 如果为任务配置的 TargetTablePrepMode 为 DO\_NOTHING，并且任务在完全加载阶段突然停止并恢复，则 DMS 可能会将重复的记录写入 S3 存储桶。
- 如果为目标端点配置的 PreserveTransactions 设置为 true，则重新加载表不会清除之前生成的 CDC 文件。有关更多信息，请参阅[捕获数据更改 \(CDC\) 时包括 S3 目标上的事务顺序](#)。

有关将 S3 作为目标使用验证的限制，请参阅[使用 S3 目标验证的限制](#)。

## 安全性

要使用 Amazon S3 作为目标，在迁移中使用的账户必须具有对作为目标的 Amazon S3 存储桶的写入和删除访问权限。对于具有访问 Amazon S3 所需权限的 IAM 角色，指定该角色的 Amazon 资源名称 (ARN)。

对于 Amazon S3，AWS DMS 支持一组预定义的授权（称为标准访问控制列表 (ACL)）。每个标准 ACL 具有一组被授予者和权限，可用于为 Amazon S3 存储桶设置权限。您可以将 cannedAclForObject 用于 S3 目标终端节点的连接字符串属性以指定标准 ACL。有关使用额外连接属性 cannedAclForObject 的更多信息，请参阅[使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。有关 Amazon S3 标准 ACL 的更多信息，请参阅[标准 ACL](#)。

用于迁移的 IAM 角色必须能够执行 s3:PutObjectAcl API 操作。

## 使用 Apache Parquet 存储 Amazon S3 对象

逗号分隔值 (.csv) 格式是 Amazon S3 目标对象的默认存储格式。对于更紧凑的存储和更快速的查询，您可以改用 Apache Parquet (.parquet) 作为存储格式。

Apache Parquet 是一种最初专为 Hadoop 设计的开源文件存储格式。有关 Apache Parquet 的更多信息，请参阅 <https://parquet.apache.org/>。

要将 `.parquet` 设置为您的已迁移 S3 目标对象的存储格式，您可以使用以下机制：

- 您在使用 AWS CLI 或 AWS DMS 的 API 创建终端节点时提供作为 JSON 对象的参数的终端节点设置。有关更多信息，请参阅[在 Amazon S3 目标上使用数据加密、parquet 文件和 CDC](#)。
- 您在创建终端节点时提供作为分号分隔列表的额外连接属性。有关更多信息，请参阅[使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。

## Amazon S3 对象标记



您可以标记复制实例所创建的 Amazon S3 对象，方式是指定适当的 JSON 对象作为任务-表映射规则的一部分。有关 S3 对象标记的要求和选项（包括有效标签名称）的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的[对象标记](#)。有关使用 JSON 的表映射的更多信息，请参阅[使用 JSON 指定表选择和转换规则](#)。

您可以标记为指定表创建的 S3 对象，方式是使用 `selection` 规则类型的一个或多个 JSON 对象。然后，您可以使用 `add-tag` 操作在此 `selection` 对象后添加 `post-processing` 规则类型的一个或多个 JSON 对象。这些后处理规则标识要标记的 S3 对象并指定要添加到这些 S3 对象的标签的名称和价值。

您可以在下表中找到要在 `post-processing` 规则类型的 JSON 对象中指定的参数。

参数	可能的值	描述
<code>rule-type</code>	<code>post-processing</code>	一个将后处理操作应用于生成的目标对象的值。您可以指定一个或多个后处理规则以标记所选 S3 对象。
<code>rule-id</code>	一个数字值。	一个用于标识规则的唯一数字值。
<code>rule-name</code>	一个字母数字值。	一个用于标识规则的唯一名称。
<code>rule-action</code>	<code>add-tag</code>	要应用于 S3 对象的后处理操作。您可以通过对 <code>add-tag</code> 操作使用单个 JSON 后处理对象来添加一个或多个标签。
<code>object-locator</code>	<code>schema-name</code> – 表架构的名称。 <code>table-name</code> – 表的名称。	规则应用于的每个架构和表的名称。您可以使用“%”百分比符号作为每个 <code>object-locator</code> 参数的值的全部

参数	可能的值	描述
		<p>或部分的通配符。因此，您可以匹配这些项目：</p> <ul style="list-style-type: none"><li>• 单个架构中的单个表</li><li>• 部分或全部架构中的单个表</li><li>• 单个架构中的部分或全部表</li><li>• 部分或全部架构中的部分或全部表</li></ul>

参数	可能的值	描述
tag-set	<p>key – 单个标签的任何有效名称。</p> <p>value – 此标签的任何有效的 JSON 值。</p>	<p>要在匹配指定的 object-locator 的每个已创建 S3 对象上设置的一个或多个标签的名称和值。您可以在单个 tag-set 参数对象中最多指定 10 个键值对。有关 S3 对象标记的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的<a href="#">对象标记</a>。</p> <p>您还可以使用 <code>\${dyn-value}</code> 为 key 和 value 参数的值的全部或部分指定一个动态值。此处，<code>\${dyn-value}</code> 可以是 <code>\${schema-name}</code> 或 <code>\${table-name}</code>。因此，您可以插入当前选定架构或表的名称作为参数值的完整内容或任何部分。</p> <div data-bbox="1003 1087 1122 1125"><p> Note</p></div> <div data-bbox="1081 1203 1273 1241"><p> Important</p></div> <div data-bbox="1130 1262 1430 1629"><p>如果为 key 参数插入一个动态值，则可以使用 S3 对象的重复名称生成标签，具体取决于其使用方式。在这种情况下，仅将一个重复标签设置添加到对象。</p></div>

在指定多个 post-processing 规则类型来标记一系列 S3 对象时，将仅使用来自一个后处理规则的一个 tag-set 对象来标记每个 S3 对象。用于标记给定 S3 对象的特定标记集是来自后处理规则的标记集，其关联对象定位器与该 S3 对象最匹配。

例如，假设两个后处理规则标识同一 S3 对象。另外，假设一个规则中的对象定位器使用通配符，而另一个规则中的对象定位器使用精确匹配来标识 S3 对象（不带通配符）。在这种情况下，与后处理规则关联的带精确匹配的标签集将用于标记 S3 对象。如果多个后处理规则同样匹配某个给定 S3 对象，则与第一个此类后处理规则关联的标签集将用于标记该对象。

**Example** 将静态标签添加到为单个表和架构创建的 S3 对象

以下选择和后处理规则将三个标签（tag\_1、tag\_2 和 tag\_3，分别带有相应的静态值 value\_1、value\_2 和 value\_3）添加到创建的 S3 对象。此 S3 对象对应于带有名为 aat2 的架构的名为 STOCK 的源中的单个表。

```
{
  "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** 将静态和动态标签添加到为多个表和架构创建的 S3 对象

以下示例具有一个选择规则和两个后处理规则，其中来自源的输入包括所有表及其所有架构。

```

{
  "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"
    }
  ]
}
]
}

```

第一个后处理规则将带有相应的动态值 ( `${schema-name}` 和 `my_prefix_${table-name}` ) 的两个标签 ( `dw-schema-name` 和 `dw-schema-table` ) 添加到目标中创建的几乎所有 S3 对象。例外情况是使用第二个后处理规则标识和标记的 S3 对象。因此，由通配符对象定位器标识的每个目标 S3 对象是使用标识其在源中对应的架构和表的标签创建的。

第二个后处理规则将 `tag_1` 和 `tag_2` ( 分别带有相应的静态值 `value_1` 和 `value_2` ) 添加到由精确匹配对象定位器标识的创建的 S3 对象。因此，该创建的 S3 对象对应于带有名为 `aat` 的架构的名为 `ITEM` 的源中的单个表。由于精确匹配，这些标签将替换第一个后处理规则中添加的此对象上的任何标签，这仅按通配符匹配 S3 对象。

#### Example 将动态标签名称和值添加到 S3 对象

以下示例具有两个选择规则和一个后处理规则。在这里，源中的输入仅包括 `retail` 或 `wholesale` 架构中的 `ITEM` 表。

```

{
  "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"
      }
    ]
  }
}

```

后处理规则的标签集将两个标签 ( `dw-schema-name` 和 `dw-schema-table` ) 添加到为目标中的 ITEM 表创建的所有 S3 对象。第一个标签具有动态值 `"${schema-name}"`，第二个标签具有静态值 `"my_prefix_ITEM"`。因此，每个目标 S3 对象是使用标识其在源中对应的架构和表的标签创建的。

此外，该标签集还添加带动态名称 ( `"${schema-name}_ITEM_tag_1"` 和 `"${schema-name}_ITEM_tag_2"` ) 的两个额外标签。这些标签具有相应的静态值 `value_1` 和 `value_2`。因此，这些标签均针对当前架构 `retail` 和 `wholesale` 进行命名。您无法在此对象中创建重复的动态标签名称，因为每个对象都为单个唯一的架构名称创建的。架构名称用于创建另一个唯一的标签名称。

## 创建 AWS KMS 密钥以加密 Amazon S3 目标对象

您可以创建并使用自定义 AWS KMS 密钥来加密 Amazon S3 目标对象。创建 KMS 密钥后，您可以在创建 S3 目标端点时，使用该密钥通过下列方法之一加密对象：

- 当您使用 AWS CLI 运行 `create-endpoint` 命令时对 S3 目标对象 ( 采用默认的 `.csv` 文件存储格式 ) 使用以下选项。

```

--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"}'

```

此处，`your-KMS-key-ARN` 是您的 KMS 密钥的 Amazon 资源名称 (ARN)。有关更多信息，请参阅 [在 Amazon S3 目标上使用数据加密、parquet 文件和 CDC](#)。

- 将额外连接属性 `encryptionMode` 设置为值 `SSE_KMS`，并将额外连接属性 `serverSideEncryptionKmsKeyId` 设置为您的 KMS 密钥的 ARN。有关更多信息，请参阅 [使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。

要使用 KMS 密钥加密 Amazon S3 目标对象，您需要一个有权访问 Amazon S3 存储桶的 IAM 角色。随后，在附加到您创建的加密密钥的政策 ( 密钥政策 ) 中访问此 IAM 角色。您可以通过创建以下项在 IAM 控制台中执行此操作：

- 一个具有访问 Amazon S3 存储桶的权限的策略。

- 一个带有此策略的 IAM 角色。
- 一个带引用此角色的密钥政策的 KMS 加密密钥。

以下过程介绍了如何执行此操作。

### 创建有权访问 Amazon S3 存储桶的 IAM policy

1. 通过 <https://console.aws.amazon.com/iam/> 打开 IAM 控制台。
2. 在导航窗格中，选择 Policies (策略)。Policies (策略) 页面随即打开。
3. 选择创建策略。此时将打开创建策略页面。
4. 选择 Service (服务)，然后选择 S3。此时会显示操作权限列表。
5. 选择全部展开以展开列表并至少选择以下权限：
  - ListBucket
  - PutObject
  - DeleteObject

选择所需的任何其他权限，然后选择 Collapse all (全部折叠) 以折叠列表。

6. 选择 Resources (资源) 以指定要访问的资源。至少选择所有资源以提供一般 Amazon S3 资源访问权限。
7. 添加所需的任何其他条件或权限，然后选择 Review policy (查看策略)。在 Review policy (查看策略) 页面上查看您的结果。
8. 如果这些设置是您所需的，请输入策略名称 (例如，DMS-S3-endpoint-access) 及任何其他描述，然后选择 Create policy (创建策略)。Policies (策略) 页面随即打开，并显示一条指示您的策略已创建的消息。
9. 在 Policies (策略) 列表中搜索并选择策略名称。此时会显示 Summary (摘要) 页面，并显示与以下内容类似的策略的 JSON。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
```

```
        "s3:ListBucket",
        "s3:DeleteObject"
    ],
    "Resource": "*"
}
]
```

现在，您已创建新策略来访问 Amazon S3 资源以使用指定名称（例如 DMS-S3-endpoint-access）进行加密。

### 创建具有此策略的 IAM 角色

1. 在 IAM 控制台上的导航窗格中，选择角色。Roles (角色) 详细信息页面随即打开。
2. 选择创建角色。Create role (创建角色) 页面随即打开。
3. 选择 AWS 服务作为可信实体后，选择 DMS 作为服务以使用 IAM 角色。
4. 选择下一步: 权限。Attach permissions policies (附加权限策略) 视图将显示在 Create role (创建角色) 页面中。
5. 查找并选择您在上一过程中创建的 IAM 角色的 IAM policy (DMS-S3-endpoint-access)。
6. 选择下一步: 标签。Add tags (添加标签) 视图将显示在 Create role (创建角色) 页面中。您可以在此处添加所需的任何标签。
7. 选择下一步: 审核。Review (查看) 视图将显示在 Create role (创建角色) 页面中。在这里，您可以验证结果。
8. 如果这些设置是您所需的，请输入角色名称（必填，例如 DMS-S3-endpoint-access-role）及任何其他描述，然后选择 Create role (创建角色)。Roles (角色) 详细信息页面随即打开，并显示一条指示您的角色已创建的消息。


现在，您已创建新角色来访问 Amazon S3 资源以使用指定名称（例如，DMS-S3-endpoint-access-role）进行加密。

### 创建具有引用您的 IAM 角色的密钥政策的 KMS 加密密钥

#### Note

有关如何将 AWS DMS 与 AWS KMS 加密密钥结合使用的更多信息，请参阅 [设置加密密钥并指定 AWS KMS 权限](#)。

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS Key Management Service (AWS KMS) 控制台：<https://console.aws.amazon.com/kms>。
2. 要更改 AWS 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择 Create key。此时会打开 Configure key (配置密钥) 页面。
5. 对于 Key type (密钥类型)，选择 Symmetric (对称)。

 Note

创建此密钥时，您只能创建对称密钥，因为所有 AWS 服务（如 Amazon S3）仅使用对称加密密钥。

6. 选择 Advanced Options。对于 Key material origin (密钥材料来源)，请确保选择了 KMS，然后选择 Next (下一步)。此时会打开 Add labels (添加标签) 页面。
7. 对于 Create alias and description (创建别名和描述)，输入密钥的别名（例如 DMS-S3-endpoint-encryption-key）和任何其他描述。
8. 对于 Tags (标签)，添加要用于帮助标识密钥并跟踪其使用情况的任何标签，然后选择 Next (下一步)。此时会打开 Define key administrative permissions (定义密钥管理权限) 页面，其中显示了可供选择的用户和角色的列表。
9. 添加要管理密钥的用户和角色。确保这些用户和角色具有管理密钥所需的权限。
10. 对于 Key deletion (密钥删除)，选择密钥管理员是否能删除密钥，然后选择 Next (下一步)。此时会打开 Define Key Usage Permissions (定义密钥使用权限) 页面，其中显示了可供选择的用户和角色的附加列表。
11. 对于此账户，选择要对 Amazon S3 目标执行加密操作的可用用户。此外，还应选择您之前在角色中创建的角色以启用访问权限来加密 Amazon S3 目标对象，例如 DMS-S3-endpoint-access-role。
12. 如果要添加未列出的其他账户以获得相同访问权限，请为其他 AWS 账户选择添加其他 AWS 账户，然后选择下一步。此时会打开 Review and edit key policy (查看和编辑密钥策略) 页面，其中显示了密钥策略的 JSON，您可以通过键入到现有 JSON 来查看和编辑该策略。在此处，您可以看到密钥策略引用您在上一步中选择的角色和用户（例如，Admin 和 User1）的位置。您还可以查看不同委托人（用户和角色）所允许的不同密钥操作，如以下示例所示。

```
{
  "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. 选择完成。加密密钥页面随即打开，并显示一条指示您的 KMS 密钥已创建的消息。

现在，您已创建一个具有指定别名（例如，DMS-S3-endpoint-encryption-key）的新 KMS 密钥。此密钥启用 AWS DMS 以加密 Amazon S3 目标对象。

## 使用基于日期的文件夹分区

当您使用 Amazon S3 作为目标端点时，AWS DMS 支持基于事务提交日期的 S3 文件夹分区。通过使用基于日期的文件夹分区，您可以将数据从单个源表写入 S3 存储桶中的时间层次结构文件夹结构中。通过在创建 S3 目标端点时对文件夹进行分区，您可以：

- 更好地管理您的 S3 对象
- 限制每个 S3 文件夹的大小
- 优化数据湖查询或其他后续操作

创建 S3 目标端点时，您可以启用基于日期的文件夹分区。在迁移现有数据和复制持续更改（完全加载 + CDC）或仅复制数据更改（仅 CDC）时，您可以启用它。使用以下目标端点设置：

- `DatePartitionEnabled` – 指定基于日期的分区。将该布尔选项设置为 `true` 时，可根据事务提交日期对 S3 存储桶文件夹进行分区。

不能将此设置与 `PreserveTransactions` 或 `CdcPath` 一起使用。

默认值为 `false`。

- `DatePartitionSequence` – 标识文件夹分区期间使用的日期格式顺序。将此 ENUM 选项设置为 `YYYYMMDD`、`YYYYMMDDHH`、`YYYYMM`、`MMYYYYDD` 或 `DDMMYYYY`。默认值为 `YYYYMMDD`。在 `DatePartitionEnabled` 设置为 `true` 时使用此设置。
- `DatePartitionDelimiter` – 指定要在文件夹分区期间使用的分隔日期分隔符。将此 ENUM 选项设置为 `SLASH`、`DASH`、`UNDERSCORE` 或 `NONE`。默认值为 `SLASH`。在 `DatePartitionEnabled` 设置为 `true` 时使用此设置。

以下示例说明如何启用基于日期的文件夹分区，并使用数据分区顺序和分隔符的默认值。它使用 `AWS CLI.create-endpoint` 命令的 `--s3-settings '{json-settings}'` 选项。

```
--s3-settings '{"DatePartitionEnabled": true,"DatePartitionSequence": "YYYYMMDD","DatePartitionDelimiter": "SLASH"}
```

## 使用 Amazon S3 作为 AWS DMS 的目标时并行加载已分区的源

您可以配置将已分区数据源并行完全加载到 Amazon S3 目标。该方法可缩短将已分区数据从支持的源数据库引擎迁移到 S3 目标的加载时间。为缩短已分区源数据的加载时间，您可以创建映射到源数据库

中每个表的分区的 S3 目标子文件夹。这些已绑定分区的子文件夹允许 AWS DMS 运行并行进程来填充目标上的每个子文件夹。

要配置 S3 目标的并行完全加载，S3 支持表映射 `table-settings` 规则的三种 `parallel-load` 规则类型：

- `partitions-auto`
- `partitions-list`
- `ranges`

有关这些并行加载规则类型的更多信息，请参阅[表和集合设置规则和操作](#)。

对于 `partitions-auto` 和 `partitions-list` 规则类型，AWS DMS 使用源端点中的各分区名称来标识目标子文件夹结构，如下所示。

```
bucket_name/bucket_folder/database_schema_name/table_name/partition_name/  
LOADseq_num.csv
```

此时，S3 目标上用于迁移和存储数据的子文件夹路径包括一个额外的 `partition_name` 子文件夹，并与同名的源分区相对应。之后，该 `partition_name` 子文件夹存储一个或多个 `LOADseq_num.csv` 文件，其中包含从指定源分区迁移的数据。此处，`seq_num` 是.csv 文件名的序列号后缀，例如名为 `LOAD00000001.csv` 的 .csv 文件中的序列号后缀 `00000001`。

但是，MongoDB 和 DocumentDB 等部分数据库引擎没有分区概念。对于这些数据库引擎，AWS DMS 将运行源段索引作为前缀添加到目标 .csv 文件名中，如下所示。

```
.../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
```

此处，文件 `SEGMENT1_LOAD00000001.csv` 和 `SEGMENT1_LOAD00000002.csv` 以相同的运行源段索引前缀 `SEGMENT1` 命名。之所以这样命名，是因为这两个 .csv 文件的迁移源数据与同一个运行源段索引相关联。另一方面，存储在每个目标 `SEGMENT2_LOAD00000009.csv` 和 `SEGMENT3_LOAD0000000A.csv` 文件中的迁移数据与不同的运行源段索引相关联。每个文件的文件名都以其运行段索引的名称作为前缀，即 `SEGMENT2` 和 `SEGMENT3`。



对于 `ranges` 并行加载类型，您可以使用 `table-settings` 规则的 `columns` 和 `boundaries` 设置来定义列名和列值。使用这些规则，您可以指定与分段名称对应的分区，如下所示。

```
"parallel-load": {
  "type": "ranges",
  "columns": [
    "region",
    "sale"
  ],
  "boundaries": [
    [
      "NORTH",
      "1000"
    ],
    [
      "WEST",
      "3000"
    ]
  ],
  "segment-names": [
    "custom_segment1",
    "custom_segment2",
    "custom_segment3"
  ]
}
```

此处，`segment-names` 设置定义了三个分区的名称，以便在 S3 目标上并行迁移数据。所迁移数据以并行方式加载并按顺序存储在分区子文件夹下的 `.csv` 文件中，如下所示。

```
.../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
```

此处，AWS DMS 将一系列 `.csv` 文件存储在三个分区子文件夹中。各个分区子文件夹中的一系列 `.csv` 文件从 `LOAD00000001.csv` 开始递增命名，直至所有数据已完成迁移。

在某些情况下，您可能不会使用 `segment-names` 设置为 `ranges` 并行加载类型明确命名分区子文件夹。在这种情况下，AWS DMS 应用在其 `table_name` 子文件夹下创建各系列 `.csv` 文件的默认设置。此时，AWS DMS 在各系列 `.csv` 文件的文件名前面加上运行源段索引的名称，如下所示。

```
.../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
```

## 使用 Amazon S3 作为 AWS DMS 的目标时的端点设置

您可以使用端点设置来配置 Amazon S3 目标数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定对应设置，或者在 [AWS CLI](#) 中使用符合 `--s3-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令。

下表显示了将 Amazon S3 作为目标时您可以使用的端点设置。

选项	描述
CsvNullValue	<p>一个可选参数，用于指定 AWS DMS 如何处理 null 值。在处理 null 值时，您可以使用此参数在写入目标时将用户定义的字符串作为 null 传递。例如，当目标列可以为空时，您可以使用此选项来区分空字符串值和空值。因此，如果您将此参数值设置为空字符串 (" " 或 " ")，AWS DMS 会将空字符串视为空值而不是 NULL。</p> <p>默认值：NULL</p> <p>有效值：任何有效的字符串</p> <p>例如：<code>--s3-settings '{"CsvNullValue": " "}'</code></p>
AddColumnName	<p>一个可选参数，当设置为 true 或 y 时，可用于将列名称信息添加到 .csv 输出文件中。</p> <p>此参数不能与 PreserveTransactions 或 CdcPath 一起使用。</p> <p>默认值：false</p> <p>有效值：true、false、y、n</p> <p>例如：<code>--s3-settings '{"AddColumnName": true}'</code></p>
AddTrailingPaddingCharacter	<p>使用 S3 目标端点设置 AddTrailingPaddingCharacter 为字符串数据添加填充。默认值为 false。</p> <p>类型：布尔值</p>

选项	描述
	<p>例如：<code>--s3-settings '{"AddTrailingPaddingCharacter": true}'</code></p>
BucketFolder	<p>一个可选参数，用于在 S3 桶中设置文件夹的名称。如果提供此参数，则在路径 <i>BucketFolder /schema_name /table_name /</i> 中创建目标对象作为 .csv 或 .parquet 文件。如果未指定此参数，则使用的路径为 <i>schema_name /table_name /</i>。</p> <p>例如：<code>--s3-settings '{"BucketFolder": "testFolder"}</code></p>
BucketName	<p>在其中将 S3 目标对象创建为 .csv 或 .parquet 文件的 S3 存储桶的名称。</p> <p>例如：<code>--s3-settings '{"BucketName": "buckettest"}</code></p>
CannedAclForObjects	<p>该值允许 AWS DMS 为在 S3 存储桶中创建为 .csv 或 .parquet 文件的对象指定预定义（标准）访问控制列表。有关 Amazon S3 标准 ACL 的更多信息，请参阅《Amazon S3 开发人员指南》中的<a href="#">标准 ACL</a>。</p> <p>默认值：NONE</p> <p>该属性的有效值为：NONE、PRIVATE、PUBLIC_READ、PUBLIC_READ_WRITE、AUTHENTICATED_READ、AWS_EXEC_READ、BUCKET_OWNER_READ、BUCKET_OWNER_FULL_CONTROL。</p> <p>例如：<code>--s3-settings '{"CannedAclForObjects": "PUBLIC_READ"}</code></p>

选项	描述
CdcInsertsOnly	<p>一个在更改数据捕获 (CDC) 加载期间仅将 INSERT 操作写入逗号分隔值 (.csv) 或列式存储 (.parquet) 输出文件的可选参数。默认情况下 ( false 设置 ) , .csv 或 .parquet 记录中的第一个字段包含字母 I (INSERT)、U (UPDATE) 或 D (DELETE)。该字母表示是在源数据库中插入、更新还是删除行, 以将 CDC 加载到目标。如果将 cdcInsertsOnly 设置为 true 或 y , 则仅将来自源数据库的 INSERT 迁移到 .csv 或 .parquet 文件。</p> <p>仅对于 .csv 格式, 记录这些 INSERTS 的方式取决于 IncludeOpForFullLoad 的值。如果将 IncludeOpForFullLoad 设置为 true , 则每条 CDC 记录的第一个字段都将设置为 I 以指示源上的 INSERT 操作。如果将 IncludeOpForFullLoad 设置为 false , 则写入每条 CDC 记录 ( 不带用于指示源上的 INSERT 操作的第一个字段 ) 。有关这些参数的协作方式的更多信息, 请参阅<a href="#">指示迁移的 S3 数据中的源数据库操作</a>。</p> <p>默认值 : false</p> <p>有效值 : true、false、y、n</p> <p>例如 : --s3-settings '{"CdcInsertsOnly": true}'</p>

选项	描述
CdcInsertsAndUpdates	<p>启用更改数据捕获 (CDC) 加载以将 INSERT 和 UPDATE 操作写入 .csv 或 .parquet (列式存储) 输出文件。默认设置为 false, 但当 cdcInsertsAndUpdates 设置为 true 或 y 时, 源数据库中的 INSERT 和 UPDATE 将迁移到 .csv 或 .parquet 文件。</p> <p>仅对于 .csv 文件格式, 记录这些 INSERT 和 UPDATE 的方式取决于 includeOpForFullLoad 参数的值。如果 includeOpForFullLoad 设置为 true, 则每条 CDC 记录的第一个字段都将设置为 I 或 U 以指示源上的 INSERT 和 UPDATE 操作。但如果 includeOpForFullLoad 设置为 false, 则 CDC 记录在不指示源上的 INSERT 和 UPDATE 操作的情况下写入。</p> <p>有关这些参数的协作方式的更多信息, 请参阅<a href="#">指示迁移的 S3 数据中的源数据库操作</a>。</p> <div data-bbox="472 894 1507 1209" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>对于同一终端节点, CdcInsertsOnly 和 cdcInsertsAndUpdates 不同时设置为 true。对于同一终端节点, 将 cdcInsertsOnly 或 cdcInsertsAndUpdates 设置为 true, 但不能同时设置这两者。</p> </div> <p>默认值 : false</p> <p>有效值 : true、false、y、n</p> <p>例如 : --s3-settings '{"CdcInsertsAndUpdates": true}'</p>

选项	描述
CdcPath	<p>指定 CDC 文件的文件夹路径。对于 S3 来源，如果任务捕获更改数据，则该设置是必需的，否则，该设置是可选的。如果设置了 CdcPath，则 DMS 将通过该路径读取 CDC 文件，然后将数据更改复制到目标端点。</p> <p>对于 S3 目标，如果将 PreserveTransactions 设置为 true，DMS 会验证是否已将此参数设置为 S3 目标上的文件夹路径，使 DMS 可在其中保存 CDC 加载的事务顺序。DMS 在 S3 目标工作目录或 BucketFolder 和 BucketName 指定的 S3 目标位置中创建此 CDC 文件夹路径。</p> <p>此参数不能与 DatePartitionEnabled 或 AddColumnName 一起使用。</p> <p>类型：字符串</p> <p>例如，如果您将 CdcPath 指定为 MyChangedData，且将 BucketName 指定为 MyTargetBucket，但没有指定 BucketFolder，DMS 会创建如下 CDC 文件夹路径：MyTargetBucket/MyChangedData。</p> <p>如果您指定相同的 CdcPath，并将 BucketName 指定为 MyTargetBucket，BucketFolder 指定为 MyTargetData，DMS 创建如下 CDC 文件夹路径：MyTargetBucket/MyTargetData/MyChangedData。</p> <div data-bbox="472 1199 1507 1514" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>AWS DMS 版本 3.4.2 及更高版本支持此设置。</p> <p>按事务顺序捕获数据更改时，无论目标上的 DataFormat S3 设置为何值，DMS 始终将行更改存储在 .csv 文件中。DMS 不会使用 .parquet 文件按事务顺序保存数据更改。</p> </div>
CdcMaxBatchInterval	<p>最大间隔长度条件（以秒为单位），之后将文件输出到 Amazon S3。</p> <p>默认值：60 秒</p> <p>指定 CdcMaxBatchInterval 并指定 CdcMinFileSize 时，文件写入由首先满足的参数条件触发。</p>

选项	描述
CdcMinFileSize	<p>将文件输出到 Amazon S3 所要达到的最小文件大小条件 ( 以千字节为单位 )。</p> <p>默认值 : 32000 KB</p> <p>指定 CdcMinFileSize 并指定 CdcMaxBatchInterval 时, 文件写入由首先满足的参数条件触发。</p>
PreserveTransactions	<p>如果设置为 true, DMS 将在 CdcPath 指定的 Amazon S3 目标上保存更改数据捕获 (CDC) 的事务顺序。</p> <p>此参数不能与 DatePartitionEnabled 或 AddColumnName 一起使用。</p> <p>类型 : 布尔值</p> <p>按事务顺序捕获数据更改时, 无论目标上的 DataFormat S3 设置为何值, DMS 始终将行更改存储在 .csv 文件中。DMS 不会使用 .parquet 文件按事务顺序保存数据更改。</p> <div data-bbox="472 1066 1507 1243"><p> Note</p><p>AWS DMS 版本 3.4.2 及更高版本支持此设置。</p></div>

选项	描述
IncludeOpForFullLoad	<p>一个在完全加载期间仅将 INSERT 操作写入逗号分隔值 (.csv) 输出文件的可选参数。</p> <p>对于完全加载，只能插入记录。默认情况下，( false 设置 )，这些输出文件中没有记录完全加载的信息，以指示行已插入源数据库。如果将 IncludeOpForFullLoad 设置为 true 或 y，则在 .csv 文件的第一个字段中将 INSERT 记录为 I 注释。</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>此参数与 CdcInsertsOnly 或 CdcInsertsAndUpdates 结合使用，以仅输出到 .csv 文件。有关这些参数的协作方式的更多信息，请参阅<a href="#">指示迁移的 S3 数据中的源数据库操作</a>。</p> </div> <p>默认值：false</p> <p>有效值：true、false、y、n</p> <p>例如：--s3-settings '{"IncludeOpForFullLoad": true}'</p>
CompressionType	<p>一个可选参数，当设置为 GZIP 时，用于使用 GZIP 压缩目标 .csv 或 .parquet 文件。当此参数设置为默认值时，它会使文件保持未压缩状态。</p> <p>默认值：NONE</p> <p>有效值：GZIP 或 NONE</p> <p>例如：--s3-settings '{"CompressionType": "GZIP"}</p>
CsvDelimiter	<p>.csv 源文件中用于分隔列的分隔符。默认值为逗号 (,)。</p> <p>例如：--s3-settings '{"CsvDelimiter": ","}'</p>
CsvRowDelimiter	<p>.csv 源文件中用于分隔行的分隔符。默认值为换行符 (\n)。</p> <p>例如：--s3-settings '{"CsvRowDelimiter": "\n"}</p>



选项	描述
MaxFileSize	<p>一个值，指定在完整加载期间，迁移到 S3 目标时所要创建的任意 .csv 文件的最大大小（以 KB 为单位）。</p> <p>默认值：1048576 KB (1 GB)</p> <p>有效值：1–1048576</p> <p>例如：<code>--s3-settings '{"MaxFileSize": 512}'</code></p>
Rfc4180	<p>一个可选参数，用于设置行为以符合 RFC，仅适用于使用 .csv 文件格式迁移到 Amazon S3 的数据。当此值设置为 true 或 y（使用 Amazon S3 作为目标）时，如果数据中有引号、逗号或换行符，则 AWS DMS 会使用一对额外的双引号 (") 将整个列引起来。数据中的每个引号将被重复两次。此格式符合 RFC 4180。</p> <p>默认值：true</p> <p>有效值：true、false、y、n</p> <p>例如：<code>--s3-settings '{"Rfc4180": false}'</code></p>
EncryptionMode	<p>要用于加密复制到 S3 的 .csv 或 .parquet 对象文件的服务器端加密模式。有效值为 SSE_S3（S3 服务器端加密）或 SSE_KMS（KMS 密钥加密）。如果您选择 SSE_KMS，请将 ServerSideEncryptionKmsKeyId 参数设置为要用于加密的 KMS 密钥的 Amazon 资源名称 (ARN)。</p> <div data-bbox="472 1325 1507 1591" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>您也可以使用 CLI <code>modify-endpoint</code> 命令将现有端点的 <code>EncryptionMode</code> 属性值从 SSE_KMS 更改为 SSE_S3。但您不能将 <code>EncryptionMode</code> 值从 SSE_S3 更改为 SSE_KMS。</p> </div> <p>默认值：SSE_S3</p> <p>有效值：SSE_S3 或 SSE_KMS</p> <p>示例：<code>--s3-settings '{"EncryptionMode": SSE_S3}'</code></p>

选项	描述
ServerSideEncryptionKmsKeyId	<p>如果您将 EncryptionMode 设置为 SSE_KMS，请将此参数设置为 KMS 密钥的 Amazon 资源名称 (ARN)。您可以通过在为您的账户创建的 AWS KMS 密钥列表中选择密钥别名来查找此 ARN。在创建密钥时，您必须将特定策略和角色与此 KMS 密钥关联。有关更多信息，请参阅<a href="#">创建 AWS KMS 密钥以加密 Amazon S3 目标对象</a>。</p> <p>例如：<code>--s3-settings '{"ServerSideEncryptionKmsKeyId":"arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'</code></p>
DataFormat	<p>AWS DMS 用于创建 S3 对象的文件的输出格式。对于 Amazon S3 目标，AWS DMS 支持 .csv 或 .parquet 文件。.parquet 文件具有一种二进制列式存储格式，并具有有效的压缩选项和更快的查询性能。有关 .parquet 文件的更多信息，请参阅<a href="https://parquet.apache.org/">https://parquet.apache.org/</a>。</p> <p>默认值：csv</p> <p>有效值：csv 或 parquet</p> <p>例如：<code>--s3-settings '{"DataFormat": "parquet"}'</code></p>
EncodingType	<p>Parquet 编码类型。编码类型选项包括：</p> <ul style="list-style-type: none"> <li>• rle-dictionary - 此词典编码使用位压缩和运行长度编码的组合以更高效地存储重复值。</li> <li>• plain - 无编码。</li> <li>• plain-dictionary - 此词典编码构建给定列中遇到的值的词典。该词典存储在每个列数据块的词典页面中。</li> </ul> <p>默认值：rle-dictionary</p> <p>有效值：rle-dictionary、plain 或 plain-dictionary</p> <p>例如：<code>--s3-settings '{"EncodingType": "plain-dictionary"}'</code></p>

选项	描述
DictPageSizeLimit	<p>.parquet 文件中的词典页面允许的最大大小 (以字节为单位)。如果词典页面超出此值, 则该页面使用纯编码。</p> <p>默认值: 1024000 (1 MB)</p> <p>有效值: 任何有效的整数值</p> <p>例如: <code>--s3-settings '{"DictPageSizeLimit": 2,048,000}'</code></p>
RowGroupLength	<p>.parquet 文件的一个行组中的行数。</p> <p>默认值: 10024 (10 KB)</p> <p>有效值: 任何有效整数</p> <p>例如: <code>--s3-settings '{"RowGroupLength": 20,048}'</code></p>
DataPageSize	<p>.parquet 文件中的数据页面允许的最大大小 (以字节为单位)。</p> <p>默认值: 1024000 (1 MB)</p> <p>有效值: 任何有效整数</p> <p>例如: <code>--s3-settings '{"DataPageSize": 2,048,000}'</code></p>
ParquetVersion	<p>.parquet 文件格式的版本。</p> <p>默认值: PARQUET_1_0</p> <p>有效值: PARQUET_1_0 或 PARQUET_2_0</p> <p>例如: <code>--s3-settings '{"ParquetVersion": "PARQUET_2_0"}</code></p>

选项	描述
EnableStatistics	<p>设置为 true 或 y 以启用有关 .parquet 文件页面和行组的统计数据。</p> <p>默认值 : true</p> <p>有效值 : true、false、y、n</p> <p>例如 : <code>--s3-settings '{"EnableStatistics": false}'</code></p>
TimestampColumnName	<p>一个可选参数，用于在 S3 目标终端节点数据中包含时间戳列。</p> <p>在将 TimestampColumnName 设置为非空值时，AWS DMS 在迁移数据的 .csv 或 .parquet 对象文件中包含一个额外的 STRING 列。</p> <p>对于完全加载，此时间戳列的每个行均包含一个时间戳，用于指示 DMS 将数据从源传输到目标的时间。</p> <p>对于 CDC 加载，时间戳列的每个行均包含该行在源数据库中的提交的时间戳。</p> <p>此时间戳列值的字符串格式为 <code>yyyy-MM-dd HH:mm:ss.SSSSSS</code>。默认情况下，此值是以微秒为单位的时间精度。对于 CDC 负载，精度的舍入取决于 DMS 支持的源数据库的提交时间戳。</p> <p>在将 AddColumnName 参数设置为 true 时，DMS 还包含设置为 TimestampColumnName 的非空值的时间戳列的名称。</p> <p>例如 : <code>--s3-settings '{"TimestampColumnName": "TIMESTAMP"}</code></p>

选项	描述
UseTaskStartTimeForFullLoadTimestamp	<p>设置为 <code>true</code> 时，此参数使用任务开始时间作为时间戳列值，而不是将数据写入目标的时间。对于完全加载，当 <code>UseTaskStartTimeForFullLoadTimestamp</code> 设置为 <code>true</code> 时，时间戳列的每一行都包含任务的开始时间。对于 CDC 加载，时间戳列的每个行均包含事务提交时间。</p> <p>当 <code>UseTaskStartTimeForFullLoadTimestamp</code> 被设置为 <code>false</code> 时，时间戳列中的满载时间戳会随着数据到达目标的时间而递增。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true</code>、<code>false</code></p> <p>例如：<code>--s3-settings '{"UseTaskStartTimeForFullLoadTimestamp": true}'</code></p> <p><code>UseTaskStartTimeForFullLoadTimestamp: true</code> 有助于使完全加载的 S3 目标 <code>TimestampColumnName</code> 可与 CDC 加载的 <code>TimestampColumnName</code> 一起排序。</p>

选项	描述
ParquetTimestampInMillisecond	<p>一个可选参数，指定写入 .parquet 格式 S3 对象文件的任意 <code>TIMESTAMP</code> 列值的精度。</p> <p>将此属性设置为 <code>true</code> 或 <code>y</code> 时，AWS DMS 以毫秒为精度，将所有 <code>TIMESTAMP</code> 列写入 .parquet 格式的文件中。否则，DMS 以微秒为精度写入它们。</p> <p>目前，Amazon Athena 和 AWS Glue 只能处理以毫秒为精度的 <code>TIMESTAMP</code> 值。对于 .parquet 格式的 S3 端点对象文件，仅在您计划使用 Athena 或 AWS Glue 查询或处理数据时，才设置此属性。</p> <div data-bbox="472 701 1507 1050" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <ul style="list-style-type: none"> <li>• AWS DMS 以 .csv 格式将任意 <code>TIMESTAMP</code> 列值写入到 S3 文件，精度为微秒。</li> <li>• 此属性的设置对通过设置 <code>TimestampColumnName</code> 属性插入的时间戳列的字符串格式没有任何影响。</li> </ul> </div> <p>默认值：<code>false</code></p> <p>有效值：<code>true</code>、<code>false</code>、<code>y</code>、<code>n</code></p> <p>例如：<code>--s3-settings '{"ParquetTimestampInMillisecond": true}'</code></p>
GlueCatalogGeneration	<p>要生成 AWS Glue Data Catalog，将此端点设置设为 <code>true</code>。</p> <p>默认值：<code>false</code></p> <p>有效值：<code>true</code>、<code>false</code></p> <p>例如：<code>--s3-settings '{"GlueCatalogGeneration": true}'</code></p> <p>注意：请勿将 <code>GlueCatalogGeneration</code> 与 <code>PreserveTransactions</code> 和 <code>CdcPath</code> 一起使用。</p>

## 将 AWS Glue Data Catalog 与 Amazon S3 目标一起用于 AWS DMS

AWS Glue 是一项提供简单数据分类方法的服务，由名为 AWS Glue Data Catalog 的元数据存储库组成。您可以将 AWS Glue Data Catalog 与 Amazon S3 目标端点集成，并通过其他 AWS 服务（例如，Amazon Athena）查询 Amazon S3 数据。Amazon Redshift 使用 AWS Glue，但 AWS DMS 不支持将其作为预建选项。

要生成数据目录，请将 GlueCatalogGeneration 端点设置设为 true，如以下 AWS CLI 示例所示。

```
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}'
```

对于包含 csv 类型数据的完全加载复制任务，将 IncludeOpForFullLoad 设置为 true。

**注意：**请勿将 GlueCatalogGeneration 与 PreserveTransactions 和 CdcPath 一起使用。AWS Glue 爬网程序无法协调存储在指定 CdcPath 下的不同文件架构。

要让 Amazon Athena 为您的 Amazon S3 数据编制索引，并且您能够通过 Amazon Athena 使用标准 SQL 查询来查询数据，附加到端点的 IAM 角色必须具有以下策略：

```
{
  "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_*"
}
]
}

```

## 参考信息

- 有关 AWS Glue 的更多信息，请参阅《AWS Glue 开发人员指南》中的[概念](#)。
- 有关 AWS Glue Data Catalog 的更多信息，请参阅《AWS Glue 开发人员指南》中的[组件](#)。



## 在 Amazon S3 目标上使用数据加密、parquet 文件和 CDC

您可以使用 S3 目标终端节点设置来配置以下内容：

- 用于加密 S3 目标对象的自定义 KMS 密钥。
- 作为 S3 目标对象的存储格式的 Parquet 文件。
- 将数据捕获 (CDC) 更改为包括 S3 目标上的事务顺序。
- 将 AWS Glue Data Catalog 与您的 Amazon S3 目标端点集成，并通过其他服务（例如，Amazon Athena）查询 Amazon S3 数据。

### 用于数据加密的 AWS KMS 密钥设置

以下示例说明如何配置自定义 KMS 密钥以加密 S3 目标对象。要开始，您可以运行以下 `create-endpoint` CLI 命令。

```
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"}'
```

此处，`--s3-settings` 选项所指定的 JSON 对象定义了两个参数。一个是带值 `SSE_KMS` 的 `EncryptionMode` 参数。另一个是带值 `arn:aws:kms:us-east-1:111122223333:key/72abb6fb-1e49-4ac1-9aed-c803dfcc0480` 的 `ServerSideEncryptionKmsKeyId` 参数。此值是自定义 KMS 密钥的 Amazon 资源名称 (ARN)。对于 S3 目标，您还可以指定其他设置。这些设置标识服务器访问角色，为默认 CSV 对象存储格式提供分隔符，并提供存储桶位置和名称来存储 S3 目标对象。

默认情况下，使用 S3 服务器端加密进行 S3 数据加密。对于上一示例中的 S3 目标，这也等同于指定其终端节点设置，如以下示例所示。

```
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"}
```

有关使用 S3 服务器端加密的更多信息，请参阅[使用服务器端加密保护数据](#)。

### Note

您也可以使用 CLI `modify-endpoint` 命令将现有端点的 `EncryptionMode` 参数值从 `SSE_KMS` 更改为 `SSE_S3`。但您不能将 `EncryptionMode` 值从 `SSE_S3` 更改为 `SSE_KMS`。

## 使用 .parquet 文件存储 S3 目标对象的设置

用于创建 S3 目标对象的默认格式为 .csv 文件。以下示例显示用于指定 .parquet 文件作为创建 S3 目标对象的格式的某些终端节点设置。您可以指定带所有默认值的 .parquet 文件格式，如以下示例所示。

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "DataFormat":
"parquet"}'
```

在这里，`DataFormat` 参数设置为 `parquet` 以启用带所有 S3 默认值的格式。这些默认值包括一个词典编码 (`"EncodingType": "rle-dictionary"`)，该编码使用位压缩和运行长度编码的组合以更高效地存储重复值。

您可以为选项添加默认值之外的其他设置，如以下示例所示。

```
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 }'
```

在这里，除了几个标准 S3 存储桶选项的参数和 `DataFormat` 参数外，将设置以下额外的 .parquet 文件参数：

- `EncodingType` – 设置为一个词典编码 (`plain-dictionary`)，该编码将每个列中遇到的值存储在词典页面的每列数据块中。
- `DictPageSizeLimit` – 最大词典页面大小设置为 3 MB。
- `EnableStatistics` – 禁用用来启用有关 Parquet 文件页面和行组的统计数据集合的默认值。

### 捕获数据更改 (CDC) 时包括 S3 目标上的事务顺序

默认情况下，在 AWS DMS 运行 CDC 任务时，它会将一个或多个源数据库中记录的所有行更改存储在每个表的一个或多个文件中。包含同一个表的更改的每组文件都位于与该表关联的单个目标目录中。AWS DMS 创建的目标目录与迁移到 Amazon S3 目标端点的数据库表一样多。文件存储在 S3 目标上的这些目录中，而不考虑事务顺序。有关文件命名惯例、数据内容和格式的更多信息，请参阅[将 Amazon S3 作为 AWS Database Migration Service 的目标](#)。

要以同时捕获事务顺序的方式捕获源数据库的更改，您可以指定相应的 S3 端点设置，指示 AWS DMS 将所有数据库表的行更改存储在一个或多个根据事务大小创建的 .csv 文件中。这些 .csv 事务文件包含按事务顺序依次列出的、每个事务中涉及的所有表的所有行更改。这些事务文件一起保存在您同样在 S3 目标上指定的单个事务目录中。在每个事务文件中，每处行更改的事务操作以及数据库和源表的标识都作为行数据的一部分存储，如下所示。

```
operation,table_name,database_schema_name,field_value,...
```

其中，*operation* 是对已更改行的事务操作，*table\_name* 是在其中更改该行的数据库表的名，*database\_schema\_name* 是该表所在数据库架构的名称，而 *field\_value* 是为该行指定数据的一个或多个字段值中的第一个。

事务文件之后的示例显示了涉及两个表的一个或多个事务中更改的行。

```
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
```

其中，每行的事务操作由第一列中的 I (插入)、U (更新) 或 D (删除) 表示。表名是第二列的值 (例如，Names\_03cdcad11a)。数据库架构的名称是第三列的值 (例如，rdsTempsdb)。其余列将使用您自己的行数据填充 (例如，13,Daniel)。

此外，AWS DMS 根据以下命名惯例，使用时间戳命名其在 Amazon S3 目标上创建的事务文件。

```
CDC_TXN-timestamp.csv
```

其中，*timestamp* 是创建事务文件的时间，如以下示例所示。

```
CDC_TXN-20201117153046033.csv
```

文件名中的该时间戳可确保按照您在事务目录中列出相关事务的事务顺序来创建和列出事务文件。

### Note

按事务顺序捕获数据更改时，无论目标上的 DataFormat S3 设置为何值，AWS DMS 始终将行更改存储在 .csv 文件中。AWS DMS 不会使用 .parquet 文件按事务顺序保存数据更改。

要控制在执行数据复制任务期间向 Amazon S3 目标写入的频率，您可以配置 `CdcMaxBatchInterval` 和 `CdcMinFileSize` 设置。这可以提高分析数据的性能，而无需任何额外的开销操作。有关更多信息，请参阅[使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。

指示 AWS DMS 按事务顺序存储所有行更改

1. 将目标上的 `PreserveTransactions S3` 设置设置为 `true`。
2. 将目标上的 `CdcPath S3` 设置设置为您希望 AWS DMS 在其中存储 .csv 事务文件的相对文件夹路径。

AWS DMS 在默认 S3 目标存储桶和工作目录下，或者在目标上使用 `BucketName` 和 `BucketFolder S3` 设置指定的存储桶和存储桶文件夹下创建此路径。

## 指示迁移的 S3 数据中的源数据库操作

当 AWS DMS 将记录迁移到 S3 目标时，它会在每条迁移的记录中创建一个额外字段。此额外字段指示应用于源数据库中的记录的操作。AWS DMS 如何创建和设置第一个字段取决于 `includeOpForFullLoad`、`cdcInsertsOnly` 和 `cdcInsertsAndUpdates` 的迁移任务和设置类型。

对于完全加载，当 `includeOpForFullLoad` 为 `true` 时，AWS DMS 始终在每条 .csv 记录中额外创建第一个字段。该字段包含字母 I (INSERT)，指示该行已插入源数据库中。对于 CDC 加载，当

`cdcInsertsOnly` 为 `false` (默认值) 时, AWS DMS 也始终在每条 `.csv` 或 `.parquet` 记录中额外创建第一个字段。此字段包含字母 I (INSERT)、U (UPDATE) 或 D (DELETE), 指示是否在源数据库中插入、更新或删除行。

在下表中, 您可以了解 `includeOpForFullLoad` 和 `cdcInsertsOnly` 属性的设置如何协作以影响所迁移记录的设置。

使用这些参数设置		DMS 将目标记录设置为 <code>.csv</code> 和 <code>.parquet</code> 输出的后续记录	
<code>includeOpForFullLoad</code>	<code>cdcInsertsOnly</code>	对于完全加载	对于 CDC 加载
<code>true</code>	<code>true</code>	添加的第一个字段值设置为 I	添加的第一个字段值设置为 I
<code>false</code>	<code>false</code>	无添加的字段	添加的第一个字段值设置为 I、U 或 D
<code>false</code>	<code>true</code>	无添加的字段	无添加的字段
<code>true</code>	<code>false</code>	添加的第一个字段值设置为 I	添加的第一个字段值设置为 I、U 或 D

在将 `includeOpForFullLoad` 和 `cdcInsertsOnly` 设置为同一值时, 将根据控制当前迁移类型的记录设置的属性来设置目标记录。该属性为 `includeOpForFullLoad` (对于完全加载) 和 `cdcInsertsOnly` (对于 CDC 加载)。

在将 `includeOpForFullLoad` 和 `cdcInsertsOnly` 设置为不同的值时, AWS DMS 使 CDC 加载和完全加载的目标记录设置保持一致。它通过使 CDC 加载的记录设置符合 `includeOpForFullLoad` 指定的任何早期完全加载的记录设置来实现此目的。

换句话说, 假设将完全加载设置为添加第一个字段来指示插入的记录。在此情况下, 将以下 CDC 加载设置为添加第一个字段, 该字段指示在源中适当插入、更新或删除的记录。相反, 假设将完全加载设置为不添加第一个字段来指示插入的记录。在此情况下, CDC 加载也将设置为不向每条记录添加第一个字段, 而不管其在源中的相应记录操作如何。

同样, DMS 如何创建和设置附加的第一个字段取决于 `includeOpForFullLoad` 和 `cdcInsertsAndUpdates` 的设置。在下表中, 您可以了解 `includeOpForFullLoad` 和 `cdcInsertsAndUpdates` 属性的设置如何协作以影响采用此格式的迁移记录的设置。

使用这些参数设置		DMS 将目标记录设置为 .csv 输出的后续记录	
includeOpForFullLoad	cdcInsertsAndUpdates	对于完全加载	对于 CDC 加载
true	true	添加的第一个字段值设置为 I	添加的第一个字段值设置为 I 或 U
false	false	无添加的字段	添加的第一个字段值设置为 I、U 或 D
false	true	无添加的字段	添加的第一个字段值设置为 I 或 U
true	false	添加的第一个字段值设置为 I	添加的第一个字段值设置为 I、U 或 D

## S3 Parquet 的目标数据类型

下表显示了使用 AWS DMS 时支持的 Parquet 目标数据类型以及来自 AWS DMS 数据类型的默认映射。

有关 AWS DMS 数据类型的其他信息，请参阅[AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	S3 parquet 数据类型
BYTES	BINARY
DATE	DATE32
TIME	TIME32
DATETIME	TIMESTAMP
INT1	INT8
INT2	INT16
INT4	INT32

AWS DMS 数据类型	S3 parquet 数据类型
INT8	INT64
NUMERIC	DECIMAL
REAL4	FLOAT
REAL8	DOUBLE
STRING	STRING
UINT1	UINT8
UINT2	UINT16
UINT4	UINT32
UINT8	UINT64
WSTRING	STRING
BLOB	BINARY
NCLOB	STRING
CLOB	STRING
BOOLEAN	BOOL

## 将 Amazon DynamoDB 数据库作为 AWS Database Migration Service 的目标

您可以使用 AWS DMS 将数据迁移到 Amazon DynamoDB 表。Amazon DynamoDB 是一种完全托管的 NoSQL 数据库服务，提供快速且可预测的性能，同时还能够实现无缝扩展。AWS DMS 支持使用关系数据库或 MongoDB 作为源。

在 DynamoDB 中，表、项目和属性是您使用的核心组件。表是项目的集合，而每个项目是属性的集合。DynamoDB 使用称为分区键的主键来唯一标识表中的各个项目。您还可以使用键和辅助索引来提供更具灵活性的查询。

您将使用对象映射将数据从源数据库迁移到目标 DynamoDB 表。使用对象映射可以确定源数据是否位于目标中。

当 AWS DMS 在 DynamoDB 目标端点上创建表时，它创建与源数据库端点相同数量的表。AWS DMS 还会设置几个 DynamoDB 参数值。创建表的成本取决于要迁移的数据量和表数。

#### Note

AWS DMS 控制台或 API 上的 SSL 模式选项不适用于某些数据流和 NoSQL 服务，例如 Kinesis 和 DynamoDB。默认情况下，它们是安全的，因此 AWS DMS 中显示的 SSL 模式设置是等于无 (SSL 模式 = 无)。您无需为端点提供任何其他配置即可使用 SSL。例如，使用 DynamoDB 作为目标端点时，默认情况下它是安全的。所有对 DynamoDB 的 API 调用都使用 SSL，因此无需在 AWS DMS 端点中添加额外的 SSL 选项。您可以使用 HTTPS 协议通过 SSL 端点安全地存放数据和检索数据，AWS DMS 在连接到 DynamoDB 数据库时默认使用该协议。

为了帮助提高传输速度，AWS DMS 支持多线程完全加载到 DynamoDB 目标实例。对于包含下列内容的任务设置，DMS 支持此多线程处理：

- **MaxFullLoadSubTasks** – 使用此选项指示要并行加载的表的最大数目。DMS 使用专用的子任务将各个表加载到其对应的 DynamoDB 目标表。默认值是 8。最大值为 49。
- **ParallelLoadThreads** – 使用此选项指定 AWS DMS 将各个表加载到其 DynamoDB 目标表时使用的线程数。默认值是 0 (单线程)。最大值为 200。您可以请求提高此最大值限制。

#### Note

DMS 将表的各个段分配到自己的线程进行加载。因此，将 **ParallelLoadThreads** 设置您为源中表指定的最大段数量。

- **ParallelLoadBufferSize** – 使用此选项指定在缓冲区 (并行加载线程将数据加载到 DynamoDB 目标时使用) 中存储的最大记录数。默认值是 50。最大值为 1,000。将此设置与 **ParallelLoadThreads** 一起使用；仅在有多个线程时 **ParallelLoadBufferSize** 才有效。
- 各个表的 **Table-mapping** 设置 – 使用 **table-settings** 规则标识您希望并行加载的源中各个表。另外可使用这些规则来指定如何为多线程加载的每个表的行分段。有关更多信息，请参阅[表和集合设置规则和操作](#)。



**Note**

当 AWS DMS 为迁移任务设置 DynamoDB 参数值时，默认读取容量单位 (RCU) 参数值设置为 200。

还会设置写入容量单位 (WCU) 参数值，但该参数值取决于几个其他设置：

- WCU 参数的默认值为 200。
- 如果 `ParallelLoadThreads` 任务设置为大于 1 (默认值为 0)，则 WCU 参数设置为 `ParallelLoadThreads` 值的 200 倍。
- 标准 AWS DMS 使用费适用于您使用的资源。

## 从关系数据库迁移到 DynamoDB 表

AWS DMS 支持将数据迁移到 DynamoDB 标量数据类型。从 Oracle 或 MySQL 等关系数据库迁移到 DynamoDB 时，您可能希望重构存储此数据的方式。

目前，AWS DMS 支持单个表到单个表重构到 DynamoDB 标量类型属性。如果您从关系数据库表迁移数据到 DynamoDB 中，您将从表中获取数据并将其重新格式化为 DynamoDB 标量数据类型属性。这些属性可以接受多个列中的数据，并且您可以直接将列映射到属性。

AWS DMS 支持以下 DynamoDB 标量数据类型：

- 字符串
- 数字
- 布尔值

**Note**

源中的 NULL 数据在目标上会被忽略。

## 将 DynamoDB 作为 AWS Database Migration Service 目标时的先决条件

在开始使用 DynamoDB 数据库作为 AWS DMS 的目标之前，请确保您已创建 IAM 角色。此 IAM 角色应允许 AWS DMS 代入并授予对要迁移到的 DynamoDB 表的访问权限。以下 IAM policy 中显示了所需的最小访问权限集合。

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

您在迁移到 DynamoDB 时使用的角色必须具有以下权限：

```
{
  "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": "*"
    }
  ]
}
```

```
}  
  ]  
}
```

## 将 DynamoDB 作为 AWS Database Migration Service 的目标时的限制

将 DynamoDB 作为目标时存在以下限制：

- DynamoDB 将数字数据类型的精度限制为 38 位。所有具有更高精度的数据类型应作为字符串存储。您需要使用对象映射功能明确指定这一点。
- 由于 DynamoDB 没有日期数据类型，使用日期数据类型的数据将转换为字符串。
- DynamoDB 不允许更新主键属性。在将持续复制与更改数据捕获 (CDC) 一起使用时，该限制是非常重要的，因为这可能会导致在目标中包含不需要的数据。根据对象映射方式，更新主键的 CDC 操作可执行下列两项操作之一。它可能会失败，或插入具有更新后的主键和不完整数据的新项。
- AWS DMS 仅支持复制包含非复合主键的表。当您为具有自定义分区键和/或排序键的目标表指定对象映射时除外。
- AWS DMS 不支持 LOB 数据，除非它是 CLOB。迁移数据时，AWS DMS 将 CLOB 数据转换为 DynamoDB 字符串。
- 使用 DynamoDB 作为目标时，仅 Apply Exceptions 控制表 (dmslogs.aws\_dms\_apply\_exceptions) 受支持。有关控制表的更多信息，请参阅[控制表任务设置](#)。
- 当 DynamoDB 作为目标时，AWS DMS 不支持任务设置 TargetTablePrepMode=TRUNCATE\_BEFORE\_LOAD。
- 当 DynamoDB 作为目标时，AWS DMS 不支持任务设置 TaskRecoveryTableEnabled。

## 使用对象映射将数据迁移到 DynamoDB

AWS DMS 使用表映射规则将数据从源映射到目标 DynamoDB 表。要将数据映射到 DynamoDB 目标，您必须使用称为 object-mapping 的表映射规则类型。利用对象映射可以定义属性名称以及要迁移到其中的数据。您在使用对象映射时必须具有选择规则。

除了具有分区键和可选的排序键以外，DynamoDB 没有预设结构。如果有非复合主键，AWS DMS 将使用该主键。如果您有复合主键或者希望使用排序键，请在目标 DynamoDB 表中定义这些键以及其他属性。

要创建对象映射规则，您应将 rule-type 指定为 object-mapping。此规则指定您要使用的对象映射的类型。

规则的结构如下所示：

```
{ "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 目前支持 `map-record-to-record` 和 `map-record-to-document` 作为 `rule-action` 参数的唯一有效值。这些值指定 AWS DMS 默认情况下对未作为 `exclude-columns` 属性列表的一部分排除的记录执行的操作。这些值不会以任何方式影响属性映射。

- 在从关系数据库迁移到 DynamoDB 时，可以使用 `map-record-to-record`。它使用关系数据库的主键作为 DynamoDB 中的分区键，并为源数据库中的每个列创建一个属性。在使用 `map-record-to-record` 时，对于源表中未在 `exclude-columns` 属性列表中列出的任何列，AWS DMS 将在目标 DynamoDB 实例上创建对应的属性。不论是否在属性映射中使用源列，都会执行此操作。
- 使用 `map-record-to-document` 将源列放入目标上的、使用属性名称“`_doc`”的单个扁平化 DynamoDB 映射中。在使用 `map-record-to-document` 时，AWS DMS 将数据放入源上的单个扁平化 DynamoDB 映射属性中。此属性称为“`_doc`”。此放置应用于源表中的未在 `exclude-columns` 属性列表中列出的任何列。

了解 `rule-action` 参数 `map-record-to-record` 和 `map-record-to-document` 之间的差异的一种方法是，查看两个参数的实际使用情况。对于本示例，假定您使用关系数据库表行开始处理，该行具有以下结构和数据：

FirstName	LastName	NickName	WorkAddress	WorkPhone	HomeAddress	HomePhone	Income
▶ Daniel	Sheridan	Dan	101 Main St Cambridge, MA	800-867-5309	100 Secret St, Unknownville, MA	123-456-7890	12345678

要将此信息迁移到 DynamoDB，您将创建规则来将数据映射到 DynamoDB 表项目中。请注意为 `exclude-columns` 参数列出的列。这些列不直接映射到目标上；而是使用属性映射将数据

组合到新项目中，例如，将 `FirstName` 和 `LastName` 分组到一起以成为 DynamoDB 目标上的 `CustomerName`。未排除 `NickName` 和 `income`。

```
{
  "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}"
          }
        }
      }
    }
  }
}
```

通过使用 `rule-action` 参数 `map-record-to-record`，`NickName` 和 `income` 的数据将映射到 DynamoDB 目标中同名的项目。

```

  ▼ 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

```

但是，假设您使用相同的规则，但将 `rule-action` 参数更改为 `map-record-to-document`。在这种情况下，`exclude-columns` 参数中未列出的列 `NickName` 和 `income` 将映射到 `_doc` 项目。

```

  ▼ 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

```

### 使用自定义条件表达式和对象映射

您可以使用 DynamoDB 中称为条件表达式的功能来操作写入到 DynamoDB 表的数据。有关 DynamoDB 中条件表达式的更多信息，请参阅[条件表达式](#)。

条件表达式成员包括：

- 一个表达式 (必需)
- 表达式属性值 (可选)。指定属性值的 DynamoDB json 结构

- 表达式属性名 (可选)
- 确定何时使用条件表达式的选项 (可选)。默认值为 `apply-during-cdc = false` 和 `apply-during-full-load = true`

规则的结构如下所示：

```
"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>
    }
  }
```

下面的示例强调了用于条件表达式的部分。





## 使用属性映射和对象映射

利用属性映射可以使用源列名来指定模板字符串，以便重建目标上数据的结构。除了用户在模板中指定的内容以外，不进行任何其他格式设置。

以下示例显示了源数据库的结构和所需的 DynamoDB 目标结构。首先显示的是源结构，在本例中为 Oracle 数据库，然后显示 DynamoDB 中所需的数据结构。该示例在结尾处提供了用于创建所需目标结构的 JSON。

Oracle 数据的结构如下所示：

First	Las	St	Home/	HomeF	WorkAddr	Worl	DateOfBirth
			SS		SS		
主键			不适用				

First	Last	Street	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateOfBirth
Randy	Marshall	5 Baker Street	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876541230	02/29/1988

DynamoDB 数据的结构如下所示：

CustomerName	StoreId	ContactDetails	DateOfBirth
分区键	排序键	不适用	
Randy Marshall	5	<pre>{   "Name": "Randy",   "Home": {     "Address": "221B Baker Street",     "Phone": 1234567890   },   "Work": {     "Address": "31 Spooner Street, Quahog",     "Phone": 9876541230   } }</pre>	02/29/1988

下面的 JSON 显示用于实现 DynamoDB 结构的对象映射和列映射：

```
{
  "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}\"}}"}
    }
  ]
}
]
}

```

使用列映射的另一种方法是使用 DynamoDB 格式作为您的文档类型。以下代码示例将 dynamodb-map 作为属性映射的 attribute-sub-type。

```

{
  "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": {
```



## 示例 1：使用属性映射和对象映射

以下示例从两个 MySQL 数据库表 `nfl_data` 和 `sport_team` 将数据迁移到名为 `NFLTeams` 和 `SportTeams` 的两个 DynamoDB 表。表的结构以及用于将数据从 MySQL 数据库表映射到 DynamoDB 表的 JSON 如下所示。

MySQL 数据库表 `nfl_data` 的结构如下所示：

```
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    |       |
+-----+-----+-----+-----+-----+-----+
```

MySQL 数据库表 `sport_team` 的结构如下所示：

```
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    |               |
+-----+-----+-----+-----+-----+-----+
```

将这两个表映射到两个 DynamoDB 表时使用的表映射规则如下所示：

```
{
  "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}\"}"
    }
]
}
}
]
}

```

NFLTeams DynamoDB 表的示例输出如下所示：

```

    "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"
}

```

SportsTeams DynamoDB 表的示例输出如下所示：

```

{
    "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"
}

```

## DynamoDB 的目标数据类型

AWS DMS 的 DynamoDB 端点支持大多数 DynamoDB 数据类型。下表显示了使用 AWS DMS 时支持的 Amazon AWS DMS 目标数据类型以及来自 AWS DMS 数据类型的默认映射。

有关 AWS DMS 数据类型的其他信息，请参阅[AWS Database Migration Service 的数据类型](#)。

AWS DMS 从异构数据库迁移数据时，我们将数据类型从源数据库映射到称为 AWS DMS 数据类型的中间数据类型。然后，我们将中间数据类型映射到目标数据类型。下表列出了各种 AWS DMS 数据类型及其映射到 DynamoDB 中的数据类型：

AWS DMS 数据类型	DynamoDB 数据类型
String	String
WString	String
布尔值	布尔值
Date	String
DateTime	String
INT1	数字
INT2	数字
INT4	数字
INT8	数字
数值	数字

AWS DMS 数据类型	DynamoDB 数据类型
Real4	数字
Real8	数字
UINT1	数字
UINT2	数字
UINT4	数字
UINT8	数字
CLOB	String

## 使用 Amazon Kinesis Data Streams 作为目标 AWS Database Migration Service

您可以使用将数据迁移 AWS DMS 到 Amazon Kinesis 数据流。Amazon Kinesis 数据流是 Amazon Kinesis Data Streams 服务的一部分。可以使用 Kinesis 数据流实时收集和处理大型数据记录流。

Kinesis 数据流由分片组成。分片是流中数据记录的唯一标识序列。有关 Amazon Kinesis Data Streams 中的分片的更多信息，请参阅《Amazon Kinesis Data Streams 开发人员指南》中的[分片](#)。

AWS Database Migration Service 使用 JSON 将记录发布到 Kinesis 数据流。在转换期间，AWS DMS 将每个记录从源数据库序列化到 JSON 格式的属性/值对或 JSON\_UNFORMATTED 消息格式。JSON\_UNFORMATTED 消息格式是带有换行符的单行 JSON 字符串。它允许 Amazon Data Firehose 将 Kinesis 数据传送到亚马逊 S3 目的地，然后使用包括亚马逊 Athena 在内的各种查询引擎对其进行查询。

您将使用对象映射将数据从支持的数据源迁移到目标流。使用对象映射，您确定如何在流中建立数据记录结构。您还可以为每个表定义分区键，Kinesis Data Streams 用它来将数据分组为分片。

在 Kinesis Data Streams 目标端点上 AWS DMS 创建表时，它创建的表与源数据库终端节点中创建的表数量一样多。AWS DMS 还会设置多个 Kinesis Data Streams 参数值。创建表的成本取决于要迁移的数据量和表数。

**Note**

AWS DMS 控制台或 API 上的 SSL 模式选项不适用于某些数据流和 NoSQL 服务，例如 Kinesis 和 DynamoDB。默认情况下，它们是安全的，因此 AWS DMS 显示 SSL 模式设置等于无（SSL 模式=无）。您无需为端点提供任何配置即可使用 SSL。例如，使用 Kinesis 作为目标端点时，默认情况下它是安全的。所有对 Kinesis 的 API 调用都使用 SSL，因此无需在终端节点中添加额外的 SSL 选项。AWS DMS 您可以使用 HTTPS 协议安全地存放数据和通过 SSL 端点检索数据，Kinesis 数据流在连接到 AWS DMS 时默认使用该协议。

## Kinesis Data Streams 端点设置

当你使用 Kinesis Data Streams 目标端点时，你可以使用 `KinesisSettings` API 中的 AWS DMS 选项获取交易和控制细节。

您可以使用以下方法之一设置连接：

- 在 AWS DMS 控制台中，使用端点设置。
- 在 CLI 中，使用 [CreateEndpoint](#) 命令的 `kinesis-settings` 选项。

在 CLI 中，使用 `kinesis-settings` 选项的下列请求参数：

**Note**

AWS DMS 版本 3.4.1 及更高版本支持 `IncludeNullAndEmpty` 端点设置。但是，中支持 Kinesis Data Streams 目标的以下其他端点设置。AWS DMS

- `MessageFormat` – 在端点上创建的记录的输出格式。消息格式为 JSON（默认值）或 `JSON_UNFORMATTED`（单行，无制表符）。
- `IncludeControlDetails` – 显示 Kinesis 消息输出中的表定义、列定义以及表和列更改的详细控制信息。默认值为 `false`。
- `IncludeNullAndEmpty` – 在目标中包括空列。默认值为 `false`。
- `IncludePartitionValue` – 在 Kinesis 消息输出中显示分区值，除非分区类型为 `schema-table-type`。默认值为 `false`。

- `IncludeTableAlterOperations` – 包括更改控制数据中表的任何数据定义语言 ( DDL ) 操作，例如 `rename-table`、`drop-table`、`add-column`、`drop-column` 和 `rename-column`。默认值为 `false`。
- `IncludeTransactionDetails` – 提供源数据库中的详细事务信息。此信息包括提交时间戳、日志位置以及 `transaction_id`、`previous_transaction_id` 和 `transaction_record_id` ( 事务内的记录偏移 ) 的值。默认值为 `false`。
- `PartitionIncludeSchemaTable` – 当分区类型为 `primary-key-type` 时，将架构和表名称作为分区值的前缀。这样做会增加 Kinesis 分片之间的数据分布。例如，假设 SysBench 架构具有数千个表，并且每个表的主键只有有限的范围。在此况下，同一主键将从数千个表发送到同一个分片，这会导致限制。默认值为 `false`。

以下示例显示在使用 AWS CLI 发出的示例 `create-endpoint` 命令中使用的 `kinesis-settings` 选项。

```
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
```

## 多线程完全加载任务设置

为了帮助提高传输速度，AWS DMS 支持对 Kinesis Data Streams 目标实例进行多线程满载。对于包含下列内容的任务设置，DMS 支持此多线程处理：

- `MaxFullLoadSubTasks` – 使用此选项指示要并行加载的表的最大数目。DMS 使用专用子任务将各个表加载到其对应的 Kinesis 目标表。默认值为 8；最大值为 49。
- `ParallelLoadThreads`— 使用此选项指定用于将每个表加载到其 Kinesis 目标表的线程数。AWS DMS Kinesis Data Streams 目标的最大值为 32。您可以请求提高此最大值限制。
- `ParallelLoadBufferSize` – 使用此选项指定在缓冲区 ( 并行加载线程将数据加载到 Kinesis 目标时使用 ) 中存储的最大记录数。默认值是 50。最大值为 1000。将此设置与 `ParallelLoadThreads` 一起使用；仅在有多个线程时 `ParallelLoadBufferSize` 才有效。

- `ParallelLoadQueuesPerThread` – 使用此选项可以指定每个并发线程访问的队列数，以便从队列中取出数据记录并为目标生成批处理负载。默认值是 1。但是，对于各种负载大小的 Kinesis 目标，有效范围为每个线程 5-512 个队列。

## 多线程 CDC 加载任务设置

您可以提高实时数据流目标端点的更改数据捕捉 ( CDC ) 的性能，例如，Kinesis 使用任务设置来修改 `PutRecords` API 调用的行为。为此，您可以使用 `ParallelApply*` 任务设置来指定并发线程的数量、每个线程的队列数以及要存储在缓冲区中的记录数。例如，假设您要执行 CDC 加载并且要并行应用 128 个线程。您还希望对于每个线程访问 64 个队列，每个缓冲区存储 50 条记录。

为了提高 CDC 性能，AWS DMS 支持以下任务设置：

- `ParallelApplyThreads`— 指定在 CDC 加载期间 AWS DMS 用于将数据记录推送到 Kinesis 目标端点的并发线程数。默认值为零 ( 0 )，最大值为 32。
- `ParallelApplyBufferSize` – 指定在 CDC 加载过程中，要在每个缓冲区队列中存储的、供并发线程推送到目标 Kinesis 端点的最大记录数。默认值是 100，最大值是 1,000。当 `ParallelApplyThreads` 指定多个线程时，请使用此选项。
- `ParallelApplyQueuesPerThread` – 指定每个线程访问以将数据记录从队列中取出并在 CDC 期间为 Kinesis 端点生成批处理负载的队列数。默认值是 1，最大值是 512。

使用 `ParallelApply*` 任务设置时，`partition-key-type` 默认值是表的 `primary-key`，而不是 `schema-name.table-name`。

## 使用之前映像查看 Kinesis 数据流 ( 作为目标 ) 的 CDC 行的原始值

在将 CDC 更新写入数据流目标 ( 如 Kinesis ) 时，可以在更新进行更改之前查看源数据库行的原始值。为此，请根据源数据库引擎提供的数据 AWS DMS 填充更新事件之前的图像。

不同的源数据库引擎为之前映像提供不同的信息量：

- 仅当列发生更改时，Oracle 才会对列提供更新。
- PostgreSQL 仅为作为主键一部分的列 ( 已更改或未更改 ) 提供数据。要为所有列提供数据 ( 无论是否更改 )，您需要将 `REPLICA_IDENTITY` 设置为 `FULL` 而不是 `DEFAULT`。请注意，您应该仔细选择每张表的 `REPLICA_IDENTITY` 设置。如果将 `REPLICA_IDENTITY` 设置为 `FULL`，则所有列值都将连续写入预写日志 ( WAL )。这可能会导致经常更新的表出现性能或资源问题。
- MySQL 通常为 BLOB 和 CLOB 数据类型以外的所有列 ( 已更改或未更改 ) 提供数据。

要启用之前映像以便将源数据库中的原始值添加到 AWS DMS 输出，请使用 `BeforeImageSettings` 任务设置或 `add-before-image-columns` 参数。此参数应用列转换规则。

`BeforeImageSettings` 使用从源数据库系统收集的值向每个更新操作添加一个新的 JSON 属性，如下所示。

```
"BeforeImageSettings": {
  "EnableBeforeImage": boolean,
  "FieldName": string,
  "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

#### Note

仅 `BeforeImageSettings` 适用于包含 CDC 组件的 AWS DMS 任务，例如满负荷加上 CDC 任务（迁移现有数据并复制正在进行的更改），或仅适用于 CDC 的任务（仅复制数据更改）。不将 `BeforeImageSettings` 应用于仅完全加载的任务。

对于 `BeforeImageSettings` 选项，以下条件适用：

- 将 `EnableBeforeImage` 选项设置为 `true` 以启用之前映像。默认值为 `false`。
- 使用 `FieldName` 选项为新 JSON 属性指定名称。当 `EnableBeforeImage` 为 `true` 时，`FieldName` 是必填项且不能为空。
- `ColumnFilter` 选项指定要使用之前映像添加的列。要仅添加属于表主键一部分的列，请使用默认值 `pk-only`。要添加具有之前映像值的任何列，请使用 `all`。请注意，之前映像不包括含有 LOB 数据类型（如 CLOB 或 BLOB）的列。

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```



**Note**

Amazon S3 目标不支持 `BeforeImageSettings`。对于 S3 目标，在 CDC 期间仅使用 `add-before-image-columns` 转换规则来执行之前映像。

### 使用之前映像转换规则

作为任务设置的替代方法，您可以使用 `add-before-image-columns` 参数，该参数应用列转换规则。使用此参数，您可以在 CDC 期间对数据流目标（如 Kinesis）启用之前映像。

通过在转换规则中使用 `add-before-image-columns`，可以对之前映像结果应用更精细的控制。转换规则允许您使用对象定位器，该定位器允许您控制为规则选择的表。此外，您可以将转换规则链接在一起，这样可以将不同的规则应用于不同的表。然后，您可以操控使用其他规则生成的列。

**Note**

不要在同一任务中将 `add-before-image-columns` 参数与 `BeforeImageSettings` 任务设置结合使用。而是对单个任务使用此参数或此设置，但不要同时使用这两者。

包含列的 `add-before-image-columns` 参数的 `transformation` 规则类型必须提供一个 `before-image-def` 部分。下面是一个示例。

```
{
  "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,
  }
}
```

`column-prefix` 的值附加到列名称前面，`column-prefix` 的默认值为 `BI_`。`column-suffix` 的值将附加到列名称之后，默认值为空。不要同时将 `column-prefix` 和 `column-suffix` 设置为空字符串。

为 `column-filter` 选择一个值。要仅添加属于表主键一部分的列，请选择 `pk-only`。选择 `non-lob` 以仅添加不属于 LOB 类型的列。或者，选择 `all` 以添加任何具有之前映像值的列。

### 之前映像转换规则的示例

以下示例中的转换规则在目标中添加一个名为 `BI_emp_no` 的新列。因此，像 `UPDATE employees SET emp_no = 3 WHERE emp_no = 1`；这样的语句用 1 填充 `BI_emp_no` 字段。当您将 CDC 更新写入 Amazon S3 目标时，通过 `BI_emp_no` 列能够判断哪个原始行已更新。

```
{
  "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"
      }
    }
  ]
}
```

有关使用 `add-before-image-columns` 规则操作的信息，请参阅 [转换规则和操作](#)。

## 使用 Kinesis 数据流作为目标的先决条件 AWS Database Migration Service

### 用于使用 Kinesis 数据流作为目标的 IAM 角色 AWS Database Migration Service

在将 Kinesis 数据流设置为目标之前 AWS DMS，请务必创建一个 IAM 角色。此角色必须允许 AWS DMS 代入并授予对正在迁移到的 Kinesis 数据流的访问权限。以下 IAM policy 中显示了所需的最小访问权限集合。

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

您在迁移到 Kinesis 数据流时使用的角色必须具有以下权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesis:DescribeStream",
        "kinesis:PutRecord",
        "kinesis:PutRecords"
      ],
      "Resource": "arn:aws:kinesis:region:accountID:stream/streamName"
    }
  ]
}
```

## 将 Kinesis 数据流作为目标进行访问 AWS Database Migration Service

在 3.4.7 及更高 AWS DMS 版本中，要连接到 Kinesis 终端节点，必须执行以下操作之一：

- 将 DMS 配置为使用 VPC 端点。有关将 DMS 配置为使用 VPC 端点的信息，请参阅[将 VPC 端点配置为 AWS DMS 源端点和目标端点](#)。
- 将 DMS 配置为使用公有路由，即，公开您的复制实例。有关公有复制实例的信息，请参阅[公有和私有复制实例](#)。

## 使用 Kinesis Data Streams 作为目标时的限制 AWS Database Migration Service

将 Kinesis Data Streams 作为目标时存在以下限制：

- AWS DMS 无论事务如何，都会将每次更新作为给定 Kinesis 数据流中的一条数据记录发布到源数据库中的一条记录。但是，您可以使用 KinesisSettings API 的相关参数包含每条数据记录的事务详细信息。
- 不支持完整 LOB 模式。
- 支持的最大 LOB 大小为 1 MB。
- Kinesis Data Streams 不支持重复数据删除。使用流中数据的应用程序需要处理重复记录。有关更多信息，请参阅《Amazon Kinesis Data Streams 开发人员指南》中的[处理重复记录](#)。
- AWS DMS 支持以下两种形式的分区键：
  - SchemaName.TableName：架构和表名称的组合。
  - \${AttributeName}：JSON 中其中一个字段的值，或源数据库中表的主键。
- 有关在 Kinesis Data Streams 中加密静态数据的信息，请参阅《AWS Key Management Service 开发人员指南》中的[Kinesis Data Streams 中的数据保护](#)。
- Kinesis 端点不支持 BatchApply。对 Kinesis 目标使用“批应用”（例如，BatchApplyEnabled 目标元数据任务设置）可能会导致数据丢失。
- 只有与复制实例相同 AWS 账户的 Kinesis 数据流才支持 Kinesis AWS 区域 s 目标。
- 从 MySQL 源迁移时，BeforeImage 数据不包括 CLOB 和 BLOB 数据类型。有关更多信息，请参阅[使用之前映像查看 Kinesis 数据流（作为目标）的 CDC 行的原始值](#)。
- AWS DMS 不支持迁移超过 16 位 BigInt 数据类型的值。要解决此限制问题，您可以使用以下转换规则将 BigInt 列转换为字符串。有关转换规则的更多信息，请参阅[转换规则和操作](#)。

```
{
  "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
}
}
```

## 使用对象映射将数据迁移到 Kinesis 数据流

AWS DMS 使用表映射规则将数据从源映射到目标 Kinesis 数据流。要将数据映射到目标流，您必须使用称为 object mapping 的表映射规则类型。您可以使用对象映射来定义源中的数据记录如何映射到发布到 Kinesis 数据流的数据记录。

除了具有分区键以外，Kinesis 数据流没有预设结构。在对象映射规则中，数据记录的 `partition-key-type` 的可能值为 `schema-table`、`transaction-id`、`primary-key`、`constant` 和 `attribute-name`。

要创建对象映射规则，您应将 `rule-type` 指定为 `object-mapping`。此规则指定您要使用的对象映射的类型。

规则的结构如下所示。

```
{
  "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 目前支持将 `map-record-to-record` 和 `map-record-to-document` 作为该 `rule-action` 参数的唯一有效值。这些设置会影响未作为 `exclude-columns` 属性列表一部分排除的值。`map-record-to-record` 和 `map-record-to-document` 值指定默认情况下如何 AWS DMS 处理这些记录。这些值不会以任何方式影响属性映射。

从关系数据库迁移到 Kinesis 数据流时使用 `map-record-to-record`。此规则类型使用关系数据库的 `taskResourceId.schemaName.tableName` 值作为 Kinesis 数据流中的分区键，并为源数据库中的每个列创建一个属性。

使用 `map-record-to-record` 时请注意：

- 此设置仅影响 `exclude-columns` 列表排除的列。
- 对于每个这样的列，在目标主题中 AWS DMS 创建一个相应的属性。
- AWS DMS 无论源列是否用于属性映射，都会创建相应的属性。

通过使用 `map-record-to-document`，可使用属性名“`_doc`”将源列放入相应目标流中的单个平面文档中。AWS DMS 将数据放入源上名为“`_doc`”的单个平面映射中。此放置应用于源表中的未在 `exclude-columns` 属性列表中列出的任何列。

了解 `map-record-to-record` 的一种方法是在操作时加以观察。对于本示例，假定您使用关系数据库表行开始处理，该行具有以下结构和数据。

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876543210	02/29/1988

要将此信息从名为 `Test` 的架构迁移到 Kinesis 数据流，您将创建规则来将数据映射到目标流。以下规则对此映射进行了说明。

```
{
  "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"
  }
}
]
```

以下内容说明 Kinesis 数据流中生成的记录格式。

- StreamName: XXX
- PartitionKey: test.Customers //schmaname.tableNam
- 数据 : //The following JSON message

```
{
  "FirstName": "Randy",
  "LastName": "Marsh",
  "StoreId": "5",
  "HomeAddress": "221B Baker Street",
  "HomePhone": "1234567890",
  "WorkAddress": "31 Spooner Street, Quahog",
  "WorkPhone": "9876543210",
  "DateOfBirth": "02/29/1988"
}
```

但是，假设您使用相同的规则，但将 `rule-action` 参数更改为 `map-record-to-document` 并排除某些列。以下规则对此映射进行了说明。

```
{
  "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"
        ]
      }
    }
  ]
}
```

在这种情况下，`exclude-columns` 参数中未列出的列 `FirstName`、`LastName`、`StoreId` 和 `DateOfBirth` 将映射到 `_doc`。以下内容说明生成的记录格式。

```
{
  "data":{
```



```
    "_doc":{
      "FirstName": "Randy",
      "LastName": "Marsh",
      "StoreId": "5",
      "DateOfBirth": "02/29/1988"
    }
  }
}
```

## 使用属性映射调整数据结构

在使用属性映射将数据迁移到 Kinesis 数据流时，您可以调整数据结构。例如，您可能希望将源中的多个字段合并到目标中的单个字段中。以下属性映射说明如何调整数据结构。

```
{
  "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}"
                }
            }
        }
    ]
}

```

要为 `partition-key` 设置常量值，请指定 `partition-key` 值。例如，您可以执行此操作来强制将所有数据存储在一个分片内。以下映射说明了此方法。

```

{
  "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

表示特定表的控制记录的 `partition-key` 值为 `TaskId.SchemaName.TableName`。  
表示特定任务的控制记录的 `partition-key` 值为该记录的 `TaskId`。在对象映射中指定 `partition-key` 值不会影响控制记录的 `partition-key`。

## Kinesis Data Streams 的消息格式

JSON 输出只是键值对的列表。JSON\_UNFORMATTED 消息格式是带有换行符的单行 JSON 字符串。

AWS DMS 提供了以下保留字段，以便更轻松地使用 Kinesis Data Streams 中的数据：

### RecordType

记录类型可以是数据或控制。数据记录表示源中的实际行。控制记录表示流中的重要事件，例如，重新开始任务。

### 操作

对于数据记录，操作可以是 `load`、`insert`、`update` 或 `delete`。

对于控制记录，操作可以是 `create-table`、`rename-table`、`drop-table`、`change-columns`、`add-column`、`drop-column`、`rename-column` 或 `column-type-change`。

### SchemaName

记录的源架构。此字段对于控制记录可能是空的。

### TableName

记录的源表。此字段对于控制记录可能是空的。

## Timestamp

JSON 消息构建时间的时间戳。此字段采用 ISO 8601 格式。

## 使用 Apache Kafka 作为目标 AWS Database Migration Service

您可以使用将数据迁移 AWS DMS 到 Apache Kafka 集群。Apache Kafka 是一个分布式流式处理平台。您可以使用 Apache Kafka 实时提取和处理流数据。

AWS 还提供适用于 Apache Kafka 的亚马逊托管流媒体 (亚马逊 MSK) 作为目标。AWS DMS Amazon MSK 是一个完全托管的 Apache Kafka 流式处理服务，可简化 Apache Kafka 实例的实施和管理。它适用于开源 Apache Kafka 版本，你可以像任何 Apache Kafka 实例一样作为 AWS DMS 目标访问亚马逊 MSK 实例。有关更多信息，请参阅《Amazon Managed Streaming for Apache Kafka 开发人员指南》中的 [Amazon MSK 是什么？](#)。

Kafka 集群将记录流存储在称为主题类别中，这些类别被划分为多个分区。分区是主题中唯一标识的数据记录 (消息) 序列。分区可分布在集群中的多个代理之间，以支持并行处理某个主题的记录。有关 Apache Kafka 中主题和分区及其分布的更多信息，请参阅[主题和日志](#)和[分布](#)。

您的 Kafka 集群可以是 Amazon MSK 实例、在 Amazon EC2 实例上运行的集群或本地集群。Amazon MSK 实例或 Amazon EC2 实例上的集群可以位于同一 VPC 中，也可以位于不同 VPC 中。如果您的集群位于本地，可以为复制实例使用自己的本地名称服务器以解析集群的主机名。有关在复制实例上设置名称服务器的信息，请参阅[使用您自己的本地名称服务器](#)。有关设置网络的更多信息，请参阅[为复制实例设置网络](#)。

使用 Amazon MSK 集群时，请确保其安全组允许从您的复制实例进行访问。有关更改 Amazon MSK 集群的安全组的信息，请参阅[更改 Amazon MSK 集群的安全组](#)。

AWS Database Migration Service 使用 JSON 将记录发布到 Kafka 主题。转换期间，AWS DMS 将每个记录从源数据库序列化到 JSON 格式的属值对。

要将数据从支持的任何数据源迁移到目标 Kafka 集群，您应使用对象映射。使用对象映射，可确定如何在目标主题中建立数据记录结构。您还可以为每个表定义分区键，Apache Kafka 使用该键将数据分组为分区。

目前，AWS DMS 支持每个任务使用一个主题。对于包含多个表的单个任务，所有消息都将转到同一个主题。每条消息都包含一个标识目标架构和表的元数据部分。AWS DMS 3.4.6 及更高版本支持使用对象映射进行多主题复制。有关更多信息，请参阅[使用对象映射进行多主题复制](#)。

## Apache Kafka 端点设置

您可以通过 AWS DMS 控制台中的端点设置或 CLI 中的 `--kafka-settings` 选项来指定连接详细信息。每个设置的要求如下：

- **Broker** – 以各个 `broker-hostname:port` 的逗号分隔列表的形式指定 Kafka 集群中一个或多个代理的位置。例  
如，`"ec2-12-345-678-901.compute-1.amazonaws.com:2345,ec2-10-987-654-321.compute-1.amazonaws.com:2345"`。  
此设置可以指定集群中任何或所有代理的位置。集群代理全部通过通信来处理迁移到主题的数据记录的分区。
- **Topic** – ( 可选 ) 指定主题名称，最大长度为 255 个字母和符号。您可以使用句点 (.)、下划线 ( \_ ) 和减号 ( - )。带句点 (.) 或下划线 ( \_ ) 的主题名称可能会在内部数据结构中发生冲突。在主题名称中使用其中一个符号，但不要同时使用这两个符号。如果您未指定主题名称，则 AWS DMS 使用 `"kafka-default-topic"` 作为迁移主题。

### Note

要 AWS DMS 创建您指定的迁移主题或默认主题，请 `auto.create.topics.enable = true` 将其设置为 Kafka 集群配置的一部分。有关更多信息，请参阅 [使用 Apache Kafka 作为目标时的限制 AWS Database Migration Service](#)。

- **MessageFormat** – 在端点上创建的记录的输出格式。消息格式为 JSON ( 默认值 ) 或 JSON\_UNFORMATTED ( 单行，无制表符 )。
- **MessageMaxBytes** – 端点上创建的记录的最大大小 ( 以字节为单位 )。默认值是 1000000。

### Note

您只能使用 AWS CLI/SDK 更改 `MessageMaxBytes` 为非默认值。例如，要修改现有的 Kafka 端点并更改 `MessageMaxBytes`，请使用以下命令。

```
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** – 提供源数据库中的详细事务信息。此信息包括提交时间戳、日志位置以及 `transaction_id`、`previous_transaction_id` 和 `transaction_record_id` ( 事务内的记录偏移 ) 的值。默认值为 `false`。

- `IncludePartitionValue` – 显示 Kafka 消息输出中的分区值，除非分区类型为 `schema-table-type`。默认值为 `false`。
- `PartitionIncludeSchemaTable` – 当分区类型为 `primary-key-type` 时，将架构和表名称作为分区值的前缀。这样做会提高数据在 Kafka 分区之间的分布广度。例如，假设 SysBench 架构具有数千个表，并且每个表的主键只有有限的范围。在这种情况下，同一主键将从数千个表发送到同一个分区，这会导致限制。默认值为 `false`。
- `IncludeTableAlterOperations` – 包括更改控制数据中表的任何数据定义语言 ( DDL ) 操作，例如 `rename-table`、`drop-table`、`add-column`、`drop-column` 和 `rename-column`。默认值为 `false`。
- `IncludeControlDetails` – 显示 Kafka 消息输出中的表定义、列定义以及表和列更改的详细控制信息。默认值为 `false`。
- `IncludeNullAndEmpty` – 在目标中包括空列。默认值为 `false`。
- `SecurityProtocol` – 使用传输层安全性协议 ( TLS ) 设置到 Kafka 目标端点的安全连接。选项包括 `ssl-authentication`、`ssl-encryption` 和 `sasl-ssl`。使用 `sasl-ssl` 需要 `SaslUsername` 和 `SaslPassword`。
- `SslEndpointIdentificationAlgorithm`— 为证书设置主机名验证。3.5.1 及更高 AWS DMS 版本支持此设置。这些选项包含以下内容：
  - `NONE`：在客户端连接中禁用代理的主机名验证。
  - `HTTPS`：在客户端连接中启用代理的主机名验证。

您可以使用设置来帮助提高传输速度。为此，AWS DMS 支持多线程完全加载到 Apache Kafka 目标集群。AWS DMS 通过包含以下选项的任务设置支持此多线程：

- `MaxFullLoadSubTasks`— 使用此选项表示要并行加载的最大源表数。AWS DMS 使用专用子任务将每个表加载到其相应的 Kafka 目标表中。默认值为 8；最大值为 49。
- `ParallelLoadThreads`— 使用此选项指定用于将每个表加载到其 Kafka 目标表中的线程数。AWS DMS Apache Kafka 目标的最大值为 32。您可以请求提高此最大值限制。
- `ParallelLoadBufferSize` – 使用此选项指定在缓冲区 ( 并行加载线程将数据加载到 Kafka 目标时使用 ) 中存储的最大记录数。默认值是 50。最大值为 1000。将此设置与 `ParallelLoadThreads` 一起使用；仅在有多个线程时 `ParallelLoadBufferSize` 才有效。
- `ParallelLoadQueuesPerThread` – 使用此选项可以指定每个并发线程访问的队列数，以便从队列中取出数据记录并为目标生成批处理负载。默认值是 1。最大值为 512。

您可以通过调整并行线程和批量操作的任务设置来提高 Kafka 端点的更改数据捕获 ( CDC ) 性能。为此，您可以使用 `ParallelApply*` 任务设置来指定并发线程的数量、每个线程的队列数以及要存储在缓冲区中的记录数。例如，假设您要执行 CDC 加载并且要并行应用 128 个线程。您还希望对于每个线程访问 64 个队列，每个缓冲区存储 50 条记录。

为了提高 CDC 性能，AWS DMS 支持以下任务设置：

- `ParallelApplyThreads`— 指定在 CDC 加载期间 AWS DMS 用于将数据记录推送到 Kafka 目标端点的并发线程数。默认值为零 ( 0 )，最大值为 32。
- `ParallelApplyBufferSize` – 指定在 CDC 加载期间要在每个缓冲区队列中存储的最大记录数，以便将并发线程推送到 Kafka 目标端点。默认值是 100，最大值是 1,000。当 `ParallelApplyThreads` 指定多个线程时，请使用此选项。
- `ParallelApplyQueuesPerThread` – 指定每个线程访问的队列数，以便从队列中取出数据记录并在 CDC 期间为 Kafka 端点生成批处理负载。默认值是 1。最大值为 512。

使用 `ParallelApply*` 任务设置时，`partition-key-type` 默认值是表的 `primary-key`，而不是 `schema-name.table-name`。

## 使用传输层安全性协议 ( TLS ) 连接到 Kafka

Kafka 集群仅接受使用传输层安全性协议 ( TLS ) 的安全连接。通过 DMS，您可以使用以下三个安全协议选项中的任何一个来保护 Kafka 端点连接。

### SSL 加密 ( **server-encryption** )

客户端通过服务器的证书验证服务器身份。然后在服务器和客户端之间建立加密连接。

### SSL 身份验证 ( **mutual-authentication** )

服务器和客户端通过自己的证书相互验证身份。然后在服务器和客户端之间建立加密连接。

### SASL-SSL ( **mutual-authentication** )

简单身份验证和安全层 ( SASL ) 方法将客户端的证书替换为用户名和密码以验证客户端身份。具体而言，您需要提供服务器已注册的用户名和密码，以便服务器可以验证客户端的身份。然后在服务器和客户端之间建立加密连接。



**⚠ Important**

Apache Kafka 和 Amazon MSK 接受已解析的证书。这是 Kafka 和 Amazon MSK 有待解决的已知限制。有关更多信息，请参阅 [Apache Kafka 问题 , KAFKA-3700](#)。

如果您使用的是 Amazon MSK，请考虑使用访问控制列表 ( ACL ) 来临时解决此已知限制。有关使用 ACL 的更多信息，请参阅《Amazon Managed Streaming for Apache Kafka 开发人员指南》中的 [Apache Kafka ACL](#) 部分。

如果您使用的是自管理 Kafka 集群，请参阅 [2018 年 10 月 21 日的评论](#)，了解有关配置集群的信息。

## 将 SSL 加密与 Amazon MSK 或自管理 Kafka 集群结合使用

您可以使用 SSL 加密来保护与 Amazon MSK 或自管理 Kafka 集群的端点连接。使用 SSL 加密身份验证方法时，客户端会通过服务器的证书验证服务器的身份。然后在服务器和客户端之间建立加密连接。

### 使用 SSL 加密连接到 Amazon MSK

- 创建目标 Kafka 端点时，使用 `ssl-encryption` 选项设置安全协议端点设置 ( `SecurityProtocol` )。

以下 JSON 示例将安全协议设置为 SSL 加密。

```
"KafkaSettings": {
  "SecurityProtocol": "ssl-encryption",
}
```

### 对自管理 Kafka 集群使用 SSL 加密

1. 如果您在本地 Kafka 集群中使用私有证书颁发机构 ( CA )，请上传您的私有 CA 证书并获取 Amazon 资源名称 ( ARN )。
2. 创建目标 Kafka 端点时，使用 `ssl-encryption` 选项设置安全协议端点设置 ( `SecurityProtocol` )。以下 JSON 示例将安全协议设置为 `ssl-encryption`。

```
"KafkaSettings": {
  "SecurityProtocol": "ssl-encryption",
}
```

```
}
```

- 如果您使用的是私有 CA，请在上面第一步中获得的 ARN 中设置 SslCaCertificateArn。

## 使用 SSL 身份验证

您可以使用 SSL 身份验证来保护与 Amazon MSK 或自管理 Kafka 集群的端点连接。

要使用 SSL 身份验证启用客户端身份验证和加密以连接到 Amazon MSK，请执行以下操作：

- 为 Kafka 准备私钥和公有证书。
- 将证书上传到 DMS 证书管理器。
- 使用 Kafka 端点设置中指定的相应证书 ARN 创建一个 Kafka 目标端点。

## 为 Amazon MSK 准备私钥和公有证书

- 创建 EC2 实例并设置客户端以使用身份验证，如《Amazon Managed Streaming for Apache Kafka 开发人员指南》中的[客户端身份验证](#)部分的步骤 1 到 9 中所述。

完成这些步骤后，您将获得证书 ARN（保存在 ACM 中的公有证书 ARN）和一个包含在 `kafka.client.keystore.jks` 文件中的私钥。

- 使用以下命令获取公有证书并将证书复制到 `signed-certificate-from-acm.pem` 文件中：

```
aws acm-pca get-certificate --certificate-authority-arn Private_CA_ARN --  
certificate-arn Certificate_ARN
```

该命令会返回类似以下示例中显示的信息。

```
{"Certificate": "123", "CertificateChain": "456"}
```

然后，将 "123" 的等效内容复制到 `signed-certificate-from-acm.pem` 文件中。

- 通过导入 `msk-rsa` 密钥从 `kafka.client.keystore.jks` to `keystore.p12` 获取私钥，如下示例所示。

```
keytool -importkeystore \  
-srckeystore kafka.client.keystore.jks \  
-destkeystore keystore.p12 \  

```

```
-deststoretype PKCS12 \  
-srcalias msk-rsa-client \  
-deststorepass test1234 \  
-destkeypass test1234
```

4. 使用以下命令将 keystore.p12 导出为 .pem 格式。

```
openssl pkcs12 -in keystore.p12 -out encrypted-private-client-key.pem -nocerts
```

出现输入 PEM 密码短语消息，其中标识出用于加密证书的密钥。

5. 从 .pem 文件中删除包属性和密钥属性，确保第一行以如下字符串开头。

```
---BEGIN ENCRYPTED PRIVATE KEY---
```

将公有证书和私钥上传到 DMS 证书管理器并测试与 Amazon MSK 的连接

1. 使用以下命令上传到 DMS 证书管理器。

```
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. 创建 Amazon MSK 目标端点并测试连接，以确保 TLS 身份验证有效。

```
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**

您可以使用 SSL 身份验证来保护与自管理 Kafka 集群的连接。在某些情况下，您可能在本地 Kafka 集群中使用私有证书颁发机构 ( CA )。如果是，请将您的 CA 链、公有证书和私钥上传到 DMS 证书管理器。然后，在创建本地 Kafka 目标端点时，在端点设置中使用相应的 Amazon 资源名称 ( ARN )。

## 为自管理 Kafka 集群准备私钥和签名的证书

1. 生成以下示例所示的密钥对。

```
keytool -genkey -keystore kafka.server.keystore.jks -validity 300 -storepass your-keystore-password
-keypass your-key-passphrase -dname "CN=your-cn-name"
-alias alias-of-key-pair -storetype pkcs12 -keyalg RSA
```

2. 生成证书签名请求 ( 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. 使用集群信任库中的 CA 签署 CSR。如果您没有 CA，则可以创建自己的私有 CA。

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days validate-days
```

4. 将 ca-cert 导入服务器信任库和密钥库。如果您没有信任库，请使用以下命令创建信任库，并将 ca-cert 导入其中。

```
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. 在证书上签名。

```
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. 将签名的证书导入密钥库。

```
keytool -keystore kafka.server.keystore.jks -import -file signed-certificate.pem -
alias on-premise-rsa -storepass your-keystore-password
-keypass your-key-password
```

7. 使用以下命令将 `on-premise-rsa` 密钥从 `kafka.server.keystore.jks` 导入 `keystore.p12`。

```
keytool -importkeystore \
-srckeystore kafka.server.keystore.jks \
-destkeystore keystore.p12 \
-deststoretype PKCS12 \
-srcalias on-premise-rsa \
-deststorepass your-truststore-password \
-destkeypass your-key-password
```

8. 使用以下命令将 `keystore.p12` 导出为 `.pem` 格式。

```
openssl pkcs12 -in keystore.p12 -out encrypted-private-server-key.pem -nocerts
```

9. 将 `encrypted-private-server-key.pem`、`signed-certificate.pem` 和 `ca-cert` 上传到 DMS 证书管理器。
10. 使用返回的 ARN 创建端点。

```
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
```

## 使用 SASL-SSL 身份验证连接到 Amazon MSK

简单身份验证和安全层 ( SASL ) 方法使用用户名和密码来验证客户端身份，并在服务器和客户端之间建立加密连接。

要使用 SASL，您需要在设置 Amazon MSK 集群时先创建一个安全的用户名和密码。有关如何为 Amazon MSK 集群设置安全用户名和密码的说明，请参阅《Amazon Managed Streaming for Apache Kafka 开发人员指南》中的[为 Amazon MSK 集群设置 SASL/SCRAM 身份验证](#)。

然后，在创建 Kafka 目标端点时，使用 `sasl-ssl` 选项设置安全协议端点设置 ( `SecurityProtocol` )。您还可以设置 `SaslUsername` 和 `SaslPassword` 选项。确保这些设置与首次设置 Amazon MSK 集群时创建的安全用户名和密码一致，如以下 JSON 示例所示。

```
"KafkaSettings": {
  "SecurityProtocol": "sasl-ssl",
  "SaslUsername": "Amazon MSK cluster secure user name",
  "SaslPassword": "Amazon MSK cluster secure password"
}
```

### Note

- 当前，仅 AWS DMS 支持公共 CA 支持的 SASL-SSL。DMS 不支持 SASL-SSL 用于配置了私有 CA 的自管理 Kafka。
- 对于 SASL-SSL 身份验证，默认情况下 AWS DMS 支持 SCRAM-SHA-512 机制。AWS DMS 3.5.0 及更高版本还支持 Plain 机制。要支持 Plain 机制，请将 `KafkaSettings` API 数据类型的 `SaslMechanism` 参数设置为 PLAIN。

## 使用之前映像查看 Apache Kafka ( 作为目标 ) 的 CDC 行的原始值

在将 CDC 更新写入数据流目标 ( 如 Kafka ) 时，可以在更新进行更改之前查看源数据库行的原始值。为此，请根据源数据库引擎提供的数据 AWS DMS 填充更新事件之前的图像。

不同的源数据库引擎为之前映像提供不同的信息量：

- 仅当列发生更改时，Oracle 才会对列提供更新。
- PostgreSQL 仅为作为主键一部分的列（已更改或未更改）提供数据。如果正在使用逻辑复制，并且为源表设置了 REPLICA IDENTITY FULL，则可以在此处获取有关写入 WAL 的行的完整前后信息。
- MySQL 通常为所有列（已更改或未更改）提供数据。

要启用之前映像以便将源数据库中的原始值添加到 AWS DMS 输出，请使用 `BeforeImageSettings` 任务设置或 `add-before-image-columns` 参数。此参数应用列转换规则。

`BeforeImageSettings` 使用从源数据库系统收集的值向每个更新操作添加一个新的 JSON 属性，如下所示。

```
"BeforeImageSettings": {
  "EnableBeforeImage": boolean,
  "FieldName": string,
  "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

#### Note

将 `BeforeImageSettings` 应用于完全加载任务及 CDC 任务（这会迁移现有数据并复制正在进行的更改）或仅应用于 CDC 任务（这仅复制数据更改）。不将 `BeforeImageSettings` 应用于仅完全加载的任务。

对于 `BeforeImageSettings` 选项，以下条件适用：

- 将 `EnableBeforeImage` 选项设置为 `true` 以启用之前映像。默认值为 `false`。
- 使用 `FieldName` 选项为新 JSON 属性指定名称。当 `EnableBeforeImage` 为 `true` 时，`FieldName` 是必填项且不能为空。
- `ColumnFilter` 选项指定要使用之前映像添加的列。要仅添加属于表主键一部分的列，请使用默认值 `pk-only`。要仅添加非 LOB 类型的列，请使用 `non-lob`。要添加具有之前映像值的任何列，请使用 `all`。

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```

## 使用之前映像转换规则

作为任务设置的替代方法，您可以使用 `add-before-image-columns` 参数，该参数应用列转换规则。使用此参数，您可以在 CDC 期间对数据流目标（如 Kafka）启用之前映像。

通过在转换规则中使用 `add-before-image-columns`，可以对之前映像结果应用更精细的控制。转换规则允许您使用对象定位器，该定位器允许您控制为规则选择的表。此外，您可以将转换规则链接在一起，这样可以将不同的规则应用于不同的表。然后，您可以操控使用其他规则生成的列。

### Note

不要在同一任务中将 `add-before-image-columns` 参数与 `BeforeImageSettings` 任务设置结合使用。而是对单个任务使用此参数或此设置，但不要同时使用这两者。

包含列的 `add-before-image-columns` 参数的 `transformation` 规则类型必须提供一个 `before-image-def` 部分。下面是一个示例。

```
{
  "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,
  }
}
```

`column-prefix` 的值附加到列名称前面，`column-prefix` 的默认值为 `BI_`。`column-suffix` 的值将附加到列名称之后，默认值为空。不要同时将 `column-prefix` 和 `column-suffix` 设置为空字符串。



为 `column-filter` 选择一个值。要仅添加属于表主键一部分的列，请选择 `pk-only`。选择 `non-lob` 以仅添加不属于 LOB 类型的列。或者，选择 `all` 以添加任何具有之前映像值的列。

### 之前映像转换规则的示例

以下示例中的转换规则在目标中添加一个名为 `BI_emp_no` 的新列。因此，像 `UPDATE employees SET emp_no = 3 WHERE emp_no = 1`；这样的语句用 1 填充 `BI_emp_no` 字段。当您将 CDC 更新写入 Amazon S3 目标时，通过 `BI_emp_no` 列能够判断哪个原始行已更新。

```
{
  "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"
      }
    }
  ]
}
```

有关使用 `add-before-image-columns` 规则操作的信息，请参阅 [转换规则和操作](#)。

## 使用 Apache Kafka 作为目标时的限制 AWS Database Migration Service

将 Apache Kafka 作为目标时存在以下限制：

- AWS DMS Kafka 目标终端节点不支持适用于 Apache Kafka 的亚马逊托管流媒体（亚马逊 MSK）的 IAM 访问控制。
- 不支持完整 LOB 模式。
- 为您的集群指定 Kafka 配置文件，其属性 AWS DMS 允许自动创建新主题。包括设置 `auto.create.topics.enable = true`。如果您使用的是 Amazon MSK，可以在创建 Kafka 集群时指定默认配置，然后将 `auto.create.topics.enable` 设置更改为 `true`。有关默认配置设置的更多信息，请参阅《Amazon Managed Streaming for Apache Kafka 开发人员指南》中的[默认 Amazon MSK 配置](#)。如果您需要修改使用 Amazon MSK 创建的现有 Kafka 集群，请运行 AWS CLI 命令 `aws kafka create-configuration` 更新您的 Kafka 配置，如以下示例所示：

```
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"
}
```

此处，`~/kafka_configuration` 是您使用所需的属性设置创建的配置文件。

如果您使用自己安装在 Amazon EC2 上的 Kafka 实例，请使用您的实例提供的选项 `auto.create.topics.enable = true` 将 Kafka 集群配置修改为允许 AWS DMS 自动创建新主题。

- AWS DMS 无论事务如何，都会将源数据库中单个记录的每次更新作为给定 Kafka 主题中的一条数据记录（消息）发布。
- AWS DMS 支持以下两种形式的分区键：
  - `SchemaName.TableName`：架构和表名称的组合。
  - `${AttributeName}`：JSON 中其中一个字段的值，或源数据库中表的主键。

- 对于 Kafka 端点，不支持 BatchApply。为 Kafka BatchApplyEnabled 目标使用“批量应用”（例如，目标元数据任务设置）可能会导致数据丢失。
- AWS DMS 不支持迁移超过 16 位数 BigInt 的数据类型的值。要解决此限制问题，您可以使用以下转换规则将 BigInt 列转换为字符串。有关转换规则的更多信息，请参阅 [转换规则和操作](#)。

```
{
  "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
  }
}
```

## 使用对象映射将数据迁移到 Kafka 主题

AWS DMS 使用表映射规则将数据从源映射到目标 Kafka 主题。要将数据映射到目标主题，您必须使用称为对象映射的表映射规则类型。您可以使用对象映射来定义源中的数据记录如何映射到发布到 Kafka 主题的数据记录。

除了具有分区键以外，Kafka 主题没有预设结构。

### Note

您不一定要使用对象映射。可以使用常规表映射进行各种转换。但是，分区键类型将遵循以下默认行为：

- 主键用作完全加载时的分区键。
- 如果未使用并行应用任务设置，`schema.table` 将用作 CDC 的分区键。
- 如果未使用并行应用任务设置，主键则用作 CDC 的分区键。

要创建对象映射规则，请将 `rule-type` 指定为 `object-mapping`。此规则指定您要使用的对象映射的类型。

规则的结构如下所示。

```
{
  "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 目前支持将 `map-record-to-record` 和 `map-record-to-document` 作为该 `rule-action` 参数的唯一有效值。这些设置会影响未作为 `exclude-columns` 属性列表一部分排除的值。`map-record-to-record` 和 `map-record-to-document` 值指定默认情况下如何 AWS DMS 处理这些记录。这些值不会以任何方式影响属性映射。

从关系数据库迁移到 Kafka 主题时使用 `map-record-to-record`。此规则类型使用关系数据库中的 `taskResourceId.schemaName.tableName` 值作为 Kafka 主题中的分区键，并为源数据库中的每个列创建一个属性。

使用 `map-record-to-record` 时请注意：

- 此设置仅影响 `exclude-columns` 列表排除的列。
- 对于每个这样的列，在目标主题中 AWS DMS 创建一个相应的属性。
- AWS DMS 无论源列是否用于属性映射，都会创建相应的属性。

了解 `map-record-to-record` 的一种方法是在操作时加以观察。对于本示例，假定您使用关系数据库表行开始处理，该行具有以下结构和数据。

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

要将此信息从名为 Test 的架构迁移到 Kafka 主题，请创建规则来将数据映射到目标主题。以下规则对此映射进行了说明。

```
{
  "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"
      }
    }
  ]
}
```

给定 Kafka 主题和分区键（在本例中为 `taskResourceId.schemaName.tableName`），下面说明了使用 Kafka 目标主题中的示例数据生成的记录格式：

```
{
  "FirstName": "Randy",
  "LastName": "Marsh",
  "StoreId": "5",
  "HomeAddress": "221B Baker Street",
  "HomePhone": "1234567890",
  "WorkAddress": "31 Spooner Street, Quahog",
  "WorkPhone": "9876543210",
  "DateOfBirth": "02/29/1988"
}
```

## 主题

- [使用属性映射调整数据结构](#)
- [使用对象映射进行多主题复制](#)
- [Apache Kafka 的消息格式](#)

### 使用属性映射调整数据结构

在使用属性映射将数据迁移到 Kafka 主题时，您可以调整数据的结构。例如，您可能希望将源中的多个字段合并到目标中的单个字段中。以下属性映射说明如何调整数据结构。

```
{
  "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}"
          }
        }
      }
    ]
  }
}
```

要为 `partition-key` 设置常量值，请指定 `partition-key` 值。例如，您可以执行此操作来强制将所有数据存储在一个分区中。以下映射说明了此方法。

```
{
  "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

表示特定表的控制记录的 `partition-key` 值为 `TaskId.SchemaName.TableName`。  
表示特定任务的控制记录的 `partition-key` 值为该记录的 `TaskId`。在对象映射中指定 `partition-key` 值不会影响控制记录的 `partition-key`。

## 使用对象映射进行多主题复制

默认情况下，AWS DMS 任务会将所有源数据迁移到以下 Kafka 主题之一：

- 如 AWS DMS 目标终端节点的“主题”字段所指定。
- 如果目标端点的主题字段未填，且 Kafka `auto.create.topics.enable` 设置设为 `true`，则由 `kafka-default-topic` 指定。

在 3.4.6 及更高版本的 AWS DMS 引擎中，您可以使用 `kafka-target-topic` 属性将每个迁移的源表映射到单独的主题。例如，下面的对象映射规则分别将源表 `Customer` 和 `Address` 迁移到 Kafka 主题 `customer_topic` 和 `address_topic`。同时，将所有其他源表（包括 Test 架构中的 `Bills` 表）AWS DMS 迁移到目标终端节点中指定的主题。

```

{
  "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"
  }
}
```

```
]
}
```

通过使用 Kafka 多主题复制，您可以使用单个复制任务将源表分组并迁移到单独的 Kafka 主题。

## Apache Kafka 的消息格式

JSON 输出只是键值对的列表。

### RecordType

记录类型可以是数据或控制。数据记录表示源中的实际行。控制记录表示流中的重要事件，例如，重新开始任务。

### 操作

对于数据记录，操作可以是 load、insert、update 或 delete。

对于控制记录，操作可以是 create-table、rename-table、drop-table、change-columns、add-column、drop-column、rename-column 或 column-type-change。

### SchemaName

记录的源架构。此字段对于控制记录可能是空的。

### TableName

记录的源表。此字段对于控制记录可能是空的。

### Timestamp

JSON 消息构建时间的时间戳。此字段采用 ISO 8601 格式。

以下 JSON 消息示例说明了包含所有其他元数据的数据类型消息。

```
{
  "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"
  }
}

```

以下 JSON 消息示例说明了控件类型消息。

```

{
  "control":{
    "table-def":{
      "columns":{
        "id":{
          "type":"WSTRING",
          "length":512,
          "nullable":false
        },
        "fname":{
          "type":"WSTRING",
          "length":255,
          "nullable":true
        },
        "lname":{
          "type":"WSTRING",
          "length":255,
          "nullable":true
        },
        "REGION":{
          "type":"WSTRING",
          "length":1000,
          "nullable":true
        }
      }
    },
  },
}

```

```
    "primary-key": [
      "id"
    ],
    "collation-name": "latin1_swedish_ci"
  }
},
"metadata": {
  "timestamp": "2019-11-21T19:14:22.223792Z",
  "record-type": "control",
  "operation": "create-table",
  "partition-key-type": "task-id",
  "schema-name": "sbtest",
  "table-name": "sbtest_t1"
}
}
```

## 使用 Amazon OpenSearch Service 集群作为 AWS Database Migration Service 的目标

您可以使用 AWS DMS 将数据迁移到 Amazon OpenSearch Service (OpenSearch Service)。OpenSearch Service 是一项托管服务，让用户能够轻松部署、运行并扩展 OpenSearch Service 集群。

在 OpenSearch Service 中，您使用索引和文档。索引是文档的集合，文档是包含标量值、数组和其他对象的 JSON 对象。OpenSearch 提供基于 JSON 的查询语言，因此您可以查询索引中的数据并检索相应的文档。

在 AWS DMS 为 OpenSearch Service 的目标端点创建索引时，它会为源端点的每个表创建一个索引。创建 OpenSearch Service 索引的开销取决于多个因素。这些因素包括所创建的索引数量、这些索引中的数据总量，以及 OpenSearch 为每个文档存储的少量元数据。

使用适合您的迁移范围的计算和存储资源配置 OpenSearch Service 集群。建议您根据要使用的复制任务考虑以下因素：

- 对于完全数据加载，请考虑您要迁移的数据总量以及传输速度。
- 对于复制持续更改，请考虑更新频率，以及您的端到端延迟要求。

此外，还应在 OpenSearch 集群上配置索引设置，并密切关注文档计数。

## 多线程完全加载任务设置

为帮助提高传输速度，AWS DMS 支持向 OpenSearch Service 目标集群进行多线程完全加载。AWS DMS 支持这种多线程，其任务设置包括以下内容：

- `MaxFullLoadSubTasks` – 使用此选项指示要并行加载的表的最大数目。DMS 使用专用的子任务将各个表加载到其对应的 OpenSearch Service 目标索引。默认值为 8；最大值为 49。
- `ParallelLoadThreads` – 使用此选项指定 AWS DMS 将各个表加载到其 OpenSearch Service 目标索引时使用的线程数。OpenSearch Service 目标的最大值为 32。您可以请求提高此最大值限制。

### Note

如果您不更改 `ParallelLoadThreads` 的默认值 (0)，AWS DMS 一次传输一个记录。此方法会在您的 OpenSearch Service 集群上施加过多的负载。确保您将此选项设置为 1 或更高。

- `ParallelLoadBufferSize` – 使用此选项指定在缓冲区（并行加载线程将数据加载到 OpenSearch Service 目标时使用）中存储的最大记录数。默认值是 50。最大值为 1,000。将此设置与 `ParallelLoadThreads` 一起使用；仅在有多个线程时 `ParallelLoadBufferSize` 才有效。

有关 DMS 如何使用多线程加载 OpenSearch Service 集群的更多信息，请参阅 AWS 博客文章 [扩展 Amazon OpenSearch Service 以进行 AWS Database Migration Service 迁移](#)。

## 多线程 CDC 加载任务设置

您可以使用任务设置来修改 `PutRecords` API 调用的行为，以提高 OpenSearch Service 目标集群的更改数据捕获 (CDC) 性能。为此，您可以使用 `ParallelApply*` 任务设置来指定并发线程的数量、每个线程的队列数以及要存储在缓冲区中的记录数。例如，假设您要执行 CDC 加载并且要并行应用 32 个线程。您还希望对于每个线程访问 64 个队列，每个缓冲区存储 50 条记录。

### Note

在 AWS DMS 版本 3.4.0 及更高版本中，支持在 CDC 到 Amazon OpenSearch Service 目标端点期间使用 `ParallelApply*` 任务设置。

要提高 CDC 性能，请 AWS DMS 支持以下任务设置：

- `ParallelApplyThreads` – 指定 AWS DMS 在 CDC 加载期间用于将数据记录推送到 OpenSearch Service 目标端点的并发线程数。默认值为零 (0)，最大值为 32。
- `ParallelApplyBufferSize` – 指定在 CDC 加载过程中，要在每个缓冲区队列中存储的、供并发线程推送到目标 OpenSearch Service 端点的最大记录数。默认值是 100，最大值是 1,000。当 `ParallelApplyThreads` 指定多个线程时，请使用此选项。
- `ParallelApplyQueuesPerThread` – 指定每个线程访问以将数据记录从队列中取出且在 CDC 期间为 OpenSearch Service 端点生成批处理负载的队列数。

使用 `ParallelApply*` 任务设置时，`partition-key-type` 默认值是表的 `primary-key`，而不是 `schema-name.table-name`。

## 从关系数据库表迁移到 OpenSearch Service 索引

AWS DMS 支持将数据迁移到 OpenSearch Service 的标量数据类型。在从 Oracle 或 MySQL 等关系数据库迁移到 OpenSearch Service 时，您可能希望重构存储此数据的方式。

AWS DMS 支持以下 OpenSearch Service 标量数据类型：

- 布尔值
- 日期
- Float
- Int
- 字符串

AWS DMS 将 Date 类型的数据转换为 String 类型。您可以指定自定义映射来解释这些日期。

AWS DMS 不支持迁移 LOB 数据类型。

## 使用 Amazon OpenSearch Service 作为 AWS Database Migration Service 的目标的先决条件

在开始使用 OpenSearch Service 数据库作为 AWS DMS 的目标之前，请确保您已创建 AWS Identity and Access Management (IAM) 角色。此角色应该允许 AWS DMS 访问目标端点的 OpenSearch Service 索引。以下 IAM policy 中显示了所需的最小访问权限集合。

```
{
```

```

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

```

您在迁移到 OpenSearch Service 时使用的角色必须具有以下权限。

```

{
  "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/*"
    }
  ]
}

```

在上述示例中，将 *region* 替换为 AWS 区域标识符、*account-id* 替换为您的 AWS 账户 ID，*domain-name* 替换为您的 Amazon OpenSearch Service 域的名称。例如，arn:aws:es:us-west-2:123456789012:domain/my-es-domain

## 使用 OpenSearch Service 作为 AWS DMS 的目标时的端点设置

您可以使用端点设置来配置 OpenSearch Service 目标，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者在 [AWS CLI](#) 中使用符合 `--elasticsearch-`



settings '{"EndpointSetting": "value", ...}' JSON 语法的 create-endpoint 命令。

下表显示了将 OpenSearch Service 作为目标时您可以使用的端点设置。

属性名称	有效值	默认值和描述
FullLoadErrorPercentage	一个大于 0 但不大于 100 的正整数。	10 – 对于完全加载任务，此属性确定任务失败之前允许的错误数阈值。例如，假设源终端节点有 1,500 行且此参数设置为 10。那么，如果在写入到目标终端节点时 AWS DMS 遇到超过 150 个错误（行数的 10%），任务将失败。
ErrorRetryDuration	大于 0 的正整数。	300 – 如果目标终端节点出现错误，AWS DMS 将重试的秒数。否则，任务将失败。

## 使用 Amazon OpenSearch Service 作为 AWS Database Migration Service 目标时的限制

将 Amazon OpenSearch Service 作为目标时存在以下限制：

- OpenSearch Service 使用动态映射（自动猜测）来确定用于已迁移数据的数据类型。
- OpenSearch Service 使用唯一的 ID 存储每个文档。以下是示例 ID。

```
"_id": "D359F8B537F1888BC71FE20B3D79EAE6674BE7ACA9B645B0279C7015F6FFF19FD"
```

每个文档 ID 为 64 字节长，因此请将其视为一项存储要求。例如，如果您从 AWS DMS 源迁移 100,000 行，生成的 OpenSearch Service 索引还需要存储空间来存储额外的 6,400,000 字节。

- 使用 OpenSearch Service，无法对主键属性进行更新。在将持续复制与更改数据捕获 (CDC) 一起使用时，该限制是非常重要的，因为这可能会导致在目标中包含不需要的数据。在 CDC 模式下，主键映射到 SHA256 值，长度为 32 个字节。这些主键被转换为人类可读的 64 字节字符串，并用作 OpenSearch Service 文档 ID。
- 如果 AWS DMS 遇到无法迁移的项目，它会将错误消息写入 Amazon CloudWatch Logs。这一行为不同于其他 AWS DMS 目标终端节点将错误写入例外表的行为。
- AWS DMS 不支持连接到启用了精细访问控制且使用主用户名和密码的 Amazon ES 集群。

- AWS DMS 不支持 OpenSearch Service 无服务器。
- OpenSearch Service 不支持将数据写入预先存在的索引。

## Amazon OpenSearch Service 的目标数据类型

AWS DMS 从异构数据库迁移数据时，此服务将数据类型从源数据库映射到称为 AWS DMS 数据类型的中间数据类型。然后，此服务将中间数据类型映射到目标数据类型。下表显示了各种 AWS DMS 数据类型及其映射到 OpenSearch Service 中的数据类型。

AWS DMS 数据类型	OpenSearch Service 数据类型
布尔值	布尔值
日期	字符串
Time	date
时间戳	date
INT4	integer
Real4	float
UINT4	integer

有关 AWS DMS 数据类型的其他信息，请参阅[AWS Database Migration Service 的数据类型](#)。

## 将 Amazon DocumentDB 作为 AWS Database Migration Service 的目标

有关 AWS DMS 支持哪些版本的 Amazon DocumentDB (与 MongoDB 兼容) 的信息，请参阅[目标 AWS DMS](#)。您可以使用 AWS DMS 将数据从 AWS DMS 支持的任一源数据引擎迁移到 Amazon DocumentDB (与 MongoDB 兼容)。该引擎可以位于 AWS 托管服务 (如 Amazon RD、Aurora 或 Amazon S3) 上。或者，该引擎可以位于自管理数据库 (如在 Amazon EC2 上或本地运行的 MongoDB) 上。

您可以使用 AWS DMS 将源数据复制到 Amazon DocumentDB 数据库、集合或文档。

**Note**

如果您的源端点是 MongoDB 或 Amazon DocumentDB，请在文档模式下运行迁移。

MongoDB 以二进制 JSON 格式 (BSON) 存储数据。AWS DMS 支持 Amazon DocumentDB 支持的所有 BSON 数据类型。有关这些数据类型的列表，请参阅《Amazon DocumentDB 开发人员指南》中的[支持的 MongoDB API、操作和数据类型](#)。

如果源端点是关系数据库，则 AWS DMS 会将数据库对象映射到 Amazon DocumentDB，如下所示：

- 关系数据库或数据库架构映射到 Amazon DocumentDB 数据库。
- 关系数据库中的表映射到 Amazon DocumentDB 中的集合。
- 关系表中的记录映射到 Amazon DocumentDB 中的文档。每个文档均由源记录中的数据构成。

如果源端点为 Amazon S3，则生成的 Amazon DocumentDB 对象对应于 Amazon S3 的 AWS DMS 映射规则。例如，考虑以下 URI。

```
s3://mybucket/hr/employee
```

在这种情况下，AWS DMS 将 mybucket 中的对象映射到 Amazon DocumentDB，如下所示：

- 顶级 URI 部分 (hr) 映射到 Amazon DocumentDB 数据库。
- 接下来的 URI 部分 (employee) 映射到 Amazon DocumentDB 集合。
- employee 中的每个对象都映射到 Amazon DocumentDB 中的一个文档。

有关 Amazon S3 的映射规则的更多信息，请参阅[使用亚马逊 S3 作为来源 AWS DMS](#)。

### Amazon DocumentDB 端点设置

在 AWS DMS 3.5.0 及更高版本中，您可以通过调整并行线程和批量操作的任务设置来提高 Amazon DocumentDB 端点的更改数据捕获 (CDC) 性能。为此，您可以使用 ParallelApply\* 任务设置来指定并发线程的数量、每个线程的队列数以及要存储在缓冲区中的记录数。例如，假设您要执行 CDC 加载并且要并行应用 128 个线程。您还希望对于每个线程访问 64 个队列，每个缓冲区存储 50 条记录。

要提高 CDC 性能，请 AWS DMS 支持以下任务设置：

- `ParallelApplyThreads` – 指定 AWS DMS 在 CDC 加载期间用于将数据记录推送到 Amazon DocumentDB 目标端点的并发线程数。默认值为零 (0)，最大值为 32。
- `ParallelApplyBufferSize` – 指定在 CDC 加载过程中，要在每个缓冲区队列中存储的、供并发线程推送到目标 Amazon DocumentDB 端点的最大记录数。默认值是 100，最大值是 1,000。当 `ParallelApplyThreads` 指定多个线程时，请使用此选项。
- `ParallelApplyQueuesPerThread` – 指定每个线程访问以将数据记录从队列中取出并在 CDC 期间为 Amazon DocumentDB 端点生成批处理负载的队列数。默认值是 1。最大值为 512。

有关将 Amazon DocumentDB 作为 AWS DMS 目标的其他详细信息，请参阅以下章节：

## 主题

- [将数据从源映射到 Amazon DocumentDB 目标](#)
- [连接到 Amazon DocumentDB 弹性集群作为目标](#)
- [以 Amazon DocumentDB 作为目标的持续复制](#)
- [使用 Amazon DocumentDB 作为目标的限制](#)
- [使用以 Amazon DocumentDB 为目标的端点设置](#)
- [Amazon DocumentDB 的目标数据类型](#)

### Note

有关迁移过程的分步演练，请参阅《AWS Database Migration Service 分步迁移指南》中的[从 MongoDB 迁移到 Amazon DocumentDB](#)。

## 将数据从源映射到 Amazon DocumentDB 目标

AWS DMS 从源终端节点读取记录，并基于它所读取的数据来构建 JSON 文档。对于每个 JSON 文档，AWS DMS 必须确定一个 `_id` 字段以充当唯一标识符。然后，它使用 `_id` 字段作为主键，将 JSON 文档写入到 Amazon DocumentDB 集合。

### 单列源数据

如果源数据包含单个列，则数据必须是字符串类型。（根据源引擎，实际数据类型可能是 VARCHAR、NVARCHAR、TEXT、LOB、CLOB 或类似类型。）AWS DMS 假定数据是有效的 JSON 文档，并按原样将数据复制到 Amazon DocumentDB。

如果生成的 JSON 文档包含名为 `_id` 的字段，则该字段用作 Amazon DocumentDB 中的唯一 `_id`。

如果 JSON 不包含 `_id` 字段，则 Amazon DocumentDB 会自动生成 `_id` 值。

## 多列源数据

如果源数据包含多个列，则 AWS DMS 根据所有这些列来构建 JSON 文档。要确定文档的 `_id` 字段，AWS DMS 按如下方式继续：

- 如果其中一个列被命名为 `_id`，则该列中的数据将用作目标 `_id`。
- 如果没有 `_id` 列，但源数据包含主键或唯一索引，则 AWS DMS 使用该键或索引值作为 `_id` 值。主键或唯一索引中的数据也在 JSON 文档中显示为显式字段。
- 如果没有 `_id` 列，且没有主键或唯一索引，则 Amazon DocumentDB 会自动生成 `_id` 值。

## 在目标终端节点强制转换数据类型

AWS DMS 可以在数据写入到 Amazon DocumentDB 目标端点时修改数据结构。可以通过在源终端节点重命名列和表，或者通过提供运行任务时所应用的转换规则来请求这些更改。

## 使用嵌套 JSON 文档 ( `json_` 前缀 )

要强制转换数据类型，可以手动或使用转换的方式在源列名称中加上前缀 `json_` ( 即 `json_columnName` )。在这种情况下，该列作为目标文档中的嵌套 JSON 文档 ( 而不是作为字符串字段 ) 而创建。

例如，假设您要从 MongoDB 源终端节点迁移以下文档。

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": "{\"Home\": {\"Address\": \"Boston\", \"Phone\": \"1111111\"}, \"Work\": { \"Address\": \"Boston\", \"Phone\": \"222222222\"}}"
```

如果不强制转换任何源数据类型，则嵌入的 `ContactDetails` 文档将作为字符串进行迁移。

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
```

```
"ContactDetails": "{\\"Home\\": {\\"Address\\": \\"Boston\\",\\"Phone\\": \\"11111111\\"},
\\"Work\\": { \\"Address\\": \\"Boston\\", \\"Phone\\": \\"2222222222\\"}}"
```

但是，可以添加一个转换规则，以将 ContactDetails 强制转换为 JSON 对象。例如，假设原始源列名称为 ContactDetails。要将数据类型强制为嵌套 JSON，需要手动或通过转换规则在源端点上添加“\*json\_\*”前缀，将源端点的列重命名为“json\_contactDetails”。例如，您可以使用以下转换规则：

```
{
  "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 将 ContactDetails 字段复制为嵌套 JSON，如下所示。

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactDetails": {
    "Home": {
      "Address": "Boston",
      "Phone": "1111111111"
    },
    "Work": {
      "Address": "Boston",
```

```
        "Phone": "2222222222"
    }
}
```

使用 JSON 数组 ( `array_ 前缀` )

要强制转换数据类型，可以手动或使用转换的方式在列名称中加上前缀 `array_` ( 即 `array_columnName` )。在这种情况下，AWS DMS 将该列视为 JSON 数组，并在目标文档中以这种方式创建它。

假设您要从 MongoDB 源终端节点迁移以下文档。

```
{
  "_id" : "1",
  "FirstName": "John",
  "LastName": "Doe",

  "ContactAddresses": ["Boston", "New York"],

  "ContactPhoneNumbers": ["1111111111", "2222222222"]
}
```

如果不强制转换任何源数据类型，则嵌入的 `ContactDetails` 文档将作为字符串进行迁移。

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",

  "ContactAddresses": "[\"Boston\", \"New York\"]",

  "ContactPhoneNumbers": "[\"1111111111\", \"2222222222\"]"
}
```

但是，您可以添加转换规则，以将 `ContactAddress` 和 `ContactPhoneNumbers` 强制转换为 JSON 数组，如下表所示。

原始源列名称	重命名的源列
ContactAddress	array_ContactAddress
ContactPhoneNumbers	array_ContactPhoneNumbers

AWS DMS 复制 ContactAddress 和 ContactPhoneNumbers，如下所示。

```
{
  "_id": "1",
  "FirstName": "John",
  "LastName": "Doe",
  "ContactAddresses": [
    "Boston",
    "New York"
  ],
  "ContactPhoneNumbers": [
    "1111111111",
    "2222222222"
  ]
}
```

## 使用 TLS 连接到 Amazon DocumentDB

默认情况下，新创建的 Amazon DocumentDB 集群仅接受使用传输层安全性协议 (TLS) 的安全连接。当 TLS 启用时，每个到 Amazon DocumentDB 的连接都需要公有密钥。

您可以通过从 AWS 托管的 Amazon S3 存储桶下载文件 `rds-combined-ca-bundle.pem` 来检索 Amazon DocumentDB 的公钥。有关下载此文件的更多信息，请参阅《Amazon DocumentDB 开发人员指南》中的[使用 TLS 加密连接](#)

下载此 .pem 文件后，您可以将该文件包含的公有密钥导入到 AWS DMS 中，如下所述。

## AWS Management Console

### 导入公有密钥 (.pem) 文件

1. 打开 AWS DMS 控制台 (<https://console.aws.amazon.com/dms>)。
2. 在导航窗格中，选择证书。



### 3. 选择导入证书，然后执行以下操作：

- 对于 Certificate identifier (证书标识符)，请输入证书的唯一名称，例如 docdb-cert。
- 对于 Import file (导入文件)，请导航到要将 .pem 文件保存到的位置。

根据需要设置完毕后，选择 Add new CA certificate (添加新的 CA 证书)。

## AWS CLI

使用 `aws dms import-certificate` 命令，如以下示例所示。

```
aws dms import-certificate \  
  --certificate-identifier docdb-cert \  
  --certificate-pem file:///./rds-combined-ca-bundle.pem
```

创建 AWS DMS 目标终端节点时，请提供证书标识符（例如 docdb-cert）。另外，请将 SSL 模式参数设置为 `verify-full`。

## 连接到 Amazon DocumentDB 弹性集群作为目标

在 AWS DMS 版本 3.4.7 及更高版本中，您可以创建 Amazon DocumentDB 目标端点作为弹性集群。如果您将目标端点创建为弹性集群，则需要将新的 SSL 证书附加到您的 Amazon DocumentDB 弹性集群节点，因为您的现有 SSL 证书无法使用。

将新的 SSL 证书附加到您的 Amazon DocumentDB 弹性集群端点

1. 例如，在浏览器中打开 <https://www.amazontrust.com/repository/SFSRootCAG2.pem> 并将内容保存到具有唯一文件名的 .pem 文件中，例如 SFSRootCAG2.pem。这是您需要在后续步骤中导入的证书文件。
2. 创建弹性集群端点并设置以下选项：
  - a. 在端点配置下，选择添加新 CA 证书。
  - b. 对于证书标识符，输入 **SFSRootCAG2.pem**。
  - c. 对于导入证书文件，选择选择文件，然后导航到之前下载的 SFSRootCAG2.pem 文件。
  - d. 选择并打开下载的 SFSRootCAG2.pem 文件。
  - e. 选择导入证书。
  - f. 从选择证书下拉列表中，选择 SFSRootCAG2.pem。

下载的 SFSRootCAG2.pem 文件中的新 SSL 证书现已附加到您的 Amazon DocumentDB 弹性集群端点。

## 以 Amazon DocumentDB 作为目标的持续复制

如果以 Amazon DocumentDB 为目标启用持续复制（变更数据捕获，CDC），则 AWS DMS 版本 3.5.0 及更高版本提供的性能提升将是之前版本的 20 倍。在之前的版本中，AWS DMS 每秒最多可处理 250 条记录，AWS DMS 现在每秒处理大约 5000 条记录。AWS DMS 还可确保 Amazon DocumentDB 中的文档与源文件保持同步。创建或更新源记录时，AWS DMS 必须首先通过执行以下操作来确定哪个 Amazon DocumentDB 记录受到影响：

- 如果源记录有一个名称为 `_id` 的列，则该列的值将确定 Amazon DocumentDB 集合中的相应 `_id`。
- 如果没有 `_id` 列，但源数据有主键或唯一索引，则 AWS DMS 使用该键或索引值作为 Amazon DocumentDB 集合的 `_id`。
- 如果源记录没有 `_id` 列、主键或唯一索引，则 AWS DMS 会将所有源列与 Amazon DocumentDB 集合中的相应字段进行匹配。

创建新的源记录后，AWS DMS 会将相应文档写入 Amazon DocumentDB。如果更新了现有的源记录，则 AWS DMS 会更新 Amazon DocumentDB 中的目标文档中的相应字段。目标文档中存在而源记录中不存在的任何字段均保持不变。

删除源记录后，AWS DMS 会从 Amazon DocumentDB 中删除相应文档。

### 源位置的结构变更 (DDL)

通过持续复制，对源数据结构（如表、列等）的任何更改都将传播到其在 Amazon DocumentDB 中的对应项。在关系数据库中，通过使用数据定义语言 (DDL) 语句启动这些更改。可以在下表中查看 AWS DMS 如何将这些更改传播到 Amazon DocumentDB。

源位置的 DDL	对 Amazon DocumentDB 目标的影响
CREATE TABLE	创建一个空集合。
重命名表的语句 (RENAME TABLE、ALTER TABLE...RENAME 及类似语句)	重命名集合。
TRUNCATE TABLE	从集合中删除所有文档，但前提是 <code>HandleSourceTableTruncated</code> 为 <code>true</code> 。有关更多信息，请参阅 <a href="#">更改处理 DDL 处理的任务设置</a> 。

源位置的 DDL	对 Amazon DocumentDB 目标的影响
DROP TABLE	删除集合，但前提是 HandleSourceTableDropped 为 true。有关更多信息，请参阅 <a href="#">更改处理 DDL 处理的任务设置</a> 。
向表中添加一列的语句 ( ALTER TABLE...ADD 及类似语句 )	将忽略 DDL 语句，并发出警告。在源位置执行第一个 INSERT 时，新字段将添加到目标文档。
ALTER TABLE...RENAME COLUMN	将忽略 DDL 语句，并发出警告。在源位置执行第一个 INSERT 时，新命名的字段将添加到目标文档。
ALTER TABLE...DROP COLUMN	将忽略 DDL 语句，并发出警告。
更改列数据类型的语句 ( ALTER COLUMN...MODIFY 及类似语句 )	将忽略 DDL 语句，并发出警告。使用新数据类型在源位置执行第一个 INSERT 时，将创建包含该新数据类型的字段的文档。

## 使用 Amazon DocumentDB 作为目标的限制

将 Amazon DocumentDB 作为 AWS DMS 的目标时存在以下限制：

- 在 Amazon DocumentDB 中，集合名称不能包含美元符号 (\$)。此外，数据库名称不能包含任何 Unicode 字符。
- AWS DMS 不支持将多个源表合并到单个 Amazon DocumentDB 集合。
- 当 AWS DMS 处理没有主键的源表中的更改时，将忽略该表中的任何 LOB 列。
- 如果 Change table (更改表) 选项处于启用状态，并且 AWS DMS 发现名为“\_id”的源列，则该列将在更改表中显示为“\_\_id” ( 两条下划线 )。
- 如果选择 Oracle 作为源终端节点，则 Oracle 源必须启用完整补充日志记录。否则，如果源位置有未更改的列，则数据将作为空值加载到 Amazon DocumentDB。
- 不支持将复制任务设置 TargetTablePrepMode:TRUNCATE\_BEFORE\_LOAD 与 DocumentDB 目标端点一起使用。

## 使用以 Amazon DocumentDB 为目标的端点设置

您可以使用端点设置来配置 Amazon DocumentDB 目标，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定设置，或者在 [AWS CLI](#) 中使用符合 `--doc-db-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令。

下表显示了在将 Amazon DocumentDB 作为目标时可以使用的端点设置。

属性名称	有效值	默认值和描述
replicate ShardColl ections	布尔值  true  false	在 true 时，此端点设置会产生以下效果并施加以下限制： <ul style="list-style-type: none"> <li>• 允许 AWS DMS 将数据复制到目标分片集合。仅当目标 DocumentDB 端点是弹性集群时，此设置才适用。</li> <li>• 必须将 TargetTablePrepMode 设置为 DO_NOTHING。</li> <li>• AWS DMS 在迁移期间自动将 useUpdateLookup 设置为 false。</li> </ul>

## Amazon DocumentDB 的目标数据类型

在下表中，您可以找到使用 AWS DMS 时支持的 Amazon DocumentDB 目标数据类型以及来自 AWS DMS 数据类型的默认映射。有关 AWS DMS 数据类型的更多信息，请参阅 [AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	Amazon DocumentDB 数据类型
BOOLEAN	布尔值
BYTES	二进制数据
DATE	日期
TIME	字符串 (UTF8)
DATETIME	日期

AWS DMS 数据类型	Amazon DocumentDB 数据类型
INT1	32 位整数
INT2	32 位整数
INT4	32 位整数
INT8	64 位整数
NUMERIC	字符串 (UTF8)
REAL4	Double
REAL8	Double
STRING	如果数据被识别为 JSON，则 AWS DMS 会将其作为文档迁移到 Amazon DocumentDB。否则，数据将映射到“字符串 (UTF8)”。
UINT1	32 位整数
UINT2	32 位整数
UINT4	64 位整数
UINT8	字符串 (UTF8)
WSTRING	如果数据被识别为 JSON，则 AWS DMS 会将其作为文档迁移到 Amazon DocumentDB。否则，数据将映射到“字符串 (UTF8)”。
BLOB	二进制
CLOB	如果数据被识别为 JSON，则 AWS DMS 会将其作为文档迁移到 Amazon DocumentDB。否则，数据将映射到“字符串 (UTF8)”。
NCLOB	如果数据被识别为 JSON，则 AWS DMS 会将其作为文档迁移到 Amazon DocumentDB。否则，数据将映射到“字符串 (UTF8)”。

## 将 Amazon Neptune 作为 AWS Database Migration Service 的目标

Amazon Neptune 是一项快速、可靠且完全托管式的图数据库服务，可帮助您轻松构建和运行适用于高度互连数据集的应用程序。Neptune 的核心是一个专门打造的高性能图形数据库引擎。该引擎经过优化，可存储数十亿个关系并能以毫秒级延迟进行图形查询。Neptune 支持常见的图形查询语言 Apache TinkerPop Gremlin 和 W3C SPARQL。有关 Amazon Neptune 的更多信息，请参阅《Amazon Neptune 用户指南》中的 [Amazon Neptune 是什么？](#)。

如果没有图形数据库（如 Neptune），您可以在关系数据库中对高度互连的数据进行建模。由于数据具有潜在的动态关联，因此使用此类数据源的应用程序必须在 SQL 中为关联的数据查询建模。此方法要求您编写额外的层以将图形查询转换为 SQL。此外，关系数据库附带了架构刚度。为对更改连接建模而在架构中进行任何更改都需要停机时间以及对查询转换的额外维护，以支持新架构。查询性能也是设计应用程序时需要考虑的另一大约束。

图形数据库可大大简化此类情况。丰富的图形查询层（Gremlin 或 SPARQL）和针对图形查询优化的索引提高了灵活性和性能，可将您从架构中解放出来。Amazon Neptune 图形数据库还具有企业功能，如静态加密、安全授权层、默认备份、多可用区支持、只读副本支持等。

利用 AWS DMS，您可以将用于对高度互连的图形进行建模的关系数据从任何受支持的 SQL 数据库的 DMS 源端点迁移到 Neptune 目标端点。

有关更多详细信息，请参阅以下内容。

### 主题

- [将 Amazon Neptune 作为迁移目标的概述](#)
- [指定将 Amazon Neptune 作为目标的端点设置](#)
- [创建 IAM 服务角色以便将 Amazon Neptune 作为目标进行访问](#)
- [将 Amazon Neptune 作为目标使用时使用 Gremlin 和 R2RML 指定图形映射规则](#)
- [用于将 Gremlin 和 R2RML 迁移到作为目标的 Amazon Neptune 的数据类型](#)
- [将 Amazon Neptune 作为目标的限制](#)

## 将 Amazon Neptune 作为迁移目标的概述

在开始执行到 Neptune 目标的迁移之前，请在 AWS 账户中创建以下资源：

- 目标端点的 Neptune 集群。

- 源端点的 AWS DMS 支持的 SQL 关系数据库。
- 目标端点的 Amazon S3 存储桶。在您的 Neptune 集群所在的同一 AWS 区域中创建此 S3 存储桶。AWS DMS 将此 S3 存储桶作为中间文件存储，以便存储它批量加载到 Neptune 数据库的目标数据。有关创建 S3 存储桶的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的[创建存储桶](#)。
- Neptune 集群所在的同一 VPC 中 S3 的虚拟私有云 (VPC) 端点。
- 包含 IAM policy 的 AWS Identity and Access Management (IAM) 角色。此策略将指定目标终端节点的 S3 存储桶的 GetObject、PutObject、DeleteObject 和 ListObject 权限。此角色由 AWS DMS 和 Neptune 代入，二者具有对目标 S3 存储桶和 Neptune 数据库的 IAM 访问权限。有关更多信息，请参阅[创建 IAM 服务角色以便将 Amazon Neptune 作为目标进行访问](#)。

在获得这些资源后，设置和启动到 Neptune 目标的迁移类似于使用控制台或 DMS API 进行的任何完全加载迁移。但是，到 Neptune 目标的迁移需要一些独特的步骤。

### 将 AWS DMS 关系数据库迁移到 Neptune

1. 创建复制实例，如[创建复制实例](#)中所述。
2. 创建和测试源终端节点的 AWS DMS 支持的 SQL 关系数据库。
3. 为 Neptune 数据库创建和测试目标端点。

要将目标端点连接到 Neptune 数据库，请为 Neptune 集群端点或 Neptune 写入器实例端点指定服务器名称。此外，指定 AWS DMS 的 S3 存储桶文件夹来存储其中间文件，以便批量加载到 Neptune 数据库。

迁移过程中，AWS DMS 将所有已迁移的目标数据存储在此 S3 存储桶文件夹中，直至达到您指定的最大文件大小。当此文件存储达到该最大大小时，AWS DMS 会将存储的 S3 数据批量加载到目标数据库中。它将清空此文件夹来支持存储任何其他目标数据，以便随后将该数据加载到目标数据库。有关指定这些设置的更多信息，请参阅[指定将 Amazon Neptune 作为目标的端点设置](#)。

4. 使用步骤 1-3 中创建的资源创建完全加载复制任务，然后执行以下操作：
  - a. 像往常一样使用任务表映射来标识特定的源架构、表和视图，以便使用适当的选择和转换规则从关系数据库中迁移。有关更多信息，请参阅[使用表映射指定任务设置](#)。
  - b. 通过选择下列选项之一指定源表和视图到 Neptune 目标数据库图形的映射规则，来指定目标映射：
    - Gremlin JSON – 有关使用 Gremlin JSON 加载 Neptune 数据库的信息，请参阅《Amazon Neptune 用户指南》中的[Gremlin 加载数据格式](#)。

- SPARQL RDB 到资源描述框架映射语言 (R2RML) – 有关使用 SPARQL R2RML 的信息，请参阅 W3C 规范 [R2RML : RDB 到 RDF 映射语言](#)。
- c. 请执行下列操作之一：
- 使用 AWS DMS 控制台，使用创建数据库迁移任务页面上的图形映射规则指定图形映射选项。
  - 借助 AWS DMS API，使用 CreateReplicationTask API 调用的 TaskData 请求参数指定这些选项。

有关使用 Gremlin JSON 和 SPARQL R2RML 指定图形映射规则的更多信息和示例，请参阅 [将 Amazon Neptune 作为目标使用时使用 Gremlin 和 R2RML 指定图形映射规则](#)。

## 5. 开始复制迁移任务。

### 指定将 Amazon Neptune 作为目标的端点设置

要创建或修改目标终端节点，您可以使用控制台，也可以使用 CreateEndpoint 或 ModifyEndpoint API 操作。

对于 AWS DMS 控制台中的 Neptune 目标，请在创建端点或修改端点控制台页面上指定特定于端点的设置。对于 CreateEndpoint 和 ModifyEndpoint，指定 NeptuneSettings 选项的请求参数。以下示例说明如何使用 CLI 执行此操作。

```
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}'
```

在此处，CLI --server-name 选项指定 Neptune 集群写入器端点的服务器名称。也可以指定 Neptune 写入器实例端点的服务器名称。

--neptune-settings 选项请求参数如下所示：



- `ServiceAccessRoleArn` – (必需) 您为 Neptune 目标端点创建的服务角色的 Amazon 资源名称 (ARN)。有关更多信息，请参阅[创建 IAM 服务角色以便将 Amazon Neptune 作为目标进行访问](#)。
- `S3BucketName` – (必需) S3 存储桶的名称，在该存储桶中，在将迁移的图形数据批量加载到 Neptune 目标数据库之前，DMS 可以在 .csv 文件中临时存储这些图形数据。DMS 先将 SQL 源数据映射到图形数据，然后再将其存储在这些 .csv 文件中。
- `S3BucketFolder` – (必需) 您希望 DMS 将迁移的图形数据存储在与 `S3BucketName` 指定的 S3 存储桶中的文件夹路径。
- `ErrorRetryDuration` – (可选) 在引发错误之前，DMS 等待重试将已迁移的图形数据批量加载到 Neptune 目标数据库的毫秒数。默认值是 250。
- `MaxFileSize` – (可选) DMS 将数据批量加载到 Neptune 目标数据库之前，存储在 .csv 文件中的已迁移图形数据的最大大小 (以 KB 为单位)。默认值是 1048576 KB (1 GB)。如果成功，DMS 会清除存储桶，准备存储下一批迁移的图形数据。
- `MaxRetryCount` – (可选) 在引发错误之前，DMS 重试将已迁移的图形数据批量加载到 Neptune 目标数据库的次数。默认值是 5。
- `IAMAuthEnabled` – (可选) 如果要为此端点启用 IAM 授权，请将此参数设置为 `true`，并将相应的 IAM policy 文档附加到由 `ServiceAccessRoleArn` 指定的服务角色。默认值为 `false`。

## 创建 IAM 服务角色以便将 Amazon Neptune 作为目标进行访问

要将 Neptune 作为目标进行访问，请使用 IAM 创建服务角色。根据您的 Neptune 端点配置，将下列部分或所有 IAM policy 和信任文档附加到此角色。在创建 Neptune 端点时，需提供此服务角色的 ARN。这样一来，AWS DMS 和 Amazon Neptune 便能代入权限以访问 Neptune 及其关联的 Amazon S3 存储桶。

如果您在 Neptune 端点配置中，将 `NeptuneSettings` 中的 `IAMAuthEnabled` 参数设置为 `true`，请将如下所示的 IAM policy 附加到服务角色。如果将 `IAMAuthEnabled` 设置为 `false`，则可忽略此策略。

```
// 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/*"
    }
]
}

```

前面的 IAM policy 允许对 Resource 指定的 Neptune 目标集群进行完全访问。

将如下所示的 IAM 策略附加到服务角色。此策略允许 DMS 将迁移的图形数据临时存储在创建的 S3 存储桶中，该存储桶用于将数据批量加载到 Neptune 目标数据库中。

```

//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/"
    ]
  }
]

```

```
}  
]  
}
```

前面的 IAM policy 允许您的账户查询为 Neptune 目标创建的 S3 存储桶 (arn:aws:s3:::my-bucket) 的内容。它还允许您的账户完全操作所有存储桶文件和文件夹的内容 (arn:aws:s3:::my-bucket/)

编辑信任关系并将以下 IAM 角色附加到服务角色，以允许 AWS DMS 和 Amazon Neptune 数据库服务代入该角色。

```
{  
  "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"  
    }  
  ]  
}
```

有关为 Neptune 目标端点指定此服务角色的信息，请参阅[指定将 Amazon Neptune 作为目标的端点设置](#)。

## 将 Amazon Neptune 作为目标使用时使用 Gremlin 和 R2RML 指定图形映射规则

您创建的图形映射规则指定如何将从 SQL 关系数据库源中提取的数据加载到 Neptune 数据库集群目标中。这些映射规则的格式各不相同，具体取决于规则是用于通过 Apache TinkerPop Gremlin 加载属性图形数据，还是用于通过 R2RML 加载资源描述框架 (RDF) 数据。在下文中，您可以找到有关这些格式的信息并查明可从何处了解更多信息。

在使用控制台或 DMS API 创建迁移任务时，可以指定这些映射规则。

使用控制台，使用创建数据库迁移任务页面上的图形映射规则指定这些映射规则。在图形映射规则中，可以使用提供的编辑器直接输入和编辑映射规则。或者，可以浏览找到包含采用适当的图形映射格式的映射规则的文件。

借助 API，使用 CreateReplicationTask API 调用的 TaskData 请求参数指定这些选项。将 TaskData 设置为包含采用适当图形映射格式的映射规则的文件的路径。

### 用于通过 Gremlin 生成属性图形数据的图形映射规则

使用 Gremlin 生成属性图形数据，为要从源数据生成的每个图形实体指定一个具有映射规则的 JSON 对象。此 JSON 的格式是专门为批量加载 Amazon Neptune 定义的。以下模板显示了此对象中的各个规则的情况。

```
{
  "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": [
        {
```



```

        "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"
                }
            ]
        }
    ]
}
]
}

```

在此处，顶点和边定义映射来自具有员工 ID (EmpID) 的 employee 节点和具有经理 ID (managerId) 的 employee 节点的报告关系。

有关使用 Gremlin JSON 创建图形映射规则的更多信息，请参阅《Amazon Neptune 用户指南》中的 [Gremlin 加载数据格式](#)。

### 用于生成 RDF/SPARQL 数据的图形映射规则

如果要加载将使用 SPARQL 查询的 RDF 数据，请使用 R2RML 编写图形映射规则。R2RML 是用于将关系数据映射到 RDF 的标准 W3C 语言。在 R2RML 文件中，三重映射（例如，下面的 `<#TriplesMap1>`）指定用于将逻辑表的每个行转换为零个或多个 RDF 三元组的规则。主题映射（例如，下面的任意 `rr:subjectMap`）指定了一个规则，用于生成由三元组映射生成的 RDF 三元组的主题。谓词-对象映射（例如，下面的任意 `rr:predicateObjectMap`）是一个函数，用于为逻辑表的每个逻辑表行创建一个或多个谓词-对象对。

下面是 nodes 表的简单示例。

```
@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" ];
  ]
```

在上一个示例中，映射定义了从员工表映射的图形节点。

下面是 Student 表的另一个简单示例。

```
@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" ]
  ].
```

在上一个示例中，映射定义 Student 表中人员之间的图形节点映射朋友关系。

有关使用 SPARQL R2RML 创建图形映射规则的更多信息，请参阅 W3C 规范 [R2RML : RDB 到 RDF 映射语言](#)。

## 用于将 Gremlin 和 R2RML 迁移到作为目标的 Amazon Neptune 的数据类型

AWS DMS 通过两种方式之一执行从 SQL 源端到 Neptune 目标的数据类型映射。所使用的方式取决于您用来加载 Neptune 数据库的图形映射格式：

- Apache TinkerPop Gremlin，使用迁移数据的 JSON 表示形式。
- W3C 的 SPARQL，使用迁移数据的 R2RML 表示形式。

有关这两种图形映射格式的更多信息，请参阅[将 Amazon Neptune 作为目标使用时使用 Gremlin 和 R2RML 指定图形映射规则](#)。

在下文中，您可以找到每种格式的数据类型映射的描述。

### SQL 源到 Gremlin 目标数据类型映射

下表显示了从 SQL 源到 Gremlin 格式化目标的数据类型映射。

AWS DMS 将任何未列出的 SQL 源数据类型映射到 Gremlin String。

SQL 源数据类型	Gremlin 目标数据类型
NUMERIC ( 和变体 )	Double



SQL 源数据类型	Gremlin 目标数据类型
DECIMAL	
TINYINT	Byte
SMALLINT	Short
INT, INTEGER	Int
BIGINT	Long
FLOAT	Float
DOUBLE PRECISION	
REAL	Double
BIT	Boolean
BOOLEAN	
DATE	Date
TIME	
TIMESTAMP	
CHARACTER ( 和变体 )	String

有关用于加载 Neptune 的 Gremlin 数据类型的更多信息，请参阅《Neptune 用户指南》中的 [Gremlin 数据类型](#)。

### SQL 源到 R2RML (RDF) 目标数据类型映射

下表显示了从 SQL 源到 R2RML 格式化目标的数据类型映射。

所有列出的 RDF 数据类型都区分大小写，但 RDF 文本除外。AWS DMS 将任何未列出的 SQL 源数据类型映射到 RDF 文本。

RDF 文本 是多种文本词汇形式和数据类型之一。有关更多信息，请参阅 W3C 规范《资源描述框架 (RDF)：概念和抽象语法》中的 [RDF 文字](#)。

SQL 源数据类型	R2RML (RDF) 目标数据类型
BINARY ( 和变体 )	xsd:hexBinary
NUMERIC ( 和变体 )	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 ( 和变体 )	RDF 文本

有关用于加载 Neptune 的 RDF 数据类型及其与 SQL 源数据类型的映射的更多信息，请参阅 W3C 规范《R2RML：RDB 到 RDF 映射语言》中的 [数据类型转换](#)。

## 将 Amazon Neptune 作为目标的限制

将 Neptune 作为目标时存在以下限制：

- AWS DMS 当前仅支持迁移到 Neptune 目标的完全加载任务。不支持将更改数据捕获 (CDC) 迁移到 Neptune 目标。
- 在开始迁移任务之前，请确保已手动清除目标 Neptune 数据库中的所有数据，如以下示例所示。

要删除图形中的所有数据（顶点和边），请运行以下 Gremlin 命令。

```
gremlin> g.V().drop().iterate()
```

要删除具有标签 'customer' 的顶点，请运行以下 Gremlin 命令。

```
gremlin> g.V().hasLabel('customer').drop()
```

### Note

删除大型数据集可能需要花费一些时间。您可能需要对 `drop()` 进行迭代并施加一个限制，例如 `limit(1000)`。

要删除具有标签 'rated' 的边，请运行以下 Gremlin 命令。

```
gremlin> g.E().hasLabel('rated').drop()
```

### Note

删除大型数据集可能需要花费一些时间。您可能需要对 `drop()` 进行迭代并施加一个限制，例如 `limit(1000)`。

- 由于 Neptune 图形数据结构的性质，DMS API 操作 `DescribeTableStatistics` 可能会返回有关给定表的不准确结果。

在迁移过程中，AWS DMS 会扫描每个源表，并使用图形映射将源数据转换为 Neptune 图形。转换后的数据首先会存储在为目标终端节点指定的 S3 存储桶文件夹中。如果已扫描源并成功生成此中间 S3 数据，则 `DescribeTableStatistics` 会假定已成功将数据加载到 Neptune 目标数据库中。

但情况并非总是如此。要是否已为给定表正确加载数据，请比较该表的迁移两端的 `count()` 返回值。

在以下示例中，AWS DMS 已从源数据库加载 `customer` 表，该表在目标 Neptune 数据库图形中分配有标签 `'customer'`。您可以确保此标签已写入到目标数据库中。为此，请将源数据库中可用的 `customer` 行数与任务完成后在 Neptune 目标数据库中加载的带 `'customer'` 标签的行数进行比较。

要使用 SQL 获取源数据库中可用的客户行的数量，请运行以下命令。

```
select count(*) from customer;
```

要使用 Gremlin 获取加载到目标数据库图形中的带 `'customer'` 标签的行数，请运行以下命令。

```
gremlin> g.V().hasLabel('customer').count()
```

- 目前，如果任一表无法加载，则整个任务将失败。与关系数据库目标不同，Neptune 中的数据是高度互连的，这导致在许多情况下无法恢复任务。如果因无法加载此类型的数据而导致无法成功恢复任务，请创建一个新任务来加载未能加载的表。在运行此新任务之前，请从 Neptune 目标中手动清除部分已加载的表。

#### Note

如果故障可恢复（例如，网络传输错误），则可以恢复已失败的迁移到 Neptune 目标的任务。

- AWS DMS 支持大多数适用于 R2RML 的标准。但是，AWS DMS 不支持某些 R2RML 标准，包括反向表达式、联接和视图。R2RML 视图的解决方法是，在源数据库中创建相应的自定义 SQL 视图。在迁移任务中，使用表映射选择视图作为输入。然后将视图映射到一个表，R2RML 随后会使用该表来生成图形数据。
- 在使用不受支持的 SQL 数据类型迁移源数据时，生成的目标数据可能会丢失精确性。有关更多信息，请参阅[用于将 Gremlin 和 R2RML 迁移到作为目标的 Amazon Neptune 的数据类型](#)。
- AWS DMS 不支持将 LOB 数据迁移到 Neptune 目标中。

## 将 Redis 作为 AWS Database Migration Service 的目标

Redis 是一个开源的内存数据结构存储，用作数据库、缓存和消息代理。在内存中管理数据可能导致读取或写入操作花费不到一毫秒的时间，每秒可执行数亿次操作。作为内存数据存储，Redis 支持需要亚毫秒响应时间的最高要求应用程序。

使用 AWS DMS，您可以将数据从任何支持的源数据库迁移到目标 Redis 数据存储，并且可以最大限度地减少停机时间。有关 Redis 的更多信息，请参阅 [Redis 文档](#)。

除本地 Redis 之外，AWS Database Migration Service 还支持以下内容：

- [Amazon ElastiCache for Redis](#) 作为目标数据存储。ElastiCache for Redis 可与您的 Redis 客户端配合使用，并使用开放的 Redis 数据格式来存储数据。
- [Amazon MemoryDB for Redis](#) 作为目标数据存储。MemoryDB 与 Redis 兼容，使您能够使用当前在用的所有 Redis 数据结构、API 和命令来构建应用程序。

有关将 Redis 作为 AWS DMS 目标的其他信息，请参阅以下章节：

### 主题

- [将 Redis 集群作为 AWS DMS 目标的先决条件](#)
- [将 Redis 作为 AWS Database Migration Service 目标的限制](#)
- [将数据从关系或非关系数据库迁移到 Redis 目标](#)
- [指定 Redis 作为目标时的端点设置](#)

## 将 Redis 集群作为 AWS DMS 目标的先决条件

DMS 支持独立配置中的本地 Redis 目标，也支持在多个节点之间自动分片数据的 Redis 集群。分片是将数据分成小块（称为分片）的过程，这些小块分布在多个服务器或节点上。实际上，分片是一个数据分区，它包含总数据集的子集，并作为总工作负载的一部分。

由于 Redis 是键值 NoSQL 数据存储，因此当您的源是关系数据库时，使用的 Redis 键命名约定是 `schema-name.table-name.primary-key`。在 Redis 中，键和值不得包含特殊字符 `%`。否则，DMS 会跳过记录。

**Note**

如果您使用 ElastiCache for Redis 作为目标，则 DMS 仅支持启用集群模式配置。有关使用 ElastiCache for Redis 版本 6.x 或更高版本创建启用集群模式的目标数据存储的更多信息，请参阅《Amazon ElastiCache for Redis 用户指南》中的[入门](#)。

在开始数据库迁移之前，请使用以下条件启动 Redis 集群。

- 集群拥有一个或多个分片。
- 如果您使用 ElastiCache for Redis 目标，请确保您的集群不使用基于 IAM 角色的访问控制。而是使用 Redis Auth 对用户进行身份验证。
- 启用多可用区。
- 确保集群具有足够的可用内存以容纳要从数据库迁移的数据。
- 在开始初始迁移任务之前，请确保您的目标 Redis 集群中已清除所有数据。

在创建集群配置之前，应该确定数据迁移的安全要求。无论目标复制组的加密配置如何，DMS 都支持迁移到目标复制组。但是，只有在创建集群配置时才能启用或禁用加密。

## 将 Redis 作为 AWS Database Migration Service 目标的限制

将 Redis 作为目标时存在以下限制：

- 由于 Redis 是键值 no-sql 数据存储，因此当您的源是关系数据库时，使用的 Redis 键命名约定是 `schema-name.table-name.primary-key`。
- 在 Redis 中，键值不能包含特殊字符 `%`。否则，DMS 会跳过记录。
- DMS 不会迁移包含特殊字符的行。
- DMS 不会迁移字段名称中包含特殊字符的字段。
- 不支持完整 LOB 模式。
- 使用 ElastiCache for Redis 作为目标时，不支持私有证书颁发机构 (CA)。

## 将数据从关系或非关系数据库迁移到 Redis 目标

您可以将数据从任何源 SQL 或 NoSQL 数据存储直接迁移到 Redis 目标。设置和开始到 Redis 目标的迁移类似于使用 DMS 控制台或 API 进行的任何完全加载和更改数据捕获迁移。要将数据库迁移到 Redis 目标，请执行以下操作。

- 创建复制实例以执行迁移的所有过程。有关更多信息，请参阅[创建复制实例](#)。
- 指定源端点。有关更多信息，请参阅[创建源和目标终端节点](#)。
- 找到集群的 DNS 名称和端口号。
- 下载可用来验证 SSL 连接的证书捆绑包。
- 指定目标端点，如下所述。
- 创建一个任务或一组任务来定义要使用的表和复制过程。有关更多信息，请参阅[创建任务](#)。
- 将数据从源数据库迁移到目标集群。

您可以通过以下两种方式之一开始数据库迁移：

1. 您可以选择 AWS DMS 控制台并在其中执行每个步骤。
2. 您可以使用 AWS Command Line Interface (AWS CLI)。有关使用适用于 AWS DMS 的 CLI 的更多信息，请参阅[适用于 AWS DMS 的 AWS CLI](#)。

### 找到集群的 DNS 名称和端口号

- 使用以下 AWS CLI 命令为 replication-group-id 提供复制组的名称。

```
aws elasticache describe-replication-groups --replication-group-id myreplgroup
```

此处，输出在集群中主节点的 Address 属性中显示 DNS 名称，在 Port 属性中显示端口号。

```
...
"ReadEndpoint": {
  "Port": 6379,
  "Address": "myreplgroup-
111.1abc1d.1111.uuu1.cache.example.com"
}
...
```

如果您使用 MemoryDB for Redis 作为目标，请使用以下 AWS CLI 命令为您的 Redis 集群提供端点地址。

```
aws memorydb describe-clusters --clusterid clusterid
```

下载可用于验证 SSL 连接的证书捆绑包。

- 在命令行输入以下 `wget` 命令。Wget 是一个免费的 GNU 命令行实用工具，用于从互联网下载文件。

```
wget https://s3.aws-api-domain/rds-downloads/rds-combined-ca-bundle.pem
```

在这里，`aws-api-domain` 在您的 AWS 所在区域中填写访问指定 S3 存储桶所需的 Amazon S3 域及其提供的 `rds-combined-ca-bundle.pem` 文件。

### 使用 AWS DMS 控制台创建目标端点

此端点适用于已经在运行的 Redis 目标。

- 在控制台的导航窗格中，选择端点，然后选择创建端点。下表描述了设置。

对于该选项	请执行该操作
终端节点类型	选择目标端点类型。
终端节点标识符	输入端点名称。例如，在名称中包含端点的类型，如 <b>my-redis-target</b> 。
目标引擎	选择 Redis 作为您希望此端点连接的数据库引擎类型。
集群名称	输入您的 Redis 集群的 DNS 名称。
端口	输入您的 Redis 集群的端口号。
SSL 安全协议	选择纯文本或 SSL 加密。  纯文本 – 该选项不为端点和数据库之间的流量提供传输层安全性协议 (TLS) 加密。



对于该选项	请执行该操作
	<p>SSL 加密 – 如果选择此选项，请输入 SSL 证书颁发机构 (CA) 证书 ARN 以验证服务器的证书并建立加密连接。</p> <p>对于本地 Redis，DMS 同时支持公有和私有证书颁发机构 (CA)。对于 ElastiCache for Redis，DMS 仅支持公有 CA。</p>
身份验证类型	<p>选择连接到 Redis 目标时要执行的身份验证类型。选项包括无、身份验证角色和身份验证令牌。</p> <p>如果选择身份验证角色，请提供身份验证用户名和身份验证密码。</p> <p>如果选择身份验证令牌，则仅提供身份验证密码。</p>
复制实例	<p>[可选] 仅在您想要测试连接时，才选择先前在创建复制实例页面上输入的复制实例的名称。</p>

提供端点的所有信息后，AWS DMS 会创建您的 Redis 目标端点以供数据库迁移期间使用。

有关创建迁移任务和开始数据库迁移的信息，请参阅[创建任务](#)。

## 指定 Redis 作为目标时的端点设置

要创建或修改目标终端节点，您可以使用控制台，也可以使用 CreateEndpoint 或 ModifyEndpoint API 操作。

对于 Redis 目标，在 AWS DMS 控制台中的创建端点或修改端点控制台页面上指定特定于端点的设置。

使用 CreateEndpoint 和 ModifyEndpoint API 操作时，请为 RedisSettings 选项指定请求参数。以下示例说明如何使用 AWS CLI 执行此操作。

```
aws dms create-endpoint --endpoint-identifier my-redis-target
--endpoint-type target --engine-name redis --redis-settings
'{"ServerName": "sample-test-sample.zz012zz.cluster.eee1.cache.bbbxxx.com", "Port": 6379, "AuthType": "auth-token",
```

```
"SslSecurityProtocol":"ssl-encryption", "AuthPassword":"notanactualpassword"}'

{
  "Endpoint": {
    "EndpointIdentifier": "my-redis-target",
    "EndpointType": "TARGET",
    "EngineName": "redis",
    "EngineDisplayName": "Redis",
    "TransferFiles": false,
    "ReceiveTransferredFiles": false,
    "Status": "active",
    "KmsKeyId": "arn:aws:kms:us-east-1:999999999999:key/x-b188188x",
    "EndpointArn": "arn:aws:dms:us-
east-1:555555555555:endpoint:ABCDEFGHIJKLMONOPQRSTUVWXYZ",
    "SslMode": "none",
    "RedisSettings": {
      "ServerName": "sample-test-sample.zz012zz.cluster.eee1.cache.bbbxxx.com",
      "Port": 6379,
      "SslSecurityProtocol": "ssl-encryption",
      "AuthType": "auth-token"
    }
  }
}
```

--redis-settings 参数如下：

- ServerName – ( 必填 ) 类型为 string，指定数据将迁移到的 Redis 集群，并且位于同一 VPC 中。
- Port – ( 必填 ) 类型为 number，用于访问端点的端口值。
- SslSecurityProtocol – ( 可选 ) 有效值包括 plaintext 和 ssl-encryption。默认值为 ssl-encryption。

plaintext 选项不为端点和数据库之间的流量提供传输层安全性协议 ( TLS ) 加密。

使用 ssl-encryption 建立加密连接。ssl-encryption 不需要 SSL 证书颁发机构 (CA) ARN 来验证服务器的证书，但可以选择使用 SslCaCertificateArn 设置进行识别。如果没有为证书颁发机构提供 ARN，则 DMS 将使用 Amazon 根 CA。

使用本地 Redis 目标时，您可以使用 `SslCaCertificateArn` 将公共或私有证书颁发机构 (CA) 导入 DMS，并提供该 ARN 用于服务器身份验证。使用 ElastiCache for Redis 作为目标时，不支持私有 CA。

- `AuthType` – (必填) 连接到 Redis 目标时要执行的身份验证类型。有效值包括 `none`、`auth-token` 和 `auth-role`。

`auth-token` 选项要求提供“*AuthPassword*”，而 `auth-role` 选项则要求提供“*AuthUserName*”和“*AuthPassword*”。

## 使用 Babelfish 作为 AWS Database Migration Service 的目标

您可以使用 AWS Database Migration Service 将数据从 Microsoft SQL Server 源数据库迁移到 Babelfish 目标。

Aurora PostgreSQL 的 Babelfish 扩展了您的 Amazon Aurora PostgreSQL 兼容版数据库，能够接受来自 Microsoft SQL Server 客户端的数据库连接。这样做可以让最初为 SQL Server 构建的应用程序可以直接与 Aurora PostgreSQL 协作，与传统迁移相比，代码更改很少，而且无需更改数据库驱动程序。

有关 AWS DMS 支持作为目标的 Babelfish 版本的信息，请参阅[目标 AWS DMS](#)。在使用 Babelfish 端点之前，需要升级 Aurora PostgreSQL 上 Babelfish 的早期版本。

### Note

Aurora PostgreSQL 目标端点是将数据迁移到 Babelfish 的首选方式。有关更多信息，请参阅[将适用于 Aurora PostgreSQL 的 Babelfish 作为目标](#)。

有关使用 Babelfish 作为数据库端点的信息，请参阅《Amazon Aurora 用户指南》中的[适用于 Aurora PostgreSQL 的 Babelfish](#)。

## 使用 Babelfish 作为 AWS DMS 的目标的先决条件

在迁移数据之前，必须先创建表，以确保表 AWS DMS 使用正确的数据类型和表元数据。如果您在运行迁移之前未在目标上创建表，AWS DMS 可能会使用不正确的数据类型和权限创建表。例如，AWS DMS 改为将时间戳列创建为 `binary(8)`，并且不提供预期的时间戳/行版本功能。

### 在迁移之前准备和创建表

1. 运行包含任何唯一约束、主键或默认约束的创建表 DDL 语句。

请勿为视图、存储过程、函数或触发器等对象包含外键约束或任何 DDL 语句。您可以在迁移源数据库后应用它们。

2. 为您的表标识任何标识列、计算列或包含行版本或时间戳数据类型的列。然后，创建必要的转换规则，以便在运行迁移任务时处理已知问题。有关更多信息，请参阅 [转换规则和操作](#)。
3. 识别包含 Babelfish 不支持的数据类型的列。然后，将目标表中受影响的列更改为使用支持的数据类型，或者创建在迁移任务期间将这些列删除的转换规则。有关更多信息，请参阅 [转换规则和操作](#)。

下表列出了 Babelfish 不支持的源数据类型，以及推荐使用的相应目标数据类型。

源数据类型	推荐的 Babelfish 数据类型
HEIRARCHYID	NVARCHAR(250)
GEOMETRY	VARCHAR(MAX)
GEOGRAPHY	VARCHAR(MAX)

为 Aurora PostgreSQL Serverless V2 源数据库设置 Aurora 容量单位 (ACU) 级别

在运行 AWS DMS 迁移任务之前，您可以通过设置最小 ACU 值来提高其性能。

- 在 Severless v2 容量设置窗口中，将 Aurora 数据库集群的最小 ACU 设置为 2 或合理级别。

有关设置 Aurora 容量单位的其他信息，请参阅《Amazon Aurora 用户指南》中的 [为 Aurora 集群选择 Aurora Serverless v2 容量范围](#)。

运行 AWS DMS 迁移任务后，您可以将 Aurora PostgreSQL Serverless V2 源数据库的 ACU 的最小值重置为合理级别。

## 将 Babelfish 作为 AWS Database Migration Service 目标时的安全要求

下面介绍将 AWS DMS 和 Babelfish 目标结合使用时的安全要求：

- 用于创建数据库的管理员用户名（管理员用户）。
- PSQL 登录角色和具有足够的 SELECT、INSERT、UPDATE、DELETE 和 REFERENCES 权限的用户。

## 使用 Babelfish 作为 AWS DMS 目标的用户权限

### Important

出于安全目的，用于数据迁移的用户账户，必须是您作为目标的任意 Babelfish 数据库中的注册用户。

您的 Babelfish 目标端点需要最低的用户权限才能运行 AWS DMS 迁移。

### 创建登录角色和低权限的 Transact-SQL (T-SQL) 用户

1. 创建连接到服务器时使用的登录角色和密码。

```
CREATE LOGIN dms_user WITH PASSWORD = 'password';  
GO
```

2. 为您的 Babelfish 集群创建虚拟数据库。

```
CREATE DATABASE my_database;  
GO
```

3. 为您的目标数据库创建 T-SQL 用户。

```
USE my_database  
GO  
CREATE USER dms_user FOR LOGIN dms_user;  
GO
```

4. 对于 Babelfish 数据库中的每个表，授予对于这些表的权限。

```
GRANT SELECT, DELETE, INSERT, REFERENCES, UPDATE ON [dbo].[Categories] TO dms_user;
```

## 使用 Babelfish 作为 AWS Database Migration Service 目标的限制

将 Babelfish 数据库作为 AWS DMS 目标时，存在以下限制：

- 仅支持不进行任何操作的表准备模式。
- ROWVERSION 数据类型需要一个表映射规则，该规则可在迁移任务期间从表中删除列名。

- 不支持 `sql_variant` 数据类型。
- 支持完整 LOB 模式。要使用 SQL Server 作为源端点，需要设置 SQL Server 端点连接属性设置 `ForceFullLob=True`，才能将 LOB 迁移到目标端点。
- 复制任务设置具有以下限制：

```
{
  "FullLoadSettings": {
    "TargetTablePrepMode": "DO_NOTHING",
    "CreatePkAfterFullLoad": false,
  }.
}
```

- Babelfish 中的 `TIME(7)`、`DATETIME2(7)` 和 `DATETIMEOFFSET(7)` 数据类型将时间中秒部分的精度值限制为 6 位数。使用这些数据类型时，请考虑对目标表使用精度值 6。对于 Babelfish 2.2.0 及更高版本，当使用 `TIME(7)` 和 `DATETIME2(7)` 时，精度的第七位数始终为零。
- 在 `DO_NOTHING` 模式下，DMS 会检查此表是否已存在。如果目标架构中不存在此表，DMS 会根据源表定义创建表，并将任何用户定义的数据类型映射到其基本数据类型。
- 针对 Babelfish 目标的 AWS DMS 迁移任务不支持包含使用 `ROWVERSION` 或 `TIMESTAMP` 数据类型的列的表。您可以使用表映射规则，此规则可在传输过程中从表中删除列名。在以下转换规则示例中，源中名为 `Actor` 的表经过转换后从目标上的 `Actor` 中删除了以字符 `col` 开头的所有列。

```
{
  "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%"
  }
}]
}

```

- 对于具有标识列或计算列的表，如果目标表使用混合大小写的名称（如 Categories），则必须创建转换规则操作，以便在 DMS 任务中将表名转换为小写形式。以下示例显示如何使用 AWS DMS 控制台创建转换规则操作设为小写。有关更多信息，请参阅 [转换规则和操作](#)。

▼ 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 ▼

- 对于 2.2.0 之前的 Babelfish 版本，DMS 将可以复制到 Babelfish 目标端点的列数限制为二十 (20) 列。在 Babelfish 2.2.0 中，此限制增加到 100 列。但在 Babelfish 2.4.0 及更高版本中，您可以复制的列数再次增加。您可以对 SQL Server 数据库运行以下代码示例，以确定哪些表太长。

```

USE myDB;
GO

```

```

DECLARE @Babelfish_version_string_limit INT = 8000; -- Use 380 for Babelfish versions
before 2.2.0
WITH bfindpoint
AS (
SELECT
  [TABLE_SCHEMA]
    , [TABLE_NAME]
    , COUNT( [COLUMN_NAME] ) AS NumberColumns
    , ( SUM( LEN( [COLUMN_NAME] ) + 3 )
+ SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
    + LEN( TABLE_SCHEMA ) + 3
+ 12 -- INSERT INTO string
+ 12) AS InsertIntoCommandLength -- values string
    , CASE WHEN ( SUM( LEN( [COLUMN_NAME] ) + 3 )
+ SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
    + LEN( TABLE_SCHEMA ) + 3
+ 12 -- INSERT INTO string
+ 12) -- values string
    >= @Babelfish_version_string_limit
    THEN 1
    ELSE 0
    END AS IsTooLong
FROM [INFORMATION_SCHEMA].[COLUMNS]
GROUP BY [TABLE_SCHEMA], [TABLE_NAME]
)
SELECT *
FROM bfindpoint
WHERE IsTooLong = 1
ORDER BY TABLE_SCHEMA, InsertIntoCommandLength DESC, TABLE_NAME
;

```

## Babelfish 的目标数据类型

下表显示了使用 AWS DMS 时支持的 Babelfish 目标数据类型以及来自 AWS DMS 数据类型的默认映射。

有关 AWS DMS 数据类型的其他信息，请参阅[AWS Database Migration Service 的数据类型](#)。

AWS DMS 数据类型	Babelfish 数据类型
BOOLEAN	TINYINT



AWS DMS 数据类型	Babelfish 数据类型
BYTES	VARBINARY(length)
DATE	DATE
TIME	TIME
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	FLOAT
STRING	<p>如果列是日期或时间列，请执行以下操作：</p> <ul style="list-style-type: none"> <li>对于 SQL Server 2008 及更高版本，请使用 DATETIME2。</li> <li>对于早期版本，如果小数位数等于或小于 3，请使用 DATETIME。在所有其他情况下，请使用 VARCHAR (37)。</li> </ul> <p>如果列不是日期或时间列，请使用 VARCHAR (length)。</p>
UINT1	TINYINT
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT

AWS DMS 数据类型	Babelfish 数据类型
WSTRING	NVARCHAR(length)
BLOB	VARBINARY(max)  要将此数据类型用于 DMS，必须允许对特定任务使用 BLOB。DMS 仅在包含主键的表中支持 BLOB 数据类型。
CLOB	VARCHAR(max)  要将此数据类型用于 DMS，必须允许对特定任务使用 CLOB。
NCLOB	NVARCHAR(max)  要将此数据类型用于 DMS，必须允许对特定任务使用 NCLOB。在 CDC 期间，DMS 仅在包含主键的表中支持 NCLOB 数据类型。

## 将 Amazon Timestream 作为 AWS Database Migration Service 的目标

您可以使用 AWS Database Migration Service 将数据从源数据库迁移到 Amazon Timestream 目标端点，并支持完全加载和 CDC 数据迁移。

Amazon Timestream 是一项快速、可扩展的无服务器时间序列数据库服务，专为大量数据摄取而构建。时间序列数据是在某个时间间隔内收集的一系列数据点，用于测量随时间变化的事件。它用于收集、存储和分析来自物联网应用程序、DevOps 应用程序和分析应用程序的指标。将数据存入 Timestream 后，您就可以近乎实时地可视化查看和识别数据的趋势和模式。有关 Amazon Timestream 的更多信息，请参阅《Amazon Timestream 开发人员指南》中的[什么是 Amazon Timestream？](#)

### 主题

- [将 Amazon Timestream 作为 AWS Database Migration Service 目标的先决条件](#)
- [多线程完全加载任务设置](#)
- [多线程 CDC 加载任务设置](#)
- [使用 Timestream 作为 AWS DMS 的目标时的端点设置](#)

- [创建和修改 Amazon Timestream 目标端点](#)
- [使用对象映射将数据迁移到 Timestream 主题](#)
- [将 Amazon Timestream 作为 AWS Database Migration Service 目标的限制](#)

## 将 Amazon Timestream 作为 AWS Database Migration Service 目标的先决条件

在将 Amazon Timestream 设置为 AWS DMS 的目标之前，请确保您创建了 IAM 角色。此角色必须允许 AWS DMS 访问要迁移到 Amazon Timestream 的数据。以下 IAM 策略中显示了您用于迁移到 Timestream 的角色所需具备的最低访问权限集。

```
{
  "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"
    }
  ]
}
```

```
    }  
  ]  
}
```

如果您打算迁移所有表，请将上面示例中的 `TABLE_NAME` 改成 `*`。

在将 Timestream 作为目标时，请注意以下几点：

- 如果您打算摄取时间戳超过 1 年的历史数据，我们建议使用 AWS DMS 以逗号分隔值 (csv) 格式将数据写入 Amazon S3。然后，使用 Timestream 的批量加载功能将数据摄取到 Timestream 中。有关更多信息，请参阅 [Amazon Timestream 开发人员指南](#) 中的 [在 Timestream 中使用批量加载](#)。
- 对于时间不足 1 年的数据的完全加载数据迁移，我们建议将 Timestream 表的内存存储保留期设置为大于或等于最早的时间戳。等到迁移完成后，再将表的内存存储保留时间编辑为所需的值。例如，要迁移最早时间戳为 2 个月的数据，请执行以下操作：
  - 将 Timestream 目标表的内存存储保留期设置为 2 个月。
  - 使用 AWS DMS 开始迁移数据。
  - 数据迁移完成后，将目标 Timestream 表的保留期更改为所需的值。

我们建议在迁移之前使用以下页面上的信息估算内存存储成本：

- [Amazon Timestream 定价](#)
- [AWS 定价计算器](#)
- 对于 CDC 数据迁移，我们建议设置目标表的内存存储保留期，使摄取的数据处于内存存储保留范围内。有关更多信息，请参阅 [Amazon Timestream 开发人员指南](#) 中的 [写入最佳实践](#)。

## 多线程完全加载任务设置

为了帮助提高数据传输速度，AWS DMS 支持通过以下任务设置，针对 Timestream 目标端点执行多线程完全加载迁移任务：

- `MaxFullLoadSubTasks` – 使用此选项指示要并行加载的表的最大数目。DMS 使用专用的子任务将各个表加载到其对应的 Amazon Timestream 目标表。默认值为 8；最大值为 49。
- `ParallelLoadThreads` – 使用此选项指定 AWS DMS 将各个表加载到其 Amazon Timestream 目标表时使用的线程数。Timestream 目标的最大值为 32。您可以请求提高此最大值限制。
- `ParallelLoadBufferSize` – 使用此选项指定在缓冲区（并行加载线程将数据加载到 Amazon Timestream 目标时使用）中存储的最大记录数。默认值是 50。最大值为 1000。将此设置与 `ParallelLoadThreads` 一起使用；仅在有多个线程时 `ParallelLoadBufferSize` 才有效。

- `ParallelLoadQueuesPerThread` – 使用此选项可以指定每个并发线程访问的队列数，以便从队列中取出数据记录并为目标生成批处理负载。默认为 1。但是，对于各种负载大小的 Amazon Timestream 目标，有效范围为每个线程 5-512 个队列。

## 多线程 CDC 加载任务设置

要提高 CDC 性能，请 AWS DMS 支持以下任务设置：

- `ParallelApplyThreads` – 指定 AWS DMS 在 CDC 加载期间使用的并发线程数，以便将数据记录推送到 Timestream 目标端点。默认值是 0，最大值是 32。
- `ParallelApplyBufferSize` – 指定在 CDC 加载过程中，要在每个缓冲区队列中存储的、供并发线程推送到 Timestream 目标端点的最大记录数。默认值是 100，最大值是 1,000。当 `ParallelApplyThreads` 指定多个线程时，请使用此选项。
- `ParallelApplyQueuesPerThread` – 指定每个线程访问以将数据记录从队列中取出并在 CDC 期间为 Timestream 端点生成批处理负载的队列数。默认值是 1，最大值是 512。

## 使用 Timestream 作为 AWS DMS 的目标时的端点设置

您可以使用端点设置来配置 Timestream 目标数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定对应设置，或者在 [AWS CLI](#) 中使用符合 `--timestream-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令。

下表显示了将 Timestream 作为目标时您可以使用的端点设置。

名称	描述
<code>MemoryDuration</code>	<p>设置此属性可指定将迁移的数据存储在 Timestream 内存存储中的保留期。时间以小时为单位进行测量。Timestream 的内存存储经过优化，可实现高摄取吞吐量和快速访问。</p> <p>默认值：24 (小时)</p> <p>有效值：1 到 8736 (1 小时到 12 个月，以小时为单位)</p> <p>例如：<code>--timestream-settings '{"MemoryDuration": 20}'</code></p>

名称	描述
DatabaseName	<p>设置此属性可指定目标 Timestream 数据库名称。</p> <p>类型：字符串</p> <p>例如：<code>--timestream-settings '{"DatabaseName": "db_name"}</code>'</p>
TableName	<p>设置此属性可指定目标 Timestream 表名称。</p> <p>类型：字符串</p> <p>例如：<code>--timestream-settings '{"TableName": "table_name"}</code>'</p>
MagneticDuration	<p>设置此属性可指定应用于 Timestream 表的磁介质保留期（以天为单位）。这是摄取的数据的保留期限。Timestream 会删除任何超过保留期限的时间戳。有关更多信息，请参阅 <a href="#">Amazon Timestream 开发人员指南</a> 中的 <a href="#">存储</a>。</p> <p>例如：<code>--timestream-settings '{"MagneticDuration": "3"}</code>'</p>
CdcInsertsAndUpdates	<p>将此属性设置为 <code>true</code> 可指定 AWS DMS 仅应用插入和更新，而不应用删除。Timestream 不允许删除记录，因此，如果此值为 <code>false</code>，AWS DMS 会将 Timestream 数据库中的相应记录置空，而不是删除。有关更多信息，请参阅下面的 <a href="#">限制</a>。</p> <p>默认值：<code>false</code></p> <p>例如：<code>--timestream-settings '{"CdcInsertsAndUpdates": "true"}</code>'</p>

名称	描述
EnableMagneticStoreWrites	<p>将此属性设置为 true 可启用磁介质存储写入。当此值为 false 时，AWS DMS 不写入时间戳早于目标表内存存储保留期的记录，因为 Timestream 默认不允许磁介质存储写入。有关更多信息，请参阅 <a href="#">Amazon Timestream 开发人员指南</a> 中的 <a href="#">写入最佳实践</a>。</p> <p>默认值：false</p> <p>例如：<code>--timestream-settings '{"EnableMagneticStoreWrites": "true"}'</code></p>

## 创建和修改 Amazon Timestream 目标端点

创建 IAM 角色并确定最小访问权限集后，您可以使用 AWS DMS 控制台，也可以使用 [AWS CLI](#) 中的 `create-endpoint` 命令和 `--timestream-settings '{"EndpointSetting": "value", ...}'` JSON 语法创建 Amazon Timestream 目标端点。

以下示例演示如何使用 AWS CLI 创建和修改 Timestream 目标端点。

### 创建 Timestream 目标端点命令

```
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,
}
```

### 创建 Timestream 目标端点命令

```
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,
}
```

## 使用对象映射将数据迁移到 Timestream 主题

AWS DMS 使用表映射规则将数据从源映射到目标 Timestream 主题。要将数据映射到目标主题，您必须使用称为对象映射的表映射规则类型。可以使用对象映射来定义源中的数据记录如何映射到发布到 Timestream 主题的数据记录。

除了具有分区键以外，Timestream 主题没有预设结构。

### Note

您不一定要使用对象映射。可以使用常规表映射进行各种转换。但是，分区键类型将遵循以下默认行为：

- 主键用作完全加载时的分区键。
- 如果未使用并行应用任务设置，`schema.table` 将用作 CDC 的分区键。
- 如果未使用并行应用任务设置，主键则用作 CDC 的分区键。

要创建对象映射规则，请将 `rule-type` 指定为 `object-mapping`。此规则指定您要使用的对象映射的类型。规则的结构如下所示。

```
{
  "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 目前支持 `map-record-to-record` 和 `map-record-to-document` 作为 `rule-action` 参数的唯一有效值。`map-record-to-record` 和 `map-record-to-document` 值指定 AWS DMS 默认情况下对未作为 `exclude-columns` 属性列表的一部分排除的记录执行的操作。这些值不会以任何方式影响属性映射。

从关系数据库迁移到 Timestream 主题时使用 `map-record-to-record`。此规则类型使用关系数据库中的 `taskResourceId.schemaName.tableName` 值作为 Timestream 主题中的分区键，并为源数据库中的每个列创建一个属性。在使用 `map-record-to-record` 时，对于源表中未在 `exclude-columns` 属性列表中列出的任何列，AWS DMS 将在目标主题中创建对应的属性。不论是否在属性映射中使用源列，都会创建对应的属性。

了解 `map-record-to-record` 的一种方法是在操作时加以观察。对于本示例，假定您使用关系数据库表行开始处理，该行具有以下结构和数据。

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

要将此信息从名为 Test 的架构迁移到 Timestream 主题，需要创建规则来将数据映射到目标主题。以下规则对此映射进行了说明。

```
{
  "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"
      }
    }
  ]
}
```

给定 Timestream 主题和分区键（在本例中为 `taskResourceId.schemaName.tableName`），下面说明了使用 Timestream 目标主题中的示例数据生成的记录格式：

```
{
  "FirstName": "Randy",
  "LastName": "Marsh",
  "StoreId": "5",
  "HomeAddress": "221B Baker Street",
  "HomePhone": "1234567890",
  "WorkAddress": "31 Spooner Street, Quahog",
  "WorkPhone": "9876543210",
  "DateOfBirth": "02/29/1988"
}
```

## 将 Amazon Timestream 作为 AWS Database Migration Service 目标的限制

将 Amazon Timestream 作为目标时存在以下限制：

- **维度和时间戳：**Timestream 像使用复合主键一样使用源数据中的维度和时间戳，并且不允许更新插入这些值。这意味着，如果您更改源数据库中记录的时间戳或维度，Timestream 数据库将尝试创建新记录。因此，如果您更改一条记录的维度或时间戳，使其与另一条现有记录的维度或时间戳相匹配，AWS DMS 可能会更新另一条记录的值，而不是创建新记录或更新先前的相应记录。
- **DDL 命令：**当前版本的 AWS DMS 仅支持 CREATE TABLE 和 DROP TABLE DDL 命令。
- **记录限制：**Timestream 对记录有限制，例如记录大小和度量值大小。有关更多信息，请参阅《Amazon Timestream 开发人员指南》<https://docs.aws.amazon.com/>中的[配额](#)。
- **删除记录和空值：**Timestream 不支持删除记录。为了支持迁移从源中删除的记录，AWS DMS 会清除 Timestream 目标数据库中的记录中的相应字段。AWS DMS 会更改相应目标记录的字段值，将数值字段更改为 0，将文本字段更改为 null，对于布尔型字段，更改为 false。
- 将 Timestream 作为目标不支持非关系数据库 (RDBMS) 的源。
- AWS DMS 仅在以下区域中支持将 Timestream 作为目标：
  - 美国东部 (弗吉尼亚州北部)
  - 美国东部 (俄亥俄州)
  - 美国西部 (俄勒冈州)
  - 欧洲地区 (爱尔兰)
  - 欧洲地区 (法兰克福)
  - 亚太地区 (悉尼)
  - 亚太地区 (东京)
- 将 Timestream 作为目标不支持将 TargetTablePrepMode 设置为 TRUNCATE\_BEFORE\_LOAD。我们建议将 DROP\_AND\_CREATE 用于此设置。

## 将 Amazon RDS for Db2 和 IBM Db2 LUW 作为 AWS DMS 的目标

您可以使用 AWS Database Migration Service (AWS DMS) 将数据从 Db2 LUW 数据库迁移到 Amazon RDS for Db2 或本地 Db2 数据库。

有关 AWS DMS 支持作为目标的 Db2 LUW 版本的信息，请参阅 [的目标 AWS DMS](#)。

可以使用安全套接字层 (SSL) 加密 Db2 LUW 端点与复制实例之间的连接。有关将 SSL 用于 Db2 LUW 端点的更多信息，请参阅 [将 SSL 与 AWS Database Migration Service](#)。

### 将 Db2 LUW 作为 AWS DMS 目标的限制

将 Db2 LUW 数据库作为 AWS DMS 的目标时，存在以下限制。有关将 Db2 LUW 作为源时的限制，请参阅 [使用 Db2 LUW 作为源代码时的限制 AWS DMS](#)。

- 只有当源是 Db2 LUW 或 Db2 for z/OS 时，AWS DMS 才支持将 Db2 LUW 作为目标。
- 使用 Db2 LUW 作为目标不支持完整 LOB 模式的复制。
- 在完全加载阶段，使用 Db2 LUW 作为目标不支持 XML 数据类型。这是 IBM dbload 实用工具的一个限制。有关更多信息，请参阅 IBM Informix 服务器文档中的 [dbload 实用工具](#)。
- AWS DMS 会使用与双引号字符 (") 对应的值截断 BLOB 字段。这是 IBM dbload 实用工具的一个限制。

### 使用 Db2 LUW 作为 AWS DMS 的目标时的端点设置

您可以使用端点设置来配置 Db2 LUW 目标数据库，这与使用额外连接属性类似。您可以在使用 AWS DMS 控制台创建目标端点时指定对应设置，或者在 [AWS CLI](#) 中使用符合 `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` JSON 语法的 `create-endpoint` 命令。

下表显示了将 Db2 LUW 用作目标时可以使用的端点设置。

名称	描述
KeepCsvFiles	如果为 true，AWS DMS 会将用于复制数据的任何 .csv 文件保存到 Db2 LUW 目标。DMS 使用这些文件进行分析和故障排查。
LoadTimeout	AWS DMS 将 DMS 对 Db2 目标执行的操作设置为超时之前的时间（以毫秒为单位）。默认值为 1200（20 分钟）。

名称	描述
MaxFileSize	指定用于将数据传输到 Db2 LUW 的 .csv 文件的最大大小 (以 KB 为单位)。
WriteBufferSize	在 DMS 复制实例的本地磁盘上生成 .csv 文件时使用的内存中文件写入缓冲区的大小 (以 KB 为单位)。默认值为 1024 (1 MB)。

## 将 VPC 端点配置为 AWS DMS 源端点和目标端点

AWS DMS 支持将 Amazon Virtual Private Cloud (VPC) 端点作为源和目标。AWS 只要源数据库和目标数据库在其 AWS DMS VPC 中明确定义了指向它们的路径，DMS 就可以通过 Amazon VPC 端点连接到任何 AWS 源或目标数据库。

通过支持 Amazon VPC 端点，AWS DMS 可以更轻松地维护所有复制任务的端到端网络安全，而无需额外的联网配置和设置。为所有源端点和目标端点使用 VPC 端点，可确保您的所有流量保持在 VPC 内并处于您的控制之下。升级到 AWS DMS 3.4.7 及更高版本时，需要将 AWS DMS 配置为使用 VPC 端点，或者使用公共路由连接到与下列 Amazon Web Services 交互的所有源端点和目标端点：

- Amazon S3
- Amazon Kinesis
- AWS Secrets Manager
- Amazon DynamoDB
- Amazon Redshift
- Amazon OpenSearch Service

从版本 3.4.7 开始，您可能需要使用 VPC 端点来支持 AWS DMS，如下所述。

### 迁移到 AWS DMS 版本 3.4.7 及更高版本时，谁会受到影响？

如果您在使用先前列出的一个或多个 AWS DMS 端点，并且这些端点不可公开路由，或者它们没有与之关联的 VPC 端点，则您会受到影响。

### 迁移到 AWS DMS 版本 3.4.7 及更高版本时，谁不会受到影响？

在以下情况下，您不会受到影响：

- 您没有使用前面列出的一个或多个 AWS DMS 端点。
- 您在使用前面列出的任何端点，并且它们可以公开路由。
- 您在使用前面列出的任何端点，并且它们具有与之关联的 VPC 端点。

## 准备迁移到 AWS DMS 3.4.7 及更高版本

当您在使用前面描述的任何端点时，为了防止 AWS DMS 任务失败，请在将 AWS DMS 升级到版本 3.4.7 或更高版本之前执行以下步骤之一：

- 使受影响的 AWS DMS 端点可公开路由。例如，向 AWS DMS 复制实例已使用的任何 VPC 添加互联网网关 (IGW) 路由，使其所有源端点和目标端点均可公开路由。
- 创建 VPC 端点以访问 AWS DMS 使用的所有源端点和目标端点，如下所述。

对于您用于 AWS DMS 源端点和目标端点的任何现有 VPC 端点，请确保它们使用的信任策略符合 XML 策略文档 `dms-vpc-role`。有关此 XML 策略文档的更多信息，请参阅[创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

否则，请将 VPC 端点添加到包含您的复制实例的 VPC 端点，以将复制实例配置为 VPC 端点。如果您配置的复制实例没有公有端点，请向包含您的复制实例的 VPC 添加可公开访问的 VPC 端点，这样就可以让这些端点可公开访问。您无需执行其他任何操作，即可专门将您的复制实例与 VPC 端点关联。

### Note

不同的服务可能具有唯一的 VPC 端点配置。例如，在使用 AWS Secrets Manager 时，您通常不需要调整路由表。务必检查每项服务的具体要求。

在包含您的复制实例的 VPC 上创建 VPC 端点

1. 登录到 AWS Management Console，然后通过以下网址打开 Amazon VPC 控制台：<https://console.aws.amazon.com/vpc/>。
2. 在 VPC 控制台菜单栏上，选择与您的 AWS DMS 复制实例相同的 AWS 区域。
3. 在 VPC 导航窗格中，选择端点。
4. 在端点下，选择创建端点。
5. 您可以选择指定名称标签。例如，**my-endpoint-DynamoDB-01**。
6. 仅在 S3 或 DynamoDB 的服务下，选择其类型设置为网关的服务名称。

7. 在 VPC 下，选择与 AWS DMS 复制实例相同的 VPC 来创建端点。
8. 在路由表下，选择所有可用的路由表 ID 值。
9. 要指定访问控制，请在策略下选择完全访问权限。如果要使用策略创建工具来指定自己的访问控制，请选择自定义。在任何情况下，请使用符合 JSON 策略文档 `dms-vpc-role` 的信任策略。有关此策略文档的更多信息，请参阅[创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。
10. 在端点下，确认您新创建的 VPC 端点状态是否为可用。

有关为 AWS DMS 复制实例配置 VPC 端点的更多信息，请参阅[数据库迁移的网络配置](#)。有关创建用于访问 AWS 服务的接口 VPC 端点的更多信息，请参阅《AWS PrivateLink 指南》中的[使用接口 VPC 端点访问 AWS 服务](#)。有关 VPC 端点的 AWS DMS 区域可用性的信息，请参阅[AWS 区域表](#)。

## AWS DMS 支持的 DDL 语句

数据迁移流程期间，您可以在源数据库上执行数据定义语言 (DDL) 语句。这些语句由复制服务器复制到目标数据库。

支持以下 DDL 语句：

- 创建表
- 删除表
- 重命名表
- 截断表
- 添加列
- 删除列
- 重命名列
- 更改列数据类型

对于某些源引擎类型，DMS 不会捕获所有支持的 DDL 语句。而且，在将 DDL 语句应用到特定目标引擎时，DMS 会以不同的方式处理 DDL 语句。有关特定源支持哪些 DDL 语句以及如何将其应用到目标的信息，请参阅该源端点和目标端点的具体文档主题。

您可以使用任务设置来配置 DMS 在更改数据捕获 (CDC) 期间处理 DDL 行为的方式。有关更多信息，请参阅[更改处理 DDL 处理的任务设置](#)。

# 使用 AWS DMS 任务

AWS Database Migration Service (AWS DMS) 任务是处理所有工作的地方。您可以指定要用于迁移的表 ( 或视图 ) 和架构以及任何特殊处理，例如，日志记录要求、控制表数据以及错误处理。

任务可能包括三个主要阶段：

- 迁移现有数据 ( 完全加载 )
- 应用缓存的更改
- 持续复制 ( 更改数据捕获 )

有关 AWS DMS 迁移任务如何迁移数据的更多信息和概述，请参阅 [的高级视图 AWS DMS](#)

在创建迁移任务时，您需要了解一些事项：

- 请确保先创建源终端节点、目标终端节点以及复制实例，然后才能创建任务。
- 您可以指定多项任务设置来定制您的迁移任务。您可以使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS DMS API 来进行这些设置。这些设置包括指定如何处理迁移错误、错误日志记录和控制表信息。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。
- 创建任务后，您可以立即运行它。将自动创建和加载具有必要的元数据定义的目标表，并且您可以指定持续复制。
- 默认情况下，AWS DMS 会在您创建任务后立即启动它。但是，在一些情况下，您可能需要推迟任务的启动。例如，在使用 AWS CLI 时，可能有一个进程创建任务，另一个进程根据一些触发事件启动任务。根据需要，您可以推迟任务的开始。
- 您可以使用 AWS CLI 控制台或 AWS DMS API 监控、停止或重新启动任务。有关使用 AWS DMS API 停止任务的信息，请参阅 [AWS DMS API 参考](#) 中的 [StopReplicationTask](#)。

以下是在使用 AWS DMS 任务时可以执行的操作。

Task	相关文档
创建任务	<a href="#">创建任务</a>
在创建任务时，您可以指定源、目标和复制实例以及任何迁移设置。	



Task	相关文档
<p data-bbox="115 226 370 260"><b>创建持续复制任务</b></p> <p data-bbox="115 306 623 386">您可以设置任务，以便在源和目标之间提供持续复制。</p> <p data-bbox="115 432 305 466"><b>应用任务设置</b></p> <p data-bbox="115 512 634 785">每个任务都具有可根据数据库迁移需求配置的设置。您在 JSON 文件中创建这些设置，对于某些设置，可以使用 AWS DMS 控制台指定它们。有关如何使用任务配置文件来设置任务设置的信息，请参阅<a href="#">任务设置示例</a>。</p>	<p data-bbox="693 226 1211 260"><a href="#">使用 AWS DMS 为持续复制创建任务</a></p> <p data-bbox="693 432 1471 466"><a href="#">为 AWS Database Migration Service 任务指定任务设置</a></p>
<p data-bbox="115 835 272 869"><b>使用表映射</b></p> <p data-bbox="115 915 623 1188">表映射使用多种类型的规则为表指定其他任务设置。这些规则允许您指定数据来源、源架构、表和视图、数据、任务期间要进行的任何表和数据转换，以及如何将这些表和列从源迁移到目标的设置。</p>	<p data-bbox="693 835 818 869"><b>选择规则</b></p> <p data-bbox="693 915 915 949"><a href="#">选择规则和操作</a></p> <p data-bbox="693 995 818 1029"><b>转换规则</b></p> <p data-bbox="693 1075 915 1108"><a href="#">转换规则和操作</a></p> <p data-bbox="693 1155 850 1188"><b>表设置规则</b></p> <p data-bbox="693 1234 1040 1268"><a href="#">表和集合设置规则和操作</a></p>
<p data-bbox="115 1310 399 1344"><b>运行迁移前任务评测</b></p> <p data-bbox="115 1390 623 1801">您可以启用和运行迁移前任务评测，以显示支持的源数据库和目标数据库中可能存在的问题，它们在迁移期间可能会导致问题。这可以包括不支持的数据类型、索引和主键不匹配以及其他冲突的任务设置等问题。这些迁移前评测在您运行任务之前运行，以便提前识别迁移期间可能会出现的问题。</p>	<p data-bbox="693 1310 1105 1344"><a href="#">为任务启用和使用迁移前评估</a></p>

Task	相关文档
<p><b>数据验证</b></p> <p>数据验证是一种任务设置，可用于让 AWS DMS 比较目标数据存储上的数据与源数据存储中的数据。</p>	<p><a href="#">AWS DMS 数据验证</a>.</p>
<p><b>修改任务</b></p> <p>停止任务后，您可以修改任务的设置。</p>	<p><a href="#">修改任务</a></p>
<p><b>移动任务</b></p> <p>任务停止后，您可以将任务移动到其他复制实例。</p>	<p><a href="#">移动任务</a></p>
<p><b>在任务期间重新加载表</b></p> <p>如果任务期间出错，您可以重新加载表。</p>	<p><a href="#">在任务期间重新加载表</a></p>
<p><b>应用筛选器</b></p> <p>您可以使用源筛选器来限制从源传输到目标的记录的数量和类型。例如，您可以指定仅将总部员工移动到目标数据库。还可以对数据列应用筛选器。</p>	<p><a href="#">使用源筛选器</a></p>
<p><b>监控任务</b></p> <p>可通过多种方式获取有关任务性能以及任务使用的表的信息。</p>	<p><a href="#">监控 AWS DMS 任务</a></p>
<p><b>管理任务日志</b></p> <p>您可以使用 AWS DMS API 或 AWS CLI 查看和删除任务日志。</p>	<p><a href="#">查看和管理 AWS DMS 任务日志</a></p>

## 主题

- [创建任务](#)
- [使用 AWS DMS 为持续复制创建任务](#)
- [修改任务](#)
- [移动任务](#)
- [在任务期间重新加载表](#)
- [使用表映射指定任务设置](#)
- [使用源筛选器](#)
- [为任务启用和使用迁移前评估](#)
- [指定任务设置的补充数据](#)

## 创建任务

要创建 AWS DMS 迁移任务，请执行以下操作：

- 在创建迁移任务之前，创建源终端节点、目标终端节点以及复制实例。
- 选择迁移方法：
  - 将数据迁移到目标数据库 – 该过程将在目标数据库中创建文件或表，并自动定义目标所需的元数据。它还会用源数据填充表。为提高效率，系统将并行加载表中的数据。此过程是中的“迁移现有数据”选项，AWS Management Console 在 API Full Load 中调用。
  - 迁移期间捕获更改 – 该过程捕获在将数据从源迁移到目标时对源数据库进行的更改。在完成最初请求的数据的迁移后，更改数据捕获 (CDC) 过程会将捕获到的更改应用于目标数据库。捕获更改并以单个提交事务为单位应用更改，并且可以将多个不同的目标表作为单个源提交进行更新。此方法可保证目标数据库中的事务完整性。该过程是控制台中的 Migrate existing data and replicate ongoing changes (迁移现有数据并复制持续更改) 选项，在 API 中称为 full-load-and-cdc。
  - 仅复制源数据库上的数据更改 – 该过程读取源数据库管理系统 (DBMS) 的恢复日志文件，并将每个事务的条目组合在一起。在某些情况下，AWS DMS 无法在合理的时间内对目标应用更改（例如，如果目标不可访问）。在这些情况下，只要有必要，就会在复制服务器上 AWS DMS 缓冲更改的时间。它不会重新读取源 DBMS 日志，这可能需要大量时间。该过程是 控制台中的仅复制数据更改 AWS DMS 选项。
- 确定该任务如何处理源上的大型二进制对象 (LOB)。有关更多信息，请参阅 [在任务中为源数据库设置 LOB 支持 AWS DMS](#)。

- 指定迁移任务设置。这些设置包括设置日志记录，指定写入到迁移控制表的数据，如何处理错误以及其他设置。有关任务设置的更多信息，请参阅[为 AWS Database Migration Service 任务指定任务设置](#)。
- 设置表映射以定义用于选择和筛选要迁移的数据的规则。有关表映射的更多信息，请参阅[使用表映射指定任务设置](#)。在指定映射之前，请确保查看有关源数据库和目标数据库之间的数据类型映射的文档部分。
- 在运行任务之前，启用并运行迁移前任务评估。有关迁移前评估的更多信息，请参阅[为任务启用和使用迁移前评估](#)。
- 指定迁移数据的任务所需的任何补充数据。有关更多信息，请参阅[指定任务设置的补充数据](#)。

您可以选择在 Create task (创建任务) 页面上指定任务的信息后立即启动任务。或者，您也可以稍后从控制面板页面启动任务。

以下过程假定您已经指定了复制实例信息和终端节点。有关设置端点的更多信息，请参阅[创建源和目标终端节点](#)。

## 创建迁移任务

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。

如果您以 AWS Identity and Access Management (IAM) 用户身份登录，请确保您拥有相应的访问权限 AWS DMS。有关所需权限的更多信息，请参阅[使用 AWS DMS 所需的 IAM 权限](#)。

2. 在导航窗格中选择数据库迁移任务，然后选择创建任务。
3. 在创建数据库迁移任务页面的任务配置部分，指定任务选项。下表描述了设置。

# 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

对于该选项

请执行该操作

任务标识符

输入任务的名称。

对于该选项	请执行该操作
描述性 Amazon 资源名称 ( ARN ) – 可选	用于覆盖默认 AWS DMS ARN 的友好名称。您在创建任务后无法更改该名称。
复制实例	显示要使用的复制实例。
源数据库端点	显示要使用的源终端节点。
目标数据库端点	显示要使用的目标终端节点。
迁移类型	选择要使用的迁移方法。您可以选择仅让现有数据迁移到目标数据库，或将持续更改以及迁移的数据发送到目标数据库。

4. 在任务设置部分，指定用于编辑任务、目标表格准备模式、停止任务、LOB 设置、验证和日志记录的值。

对于该选项	请执行该操作
编辑模式	选择是使用向导还是 JSON 编辑器来指定任务设置。如果选择向导，将显示以下选项。
源事务的 CDC 启动模式	<p>只有在前一部分中为迁移类型选择仅复制数据更改时，此设置才可见。</p> <p>禁用自定义 CDC 启动模式 – 如果选择此选项，则可以使用后面的创建时自动启动选项自动启动任务，也可以使用控制台手动启动任务。</p> <p>启用自定义 CDC 启动模式 – 如果选择此选项，则可以指定自定义的 UTC 开始时间来开始处理更改。</p>
Target table preparation mode (目标表准备模式)	<p>只有在前一部分中为迁移类型选择迁移现有数据或迁移现有数据并复制持续更改时，此设置才可见。</p> <p>什么都不做 — 在“什么都不做”模式下，AWS DMS 假设目标表已在目标系统上预先创建。如果表不为空，则数据迁移期间可能会发生冲突，并可能导致 DMS 任务错误。如果目标表不存在，DMS 将为您创建该表。</p>

## 对于该选项


## 请执行该操作

您的表结构将保留原样，并且任何现有数据将保留在表中。在目标表已经从源回填，并且应用了持续复制来确保源和目标同步时，Do nothing (不执行任何操作) 模式对于仅 CDC 任务是合适的选择。要预先创建表，您可以使用 AWS Schema Conversion Tool (AWS SCT)。有关更多信息，请参阅[安装 AWS SCT](#)。

删除目标中的表 – 在删除目标中的表模式下，开始迁移之前会删除目标表，然后重新创建它们。这种方法可确保迁移开始时目标表为空。AWS DMS 仅创建高效迁移数据所需的对象：表、主键，在某些情况下还会创建唯一索引。AWS DMS 不会创建二级索引、非主键约束或列数据默认值。如果您执行完全加载外加 CDC 或仅 CDC 任务，我们建议您此时暂停迁移。然后，创建支持筛选更新和删除语句的二级索引。

使用删除目标中的表模式时，您可能需要在目标数据库上执行某些配置。例如，对于 Oracle 目标，出于安全原因 AWS DMS 无法创建架构 (数据库用户)。在这种情况下，您可以预先创建架构用户，以便 AWS DMS 可以在迁移开始时创建表。对于大多数其他目标类型，使用正确的配置参数 AWS DMS 创建架构和所有关联表。

截断-在 Truncate AWS DMS 模式下，在迁移开始之前截断所有目标表。如果目标表不存在，DMS 将为您创建该表。您的表结构保持原样，但表在目标位置被截断。Truncate (截断) 模式适用于完全加载迁移或完全加载加 CDC 迁移，其中目标架构在迁移开始前已预先创建。要预先创建表，您可以使用 AWS SCT。有关更多信息，请参阅[安装 AWS SCT](#)。

 Note

如果您的目标是 MongoDB，则截断模式不会在目标位置截断表。相反，它会删除集合并丢失

对于该选项	请执行该操作
	<p>所有索引。当您的目标是 MongoDB 时，请避免使用截断模式。</p>
<p>在完全加载完成后停止任务</p>	<p>只有在前一部分中为迁移类型选择迁移现有数据并复制持续更改时，此设置才可见。</p> <p>不停止 – 不停止任务，但立即应用缓存的更改，然后继续。</p> <p>在应用缓存的更改前停止 – 在应用缓存的更改之前停止任务。使用此方式时，可以添加可能会加速更改应用的辅助索引。</p> <p>在应用缓存的更改之后停止 – 在应用缓存的更改后停止任务。使用此方式时，可以在使用事务应用时添加外键。</p>
<p>在复制时包括 LOB 列</p>	<p>不包括 LOB 列 – 从迁移操作中排除 LOB 列。</p> <p>完@@ 整 LOB 模式 — 无论大小如何，都迁移完整的 LOB。AWS DMS 在 LOB 区块大小参数控制的区块中分段迁移 LOB。此模式比受限 LOB 模式的速度要慢。</p> <p>受限 LOB 模式 – 将 LOB 截断为最大 LOB 大小参数的值。此模式比使用完整 LOB 模式的速度要快。</p>
<p>最大 LOB 大小 ( kb )</p>	<p>在受限 LOB 模式下，超出 Max LOB size (最大 LOB 大小) 设置的 LOB 列将被截断到指定的 Max LOB Size (最大 LOB 大小)。</p>
<p>启用验证</p>	<p>启用数据验证，以确认准确地将数据从源迁移到目标。有关更多信息，请参阅 <a href="#">AWS DMS 数据验证</a>。</p>
<p>启用 CloudWatch 日志</p>	<p>启用 Amazon 的日志记录 CloudWatch。</p>

- 在迁移前评估部分，选择是否运行迁移前评估。在启动数据库迁移任务之前，迁移前评估会提醒您注意潜在的迁移问题。有关更多信息，请参阅 [启用和使用迁移前评估](#)。



6. 在迁移任务启动配置部分，指定是否在创建任务后自动启动。
7. 在标签部分，指定组织任务所需的任意标签。您可以使用标签管理您的 IAM 角色和策略，并跟踪您的 DMS 成本。有关更多信息，请参阅 [为资源添加标签](#)。
8. 完成任务设置后，请选择 Create task (创建任务)。

## 为 AWS Database Migration Service 任务指定任务设置

每个任务都具有可根据数据库迁移需求配置的设置。您可以在 JSON 文件中创建这些设置，或者通过某些设置，您可以使用 AWS DMS 控制台指定设置。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

有几种主要类型的任务设置，如下所列。

### 主题

- [任务设置示例](#)
- [目标元数据任务设置](#)
- [完全加载任务设置](#)
- [Time Travel 任务设置](#)
- [日志记录任务设置](#)
- [控制表任务设置](#)
- [流缓冲区任务设置](#)
- [更改处理优化设置](#)
- [数据验证任务设置](#)
- [更改处理 DDL 处理的任务设置](#)
- [字符替换任务设置](#)
- [之前映像任务设置](#)
- [错误处理任务设置](#)
- [保存任务设置](#)

Task settings ( 任务设置 )	相关文档
创建任务评估报告	<a href="#">为任务启用和使用迁移前评估</a>

Task settings ( 任务设置 )	相关文档
<p>您可以创建一个任务评估报告，以显示在迁移期间可能出现问题的任何不支持的数据类型。在运行任务之前，您可以针对任务运行该报告以找出潜在的问题。</p>	
<p><b>创建任务</b></p> <p>在创建任务时，您可以指定源、目标和复制实例以及任何迁移设置。</p>	<p><a href="#">创建任务</a></p>
<p><b>创建持续复制任务</b></p> <p>您可以设置任务，以便在源和目标之间提供持续复制。</p>	<p><a href="#">使用 AWS DMS 为持续复制创建任务</a></p>
<p><b>应用任务设置</b></p> <p>每个任务都具有可根据数据库迁移需求配置的设置。您可以在 JSON 文件中创建这些设置，或者通过某些设置，您可以使用 AWS DMS 控制台指定设置。</p>	<p><a href="#">为 AWS Database Migration Service 任务指定任务设置</a></p>
<p><b>数据验证</b></p> <p>使用数据验证将目标数据存储中的数据与源数据存储中的数据 AWS DMS 进行比较。</p>	<p><a href="#">AWS DMS 数据验证</a></p>
<p><b>修改任务</b></p> <p>停止任务后，您可以修改任务的设置。</p>	<p><a href="#">修改任务</a></p>

Task settings ( 任务设置 )	相关文档
<p>在任务期间重新加载表</p> <p>如果任务期间出错，您可以重新加载表。</p>	<p><a href="#">在任务期间重新加载表</a></p>
<p>使用表映射</p> <p>表映射使用多个类型的规则来指定任务设置，例如数据源、源架构、数据以及任务期间将发生的任何转换。</p>	<p>选择规则 <a href="#">选择规则和操作</a></p> <p>转换规则 <a href="#">转换规则和操作</a></p>
<p>应用筛选器</p> <p>您可以使用源筛选器来限制从源传输到目标的记录的数量和类型。例如，您可以指定仅将总部员工移动到目标数据库。还可以对数据列应用筛选器。</p>	<p><a href="#">使用源筛选器</a></p>
<p>监控任务</p> <p>可通过多种方式获取有关任务性能以及任务使用的表的信息。</p>	<p><a href="#">监控 AWS DMS 任务</a></p>
<p>管理任务日志</p> <p>您可以使用 AWS DMS API 或查看和删除任务日志 AWS CLI。</p>	<p><a href="#">查看和管理 AWS DMS 任务日志</a></p>

## 任务设置示例

您可以使用 AWS Management Console 或 AWS CLI 来创建复制任务。如果您使用 AWS CLI，则可以通过创建 JSON 文件来设置任务设置，然后将 JSON 文件的 file://URI 指定为 [CreateReplication任务](#) 操作的 [ReplicationTaskSettings](#) 参数。

以下示例说明如何使用调 AWS CLI 用 CreateReplicationTask 操作：

```
aws dms create-replication-task \
```

```
--replication-task-identifier MyTask \  
--source-endpoint-arn arn:aws:dms:us-  
west-2:123456789012:endpoint:ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890ABC \  
--target-endpoint-arn arn:aws:dms:us-  
west-2:123456789012:endpoint:ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890ABC \  
--replication-instance-arn arn:aws:dms:us-  
west-2:123456789012:rep:ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890ABC \  
--migration-type cdc \  
--table-mappings file://tablemappings.json \  
--replication-task-settings file://settings.json
```

前面的示例使用名为 `tablemappings.json` 的表映射文件。有关表映射的示例，请参阅[使用表映射指定任务设置](#)。

任务设置 JSON 文件可能如下所示：

```
{  
  "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
  }
}

```

## 目标元数据任务设置

目标元数据设置包括：有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

- **TargetSchema** – 目标表架构名称。如果此元数据选项为空，则使用源表中的架构。如果未定义任何源架构，AWS DMS 自动将目标数据库的所有者前缀添加到所有表。对于 MySQL 类型的目标终端节点，该选项应保留为空。在数据映射中重命名架构优先于此设置。
- **LOB 设置** – 用于确定大对象 (LOB) 的管理方式的设置。如果您设置 `SupportLobs=true`，则必须将下列项之一设置为 `true`：
  - **FullLobMode** – 如果将该选项设置为 `true`，则必须为 `LobChunkSize` 选项输入值。输入在将数据复制到目标时要使用的 LOB 数据块的大小 (以 KB 为单位)。FullLobMode 选项最适合非常大的 LOB 大小，但往往会导致加载速度较慢。LobChunkSize 的建议值为 64 KB。将 LobChunkSize 的值增加到 64 KB 以上可能会导致任务失败。

- `InlineLobMaxSize`— 此值决定了在满载期间哪些 LOB 以内联方式 AWS DMS 传输。传输小型 LOB 比从源表查找它们的效率更高。在满载期间，AWS DMS 检查所有 LOB，并对小于的 LOB 执行内联传输。`InlineLobMaxSize` AWS DMS 传输所有大于 `in` 的 LOB `InlineLobMaxSize`。`FullLobModeInlineLobMaxSize` 的默认值是 0，范围为 1–102400 KB ( 100 MB )。仅当您知道大部分 LOB 小于 `InlineLobMaxSize` 中指定的值时，为 `InlineLobMaxSize` 设置值。
- `LimitedSizeLobMode` – 如果将该选项设置为 `true`，则必须为 `LobMaxSize` 选项输入值。输入单个 LOB 的最大大小 ( 以 KB 为单位 )。`LobMaxSize` 的最大建议值为 102400 KB ( 100 MB )。

有关这些任务 LOB 设置的使用条件的更多信息，请参阅[在任务中为源数据库设置 LOB 支持 AWS DMS](#)。您还可以控制各个表的 LOB 管理。有关更多信息，请参阅[表和集合设置规则和操作](#)。

- `LoadMaxFileSize` – 基于 CSV 的目标端点 ( 例如 MySQL、PostgreSQL 和 Amazon Redshift ) 的选项，支持使用逗号分隔值 ( .csv ) 文件加载数据。`LoadMaxFileSize` 定义磁盘上存储的、未加载的数据 ( 例如 .csv 文件 ) 的最大大小。该选项会覆盖目标端点连接属性 `maxFileSize`。您可以提供介于 0 ( 指示该选项不会覆盖连接属性 ) 和 100000 KB 之间的值。
- `BatchApplyEnabled` – 确定是否单独应用了每个事务或是否批量提交了更改。默认值为 `false`。

将 `BatchApplyEnabled` 设置为 `true` 时，DMS 要求源表上有主键 ( PK ) 或唯一密钥 ( UK )。如果源表上没有 PK 或 UK，则只能应用批量插入，不能应用批量更新和删除。

将 `BatchApplyEnabled` 设置为 `true` 时，如果目标表具有唯一约束和主键，则 AWS DMS 生成错误消息。如果 `BatchApplyEnabled` 设置为 `true`，则不支持同时具有唯一约束和主键的目标表。

如果设置 `BatchApplyEnabled` 为 `true`，并且在使用默认错误处理策略的表中 AWS DMS 遇到数据错误，则其余表的 AWS DMS 任务会从批处理 `one-by-one` 模式切换到模式。要更改此行为，可以在任务设置 JSON 文件的 "ErrorBehavior" 组属性中设置对以下策略的 "SUSPEND\_TABLE" 操作：

- `DataErrorPolicy`
- `ApplyErrorDeletePolicy`
- `ApplyErrorInsertPolicy`
- `ApplyErrorUpdatePolicy`



有关此 "ErrorBehavior" 组属性的更多信息，请参阅[AWS Database Migration Service 任务指定任务设置](#)中的任务设置 JSON 文件示例。将这些策略设置为后 "SUSPEND\_TABLE"，该 AWS DMS 任务会暂停所有引发错误的表上的数据错误，并继续以批处理模式处理所有表。

您可以将 BatchApplyEnabled 参数与 BatchApplyPreserveTransaction 参数一起使用。如果 BatchApplyEnabled 设置为 true，则 BatchApplyPreserveTransaction 参数确定事务完整性。

如果 BatchApplyPreserveTransaction 设置为 true，则保留事务完整性，并确保批处理包含源中的事务的所有更改。

如果 BatchApplyPreserveTransaction 设置为 false，则会临时放弃事务完整性以提高性能。

BatchApplyPreserveTransaction 参数仅适用于 Oracle 目标终端节点，并且仅在 BatchApplyEnabled 参数设置为 true 时相关。

如果复制中包含 LOB 列，则只能在受限 LOB 模式下使用 BatchApplyEnabled。

有关将这些设置用于更改数据捕获 (CDC) 负载的更多信息，请参阅[更改处理优化设置](#)。

- MaxFullLoadSubTasks – 指示要并行加载的表的最大数目。默认值为 8；最大值为 49。
- ParallelLoadThreads— 指定用于将每个表加载到目标数据库的线程数。AWS DMS 对于非 RDBMS 目标，此参数具有最大值。DynamoDB 目标的最大值为 200。亚马逊 Kinesis Data Streams、Apache Kafka 或 OpenSearch 亚马逊服务目标的最大值为 32。您可以要求增加该最大值。ParallelLoadThreads 适用于完全加载任务。有关单个表的并行加载设置的信息，请参阅[表和集合设置规则和操作](#)。

该设置适用于以下端点引擎类型：

- DynamoDB
- Amazon Kinesis Data Streams
- Amazon MSK
- 亚马逊 OpenSearch 服务
- Amazon Redshift

AWS DMS 支持 My ParallelLoadThreads SQL 作为额外的连接属性。

ParallelLoadThreads 不适用于作为任务设置的 MySQL。

- `ParallelLoadBufferSize` 指定在缓冲区（并行加载线程将数据加载到目标时使用）中存储的最大记录数。默认值是 50。最大值为 1000。目前，此设置仅在 DynamoDB、Kinesis、Apache Kafka 或是目标时才有效。OpenSearch 将此参数与 `ParallelLoadThreads` 一起使用。仅在有多个线程时 `ParallelLoadBufferSize` 才有效。有关单个表的并行加载设置的信息，请参阅 [表和集合设置规则和操作](#)。
- `ParallelLoadQueuesPerThread` – 指定每个并发线程访问的队列数，以便从队列中取出数据记录并为目标生成批处理负载。默认值是 1。目前，只有在 Kinesis 或 Apache Kafka 为目标时，此设置才有效。
- `ParallelApplyThreads`— 指定 CDC 加载期间 AWS DMS 用于将数据记录推送到亚马逊 DocumentDB、Kinesis、Amazon MSK OpenSearch 或 Amazon Redshift 目标终端节点的并发线程数。默认值为零（0）。

该设置仅适用于 CDC。该设置不适用于完全加载。

该设置适用于以下端点引擎类型：

- Amazon DocumentDB（与 MongoDB 兼容）
- Amazon Kinesis Data Streams
- Amazon Managed Streaming for Apache Kafka
- 亚马逊 OpenSearch 服务
- Amazon Redshift
- `ParallelApplyBufferSize`— 指定在每个缓冲队列中存储的最大记录数，以便并发线程在 CDC 加载期间推送到亚马逊 DocumentDB、Kinesis、Amazon MSK OpenSearch 或 Amazon Redshift 目标终端节点。默认值是 100。最大值为 1000。当 `ParallelApplyThreads` 指定多个线程时，请使用此选项。
- `ParallelApplyQueuesPerThread`— 指定每个线程访问的队列数量，以便从队列中提取数据记录，并在 CDC 期间为 Amazon DocumentDB、Kinesis、Amazon MSK 或终端节点生成批量加载。OpenSearch 默认值是 1。

## 完全加载任务设置

完全加载设置包括：有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

- 要指示如何在完全加载启动时加载目标，请为 `TargetTablePrepMode` 选项指定下列值之一：
  - `DO_NOTHING` – 现有目标表的数据和元数据不受影响。

- `DROP_AND_CREATE` – 删除现有表并在其位置创建一个新表。
- `TRUNCATE_BEFORE_LOAD` – 数据被截断，但不会影响表元数据。
- 要将主键或唯一索引的创建延迟到完全加载后，请将 `CreatePkAfterFullLoad` 选项设置为 `true`。
- 对于完全加载和支持 CDC 的任务，您可以为 `Stop task after full load completes` 设置以下选项：
  - `StopTaskCachedChangesApplied` – 将该选项设置为 `true` 可在完全加载完成并且已应用缓存的更改后停止任务。
  - `StopTaskCachedChangesNotApplied` – 将该选项设置为 `true` 可在应用缓存的更改前停止任务。
- 要指示要并行加载的表的最大数目，请设置 `MaxFullLoadSubTasks` 选项。默认值为 8；最大值为 49。
- 设置 `ParallelLoadThreads` 选项以指示 DMS 在完全加载过程中将使用多少并发线程将数据记录推送到目标端点。默认值为零 (0)。

#### Important

`MaxFullLoadSubTasks` 控制要并行加载的表或表段的数量。`ParallelLoadThreads` 控制迁移任务用于执行并行加载的线程数。这些设置为相乘关系。因此，完全加载任务期间使用的线程总数大约为 `ParallelLoadThreads` 的值乘以 `MaxFullLoadSubTasks` 的值 (`ParallelLoadThreads * MaxFullLoadSubtasks`)。

如果您创建的任务包含大量完全加载子任务和大量并行加载线程，则任务可能会消耗过多的内存并失败。

- 您可以设置在开始满负荷操作之前 AWS DMS 等待交易关闭的秒数。为此，如果在任务开始时事务处于打开状态，请设置 `TransactionConsistencyTimeout` 选项。默认值为 600 (10 分钟)。AWS DMS 即使有未完成的交易，也会在达到超时值后开始满负荷。`full-load-only` 任务不会等待 10 分钟，而是会立即开始。
- 要指示可一起传输的事件的最大数量，请设置 `CommitRate` 选项。默认值是 10000，最大值是 50000。

## Time Travel 任务设置

要记录和调试复制任务，可以使用 Time Travel。在此方法中，您可以使用 Amazon S3 来存储日志，并使用加密密钥对其进行加密。只有具有对您的 Time Travel S3 存储桶的访问权限，才能

使用日期时间过滤器检索 S3 日志，然后根据需要查看、下载以及对日志进行模糊处理。由此，您可以安全地“回到过去”来调查数据库活动。Time Travel 的工作原理与 CloudWatch 日志记录无关。有关 CloudWatch 日志记录的更多信息，请参阅[日志记录任务设置](#)。

您可以在所有 AWS 区域使用支持的 Oracle、Microsoft SQL Server 和 PostgreSQL 源端点以及 AWS DMS 支持的 PostgreSQL 和 MySQL 目标端点的 Time Travel。AWS DMS 只能为完全加载和更改数据捕获 (CDC) 任务以及仅 CDC 的任务启用 Time Travel。要打开 Time Travel 或修改任何现有的 Time Travel 设置，请确保您的复制任务已停止。

Time Travel 设置包括以下 TTSettings 属性：

- **EnableTT** – 如果此选项设置为 `true`，则会为任务开启 Time Travel 日志记录。默认值为 `false`。

类型：布尔值

必需：否

- **EncryptionMode** – 您的 S3 存储桶上用于存储数据和日志的服务器端加密类型。您可以指定 `"SSE_S3"` (默认值) 或 `"SSE_KMS"`。

您可以将 `EncryptionMode` 从 `"SSE_KMS"` 更改为 `"SSE_S3"`，但不能反向更改。

类型：字符串

必需：否

- **ServerSideEncryptionKmsKeyId**— 如果您指定 `"SSE_KMS"` `EncryptionMode`，请提供您的自定义托管 AWS KMS 密钥的 ID。确保您使用的密钥具有启用 AWS Identity and Access Management (IAM) 用户权限并允许使用该密钥的附加策略。

`"SSE_KMS"` 选项仅支持您自己的自定义托管对称 KMS 密钥。

类型：字符串

必需：仅在 `EncryptionMode` 设置为 `"SSE_KMS"` 时

- **ServiceAccessRoleArn** – 服务用于访问 IAM 角色的 Amazon 资源名称 (ARN)。将角色名称设置为 `dms-tt-s3-access-role`。这是一项必需设置，AWS DMS 允许从 S3 存储桶中写入和读取对象。

类型：字符串

**必需：如果已启用 Time Travel**

下面是该角色的策略示例。

```
{
  "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"
      ]
    }
  ]
}
```

下面是该角色的信任策略示例。

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

- **BucketName** – 用于存储 Time Travel 日志的 S3 存储桶的名称。请务必先创建此 S3 存储桶，然后再启用 Time Travel 日志。

类型：字符串

必需：如果已启用 Time Travel

- **BucketFolder** – 一个可选参数，用于在 S3 桶中设置文件夹的名称。如果您指定此参数，DMS 将在路径 `"/BucketName/BucketFolder/taskARN/YYYY/MM/DD/hh"` 中创建 Time Travel 日志。如果您未指定此参数，则 AWS DMS 会将默认路径创建为 `"/BucketName/dms-time-travel-logs/taskARN/YYYY/MM/DD/hh"`。

类型：字符串

必需：否

- **EnableDeletingFromS3OnTaskDelete**— 当此选项设置为 `true`，如果任务已 AWS DMS 删除，则会从 S3 中删除时空旅行日志。默认值为 `false`。

类型：字符串

必需：否

- **EnableRawData** – 当该选项设置为 `true` 时，Time Travel 日志的数据操作语言 (DML) 原始数据将显示在 Time Travel 日志的 `raw_data` 列下。有关详细信息，请参阅[使用 Time Travel 日志](#)。默认值为 `false`。当该选项设置为 `false` 时，仅捕获 DML 的类型。

类型：字符串

必需：否

- **RawDataFormat**— 在 3.5.0 及更高 AWS DMS 版本中，当设置 `EnableRawData` 为 `true` 时，该属性为 Time Travel 日志中 DML 的原始数据指定了一种格式，可以表示为：
  - "TEXT" – CDC 期间捕获的 DML 事件的已解析、可读的列名和值作为 Raw 字段。
  - "HEX" – CDC 期间为 DML 事件捕获的列名和值的原始十六进制值。

该属性适用于 Oracle 和 Microsoft SQL Server 数据库源。

类型：字符串

必需：否

- **OperationsToLog** – 指定要登录 Time Travel 日志的 DML 操作的类型。您可以指定以下几项之一：

- "INSERT"
- "UPDATE"
- "DELETE"
- "COMMIT"
- "ROLLBACK"
- "ALL"

默认值为 "ALL"。

类型：字符串

必需：否

- MaxRecordSize – 指定每行记录的 Time Travel 日志记录的最大大小。使用该属性可以控制极为繁忙表格的 Time Travel 日志的增长。默认值为 64 KB。

类型：整数

必需：否

有关启用和使用 Time Travel 日志的更多信息，请参阅以下主题。

主题

- [为任务启用 Time Travel 日志](#)
- [使用 Time Travel 日志](#)
- [将时空旅行日志 AWS DMS 上传到 S3 的频率](#)

为任务启用 Time Travel 日志

您可以使用前面描述的任务设置为 AWS DMS 任务启用 Time Travel。在启用 Time Travel 之前，请确保您的复制任务已停止。

要打开“时空旅行”，请使用 AWS CLI

1. 创建 DMS 任务配置 JSON 文件并添加 TTSettings 部分，内容如下。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

.

```

.
.
  },
"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. 在相应的任务操作中，使用 `--replication-task-settings` 选项指定此 JSON 文件。例如，以下 CLI 代码片段将此 Time Travel 设置文件指定为 `create-replication-task` 的一部分。

```

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:HNX2BWIIN5ZYFF7F6UFFZVWTDFFSMTNOV2FTXZA \
--replication-instance-arn arn:aws:dms:us-
east-1:112233445566:rep:ERLHG2UA52EEJJKFYNYWRPCG6T7EPUAB5AWBUJQ \
--migration-type full-load-and-cdc --table-mappings 'file:///FilePath/
mappings.json' \
--replication-task-settings 'file:///FilePath/task-settings-tt-enabled.json' \
--replication-task-identifier test-task
.
.
.

```



其中，此 Time Travel 设置文件的名称是 `task-settings-tt-enabled.json`。

同样，您可以将此文件指定为 `modify-replication-task` 操作的一部分。

请注意，对任务操作的 Time Travel 日志的特殊处理如下所示：

- `start-replication-task` – 运行复制任务时，如果无法访问用于 Time Travel 的 S3 存储桶，该任务将被标记为 FAILED。
- `stop-replication-task`— 任务停止后，AWS DMS 立即将复制实例当前可用的所有时空旅行日志推送到用于时空旅行的 S3 存储桶。

当复制任务运行时，您可以将 `EncryptionMode` 值从 "SSE\_KMS" 更改为 "SSE\_S3"，但不能反向更改。

如果进行中任务的 Time Travel 日志大小超过 1 GB，则 DMS 会在达到该大小后的五分钟内将日志推送到 S3。任务运行后，如果 S3 存储桶或 KMS 密钥变得无法访问，DMS 将停止向该存储桶推送日志。如果您发现您的日志未被推送到 S3 存储桶，请检查您的 S3 和 AWS KMS 权限。有关 DMS 将这些日志推送到 S3 的频率的更多详细信息，请参阅[将时空旅行日志 AWS DMS 上传到 S3 的频率](#)。

要从控制台中为现有任务启用 Time Travel，请使用任务设置下的 JSON 编辑器选项添加一个 `TTSettings` 部分。

## 使用 Time Travel 日志

Time Travel 日志文件是逗号分隔值 ( CSV ) 文件，包含以下字段。

```
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

```

将 Time Travel 日志推送到 S3 后，可使用 Amazon Athena 等工具直接访问和查询这些日志。或者，您可以像从 S3 下载任何文件一样下载日志。

以下示例显示一个 Time Travel 日志，其中记录了名为 mytable 的表的事务。为了便于阅读，添加了以下日志的行尾。

```

"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'"

```

## 将时空旅行日志 AWS DMS 上传到 S3 的频率

为了最大限度地减少复制实例的存储使用量，请定期从中 AWS DMS 卸载 Time Travel 日志。

在以下情况下，Time Travel 日志会被推送到您的 Amazon S3 存储桶：

- 如果当前日志大小超过 1 GB，则在五分钟内将日志 AWS DMS 上传到 S3。因此，AWS KMS 对于每个正在运行的任务，每小时最多 AWS DMS 可以向 S3 调用 12 个调用。
- AWS DMS 无论日志大小如何，每小时都会将日志上传到 S3。
- 任务停止后，AWS DMS 立即将时空旅行日志上传到 S3。

## 日志记录任务设置

在迁移过程中 CloudWatch，日志使用 Amazon 来记录信息。使用日志记录任务设置，您可指定要记录的组件活动以及将写入日志的信息量。日志记录任务设置写入到 JSON 文件。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

您可以通过多种方式开启 CloudWatch 日志记录。创建迁移任务 AWS Management Console 时，可以选择上的 EnableLogging 选项。或者，您可以在使用 AWS DMS API 创建任务 true 时

将该 EnableLogging 选项设置为。也可以任务设置的日志记录部分的 JSON 形式指定 "EnableLogging": true。

如果设置 EnableLogging 为 true，则按如下方式 AWS DMS 分配 CloudWatch 群组名称和直播名称。您无法直接设置这些值。

- CloudWatchLogGroup: dms-tasks-<REPLICATION\_INSTANCE\_IDENTIFIER>
- CloudWatchLogStream: dms-task-<REPLICATION\_TASK\_EXTERNAL\_RESOURCE\_ID>

<REPLICATION\_INSTANCE\_IDENTIFIER> 是复制实例的标识符。<REPLICATION\_TASK\_EXTERNAL\_RESOURCE\_ID> 是任务 ARN 的 <resourcename> 部分的值。有关如何 AWS DMS 生成资源 ARN 的信息，请参阅[为其构造亚马逊资源名称 \(ARN\) AWS DMS](#)。

CloudWatch 与 AWS Identity and Access Management (IAM) 集成，您可以指定 AWS 账户中的用户可以执行哪些 CloudWatch 操作。有关使用 IAM 的更多信息 CloudWatch，请参阅《亚马逊 CloudWatch 用户指南》中的“[亚马逊身份 CloudWatch 和访问管理](#)”和“[记录亚马逊 CloudWatch API 调用](#)”。

要删除任务日志，可以在任务设置日志记录部分的 JSON 中将 DeleteTaskLogs 设置为 true。

您可以为以下事件类型指定日志记录：

- FILE\_FACTORY – 文件工厂管理用于批量应用和批量加载的文件，并管理 Amazon S3 端点。
- METADATA\_MANAGER – 元数据管理器在复制期间管理源和目标元数据、分区和表状态。
- SORTER – SORTER 接收来自 SOURCE\_CAPTURE 进程的传入事件。这些事件在事务中进行批处理，然后传送给 TARGET\_APPLY 服务组件。如果 SOURCE\_CAPTURE 进程生成事件的速度超过 TARGET\_APPLY 组件消耗事件的速度，则 SORTER 组件会将积压的事件缓存到磁盘或交换文件中。缓存的事件是复制实例中存储空间不足的常见原因。

SORTER 服务组件管理缓存的事件、收集 CDC 统计数据并报告任务延迟。

- SOURCE\_CAPTURE – 从源数据库或服务中捕获持续复制 (CDC) 数据，并传送到 SORTER 服务组件。
- SOURCE\_UNLOAD – 在完全加载期间从源数据库或服务中卸载数据。
- TABLES\_MANAGER – 表格管理器跟踪捕获的表、管理表迁移顺序并收集表统计信息。
- TARGET\_APPLY – 数据和数据定义语言 (DDL) 语句将应用于目标数据库。
- TARGET\_LOAD – 数据已加载到目标数据库中。

- TASK\_MANAGER – 任务管理器管理进行中任务，并将任务分解为子任务以进行并行数据处理。
- TRANSFORMATION – 表映射转换事件。有关更多信息，请参阅 [使用表映射指定任务设置](#)。
- VALIDATOR/ VALIDATOR\_EXT – VALIDATOR 服务组件验证数据是否已准确地从源迁移到目标。有关更多信息，请参阅 [数据验证](#)。

使用 `LOGGER_SEVERITY_DETAILED_DEBUG` 日志严重性级别时，以下日志组件会生成大量日志：

- COMMON
- ADDONS
- DATA\_STRUCTURE
- COMMUNICATION
- FILE\_TRANSFER
- FILE\_FACTORY

在故障排除期间，这些组件很少需要 `DEFAULT` 以外的其他日志级别。除非 Su AWS pport 特别要求，否则我们不建议更改这些组件的日志级别。 `DEFAULT`

指定上述任一操作后，您可以指定记录的信息量，如以下列表所示。

严重性级别按信息的最低级别到最高级别排列。较高级别始终包含较低级别的信息。

- `LOGGER_SEVERITY_ERROR` – 将错误消息写入日志。
- `LOGGER_SEVERITY_WARNING` – 将警告和错误消息写入日志。
- `LOGGER_SEVERITY_INFO` – 将信息性消息、警告和错误消息写入日志。
- `LOGGER_SEVERITY_DEFAULT` – 将信息性消息、警告和错误消息写入日志。
- `LOGGER_SEVERITY_DEBUG` – 将调试消息、信息性消息、警告和错误消息写入日志。
- `LOGGER_SEVERITY_DETAILED_DEBUG` – 将所有信息写入日志。

以下 JSON 示例显示了用于记录所有操作和严重性级别的任务设置。

```
...
  "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
},
...

```

## 控制表任务设置

控制表提供有关 AWS DMS 任务的信息。它们还提供有用的统计数据，可用于计划和管  
理当前迁移任务和将来任务。您可以在 JSON 文件中应用这些任务设置，也可以在 AWS  
DMS 控制台的“创建任务”页面上选择“高级设置”。将始终在数据库目标上创建应用异常表

( `dmslogs.aws_dms_apply_exceptions` )。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

AWS DMS 仅在 Full Load + CDC 或仅限 CDC 任务期间创建控制表，而不能在“仅限满载”任务期间创建控制表。

对于完全加载和 CDC ( 迁移现有数据并复制持续更改 ) 以及仅 CDC ( 仅复制数据更改 ) 任务，您还可以创建其他表，包括以下表：

- 复制状态 ( `dmslogs.aws_dms_status` ) – 此表提供有关当前任务的详细信息。其中包括任务状态、任务使用的内存量以及尚未应用于目标的更改的数量。此表还给出了源数据库中当前 AWS DMS 正在读取的位置。它还指示任务是完全加载阶段还是更改数据捕获 ( CDC )。
- 暂停的表 ( `dmslogs.aws_dms_suspended_tables` ) – 该表提供一系列暂停的表及其暂停原因。
- 复制历史记录 ( `dmslogs.aws_dms_history` ) – 此表提供有关复制历史记录的信息。此信息包括任务期间处理的记录的数量、CDC 任务结束时的延迟以及其他统计数据。

应用例外表 ( `dmslogs.aws_dms_apply_exceptions` ) 包含以下参数。

列	类型	描述
TASK_NAME	nvchar	AWS DMS 任务的资源 ID。资源 ID 可以在任务 ARN 中找到。
TABLE_OWNER	nvchar	表所有者。
TABLE_NAME	nvchar	表名称。
ERROR_TIME	时间戳	异常 (错误) 的出现时间。
STATEMENT	nvchar	出错时运行的语句。
错误	nvchar	错误名称和描述。

复制状态表 ( `dmslogs.aws_dms_status` ) 包含任务和目标数据库的当前状态。它具有以下设置。

列	类型	描述
SERVER_NAME	nvchar	复制任务运行于的计算机的名称。

列	类型	描述
TASK_NAME	nvarchar	AWS DMS 任务的资源 ID。资源 ID 可以在任务 ARN 中找到。
TASK_STATUS	varchar	<p>下列值之一：</p> <ul style="list-style-type: none"> <li>• FULL LOAD</li> <li>• CHANGE PROCESSING (CDC)</li> <li>• 未运行</li> </ul> <p>只要至少一个表处于完全加载状态，任务状态就会设置为 FULL LOAD。加载所有表后，任务状态将变为 CHANGE PROCESSING (如果已启用 CDC)。在开始任务之前或任务完成之后，任务设置为“未运行”。</p>
STATUS_TIME	时间戳	任务状态的时间戳。
PENDING_CHANGES	int	在源数据库中提交并缓存在复制实例的内存和磁盘中的更改记录的数量。
DISK_SWAP_SIZE	int	旧的或卸载的事务所占用的磁盘空间量。
TASK_MEMORY	int	当前使用的内存 (以 MB 为单位)。
SOURCE_CURRENT_POSITION	varchar	源数据库中当前 AWS DMS 正在读取数据的位置。
SOURCE_CURRENT_TIMESTAMP	时间戳	源数据库中当前 AWS DMS 正在读取的时间戳。

列	类型	描述
SOURCE_TAIL_POSITION	varchar	未提交的最早开始事务的位置。此值是可在不丢失任何更改的情况下恢复到的最新位置。
SOURCE_TAIL_TIMESTAMP	时间戳	未提交的最早开始事务的时间戳。此值是可在不丢失任何更改的情况下恢复到的最新时间戳。
SOURCE_TIMESTAMP_APPLIED	时间戳	最后一个事务提交的时间戳。在批量应用过程中，此值是批处理中最后一个事务的提交时间戳。

暂停表 ( `dmslogs.aws_dms_suspended_tables` ) 包含以下参数。

列	类型	描述
SERVER_NAME	nvarchar	复制任务运行于的计算机的名称。
TASK_NAME	nvarchar	AWS DMS 任务的名称
TABLE_OWNER	nvarchar	表所有者。
TABLE_NAME	nvarchar	表名称。
SUSPEND_REASON	nvarchar	暂停的原因。
SUSPEND_TIMESTAMP	时间戳	暂停发生的时间。

复制历史记录表 ( `dmslogs.aws_dms_history` ) 包含以下参数。

列	类型	描述
SERVER_NAME	nvarchar	复制任务运行于的计算机的名称。



列	类型	描述
TASK_NAME	nvarchar	AWS DMS 任务的资源 ID。资源 ID 可以在任务 ARN 中找到。
TIMESLOT_TYPE	varchar	下列值之一： <ul style="list-style-type: none"> <li>FULL LOAD</li> <li>CHANGE PROCESSING (CDC)</li> </ul> <p>如果任务正在运行完全加载和 CDC，两条历史记录将写入时段。</p>
TIMESLOT	时间戳	时段的结束时间戳。
TIMESLOT_DURATION	int	时段的持续时间（以分钟为单位）。
TIMESLOT_LATENCY	int	时段结束时的目标延迟（以秒为单位）。此值仅适用于 CDC 时段。
RECORDS	int	时段内处理的记录数。
TIMESLOT_VOLUME	int	已处理的数据量（以 MB 为单位）。

验证失败表 ( `awsdms_validation_failures_v1` ) 包含任务的所有数据验证失败。有关更多信息，请参阅[数据验证故障排除](#)。

其他控制表设置包括：

- `HistoryTimeslotInMinutes` – 使用该选项指示复制历史记录表中每个时段的长度。默认值为 5 分钟。
- `ControlSchema`— 使用此选项来指明 AWS DMS 目标控制表的数据库架构名称。如果未为选项输入任何信息，则表将复制到数据库中的默认位置，如下所示：
  - PostgreSQL，公共
  - Oracle，目标架构

- Microsoft SQL Server，目标数据库中的 dbo
- MySQL，awsdms\_control
- MariaDB，awsdms\_control
- Amazon Redshift，公共
- DynamoDB，创建作为数据库中的单独表
- IBM Db2 LUW，awsdms\_control

## 流缓冲区任务设置

您可以使用来设置流缓冲区设置 AWS CLI，包括以下内容。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

- `StreamBufferCount` – 使用该选项指定迁移任务的数据流缓冲区数。默认流缓冲区数为 3。增大该设置的值可能会加快数据提取速度。但是，此性能提高在很大程度上依赖于迁移环境，包括源系统和复制服务器的实例类。对于大多数情况，默认值已足够。
- `StreamBufferSizeInMB` – 使用该选项指示每个数据流缓冲区的最大大小。默认大小为 8 MB。在处理非常大的 LOB 时，可能需要增大此选项的值。如果日志文件中包含一条指示流缓冲区大小不足的消息，则也可能需要增大此值。计算此选项的大小时，您可以使用以下方程式：
$$[\text{Max LOB size (or LOB chunk size)}] * [\text{number of LOB columns}] * [\text{number of stream buffers}] * [\text{number of tables loading in parallel per task}(\text{MaxFullLoadSubTasks})] * 3$$
- `CtrlStreamBufferSizeInMB` – 使用该选项设置控制流缓冲区的大小。该值以 MB 为单位，范围介于 1–8 之间。默认值是 5。当处理巨大数量的表（如数万个表）时，您可能需要增大此值。

## 更改处理优化设置

以下设置决定了在更改数据捕获 (CDC) 期间如何 AWS DMS 处理目标表的更改。这些设置中的多个设置取决于目标元数据参数 `BatchApplyEnabled` 的值。有关 `BatchApplyEnabled` 参数的更多信息，请参阅[目标元数据任务设置](#)。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

更改处理优化设置包括：

以下设置仅在目标元数据参数 `BatchApplyEnabled` 设置为 `true` 时应用。

- `BatchApplyPreserveTransaction` – 如果设置为 `true`，则保留事务完整性，并确保批处理包含源中的事务的所有更改。默认值为 `true`。该设置仅适用于 Oracle 目标终端节点。

如果设置为 `false`，则会临时丢失事务完整性以提高性能。不能保证将源中事务的所有更改都应用于单个批处理中的目标。

默认情况下，AWS DMS 处理以事务模式进行更改，这样可以保持事务的完整性。如果您可以承受事务完整性的临时失效，可以改为使用批量优化应用选项。该选项有效分组事务并批量应用，以实现提高效率的目的。使用批量优化的应用选项几乎总是违反引用完整性约束。因此，我们建议您在迁移过程中关闭这些限制，然后在割接过程中再次将其打开。

- `BatchApplyTimeoutMin`— 设置每次应用批量更改之间 AWS DMS 等待的最短时间（以秒为单位）。默认值是 1。
- `BatchApplyTimeoutMax`— 设置每次应用批量更改之间在超时之前 AWS DMS 等待的最大时间（以秒为单位）。默认值为 30。
- `BatchApplyMemoryLimit` – 设置在批量优化应用模式中用于预处理的最大内存量（MB）。默认值是 500。
- `BatchSplitSize` – 设置在单个批次中应用的最大更改数。默认值 0 意味着未应用任何限制。

以下设置仅在目标元数据参数 `BatchApplyEnabled` 设置为 `false` 时应用。

- `MinTransactionSize` – 设置要包含在每个事务中的更改的最小数目。默认值是 1000。
- `CommitTimeout`— 设置在宣布超时之前分批收集交易的最长时间（AWS DMS 以秒为单位）。默认值是 1。

对于双向复制，以下设置仅在目标元数据参数 `BatchApplyEnabled` 设置为 `false` 时应用。

- `LoopbackPreventionSettings` – 这些设置为双向复制中涉及的任何一对任务中每个正在进行的复制任务提供环回防护。环回防护可防止在双向复制的两个方向上应用完全相同的更改，这可能会损坏数据。有关双向复制的更多信息，请参阅[执行双向复制](#)。

AWS DMS 尝试将事务数据保存在内存中，直到事务完全提交给源、目标或两者。但是，超过所分配内存的事务或未在指定时限内提交的事务将写入到磁盘。

以下设置适用于更改处理优化，无论更改处理模式如何。

- `MemoryLimitTotal` – 设置所有事务在写入磁盘前可占用的内存的最大内存（以 MB 为单位）。默认值是 1024。
- `MemoryKeepTime` – 设置每个事务在写入磁盘之前可在内存中保留的最长时间（以秒为单位）。持续时间是从小于 AWS DMS 开始捕获交易的时间开始计算的。默认值是 60。
- `StatementCacheSize` – 设置在向目标应用更改时要存储在服务器上以便稍后执行的预编译语句的最大数目。默认值是 50。最大值为 200。

以下示例演示了处理更改处理调优的任务设置是如何显示在任务设置 JSON 文件中的：

```
"ChangeProcessingTuning": {
  "BatchApplyPreserveTransaction": true,
  "BatchApplyTimeoutMin": 1,
  "BatchApplyTimeoutMax": 30,
  "BatchApplyMemoryLimit": 500,
  "BatchSplitSize": 0,
  "MinTransactionSize": 1000,
  "CommitTimeout": 1,
  "MemoryLimitTotal": 1024,
  "MemoryKeepTime": 60,
  "StatementCacheSize": 50
}
```

要控制在执行数据复制任务期间向 Amazon S3 目标写入的频率，您可以配置 `cdcMaxBatchInterval` 和 `cdcMinFileSize` 额外连接属性。这可以提高分析数据的性能，而无需任何额外的开销操作。有关更多信息，请参阅 [使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。

## 数据验证任务设置

您可以确保准确地将您的数据从源迁移到目标。如果为任务启用验证，则在对表执行满载后立即 AWS DMS 开始比较源数据和目标数据。有关任务数据验证、其要求、其数据库支持的范围以及它报告的指标的更多信息，请参阅 [AWS DMS 数据验证](#)。有关如何使用任务配置文件来设置任务设置的信息，请参阅 [任务设置示例](#)。

数据验证设置及其值包括：

- `EnableValidation` – 在设置为 `true` 时，启用数据验证。否则，将为任务禁用验证。默认值为 `False`。

- **ValidationMode** – 控制 DMS 如何根据源表验证目标表中的数据。AWS DMS 提供该设置是为了保证未来的可扩展性。当前，默认且唯一有效的值是 `ROW_LEVEL`。AWS DMS 验证源表和目标表之间的所有行。
- **FailureMaxCount** – 指定在暂停任务的验证之前可验证失败的记录的最大数目。默认值是 10000。如果您希望验证继续而不管验证失败的记录数如何，请将此值设置为大于源中记录数的值。
- **HandleCollationDiff** – 如果将此选项设置为 `true`，则在识别要比较的源记录和目标记录时，验证会考虑 PostgreSQL 和 Microsoft SQL Server 端点中的列排序规则差异。否则，将忽略列排序规则中的任何此类差异以进行验证。列排序规则可以规定行的顺序，这对于数据验证很重要。将 `HandleCollationDiff` 设置为 `true` 可自动解析这些排序规则差异并防止数据验证中出现误报。默认值为 `false`。
- **RecordFailureDelayInMinutes** – 指定报告任何验证失败详细信息之前的延迟（以分钟为单位）。
- **RecordFailureDelayLimitInMinutes** – 指定报告任何验证失败详细信息之前的延迟。通常，AWS DMS 使用任务延迟来识别更改的实际延迟，以使其成为目标来防止误报。此设置会覆盖实际延迟值，并允许您在报告任何验证指标之前设置更高的延迟。默认值是 0。
- **RecordSuspendDelayInMinutes** – 指定根据 `FailureMaxCount` 中设置的错误阈值而暂停验证表之前的延迟时间（以分钟为单位）。
- **SkipLobColumns**— 当此选项设置为 `true`，将 AWS DMS 跳过任务验证中表中所有 LOB 列的数据验证。默认值为 `false`。
- **TableFailureMaxCount** – 指定在暂停表的验证之前可验证失败的、一个表中的最大行数。默认值是 1000。
- **ThreadCount**— 指定验证期间 AWS DMS 使用的执行线程数。每个线程都从源和目标中选择 `not-yet-validated` 数据进行比较和验证。默认值是 5。如果设置 `ThreadCount` 为更高的数字，则 AWS DMS 可以更快地完成验证。但是，AWS DMS 随后将执行更多同步查询，这将消耗源和目标上的更多资源。
- **ValidationOnly** – 当该选项设置为 `true` 时，任务会执行数据验证，而不执行任何数据迁移或复制。默认值为 `false`。创建任务后，您将无法修改 `ValidationOnly` 设置。

必须 `TargetTablePrepMode` 将迁移类型设置为 `DO_NOTHING`（仅限验证的任务的默认值），并将“迁移类型”设置为以下选项之一：

- **满载** - 在 AWS DMS 控制台中将任务迁移类型设置为迁移现有数据。或者，在 AWS DMS API 中将迁移类型设置为满载。
- **CDC** – 在 AWS DMS 控制台中将任务迁移类型设置为仅复制数据更改。或者，在 AWS DMS API 中将迁移类型设置为 CDC。

无论选择哪种迁移类型，在仅限验证的任务中，实际上都不会迁移或复制数据。

有关更多信息，请参阅 [仅验证任务](#)。

#### Important

该 ValidationOnly 设置是不可变的。创建任务后，无法修改该任务的内容。

- ValidationPartialLobSize – 指定是否要对 LOB 列进行部分验证，而不是验证存储在该列中的所有数据。当只迁移一部分 LOB 数据而不是整个 LOB 数据集时，可能会发现这很有用。该值以 KB 为单位。默认值是 0，这意味着 AWS DMS 验证所有 LOB 列数据。例如，"ValidationPartialLobSize": 32 表示 AWS DMS 仅验证源和目标中列数据的前 32KB。
- PartitionSize – 指定要从源和目标读取的记录的批量大小，以便进行比较。默认值是 10000。
- ValidationQueryCdcDelaySeconds – 每次更新 CDC 时，源和目标上验证查询首次延迟的时间长度。当迁移延迟较长时，这可能有助于减少资源争用。仅限验证的任务会自动将此选项设置为 180 秒。默认值是 0。

例如，以下 JSON 支持使用两倍于默认线程数的线程进行数据验证。它还解释了由 PostgreSQL 终端节点中的列排序规则差异导致的记录顺序差异。此外，它还提供验证报告延迟，以考虑处理任何验证失败所需的额外时间。

```
"ValidationSettings": {
  "EnableValidation": true,
  "ThreadCount": 10,
  "HandleCollationDiff": true,
  "RecordFailureDelayLimitInMinutes": 30
}
```

#### Note

对于 Oracle 端点，AWS DMS 使用 DBMS\_CRYPTO 来验证 BLOB。如果 Oracle 端点使用 BLOB，则必须向用于访问 Oracle 端点的用户账户授予 DBMS\_CRYPTO 的 execute 权限。为此，运行以下语句。

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

## 更改处理 DDL 处理的任务设置

以下设置决定了在更改数据捕获 (CDC) 期间如何 AWS DMS 处理目标表的数据定义语言 (DDL) 更改。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

用于处理更改处理 DDL 的任务设置包括以下内容：

- `HandleSourceTableDropped` – 将该选项设置为 `true` 可在删除源表后删除目标表。
- `HandleSourceTableTruncated` – 将该选项设置为 `true` 可在截断源表时截断目标表。
- `HandleSourceTableAltered` – 将该选项设置为 `true` 可在更改源表时更改目标表。

以下示例演示了处理更改处理 DDL 的任务设置是如何显示在任务设置 JSON 文件中的：

```
"ChangeProcessingDdlHandlingPolicy": {  
  "HandleSourceTableDropped": true,  
  "HandleSourceTableTruncated": true,  
  "HandleSourceTableAltered": true  
},
```

### Note

有关特定端点所支持的 DDL 语句的信息，请参阅介绍该端点的主题。

## 字符替换任务设置

您可以指定您的复制任务在目标数据库上对 WSTRING 数据类型为 AWS DMS STRING 或的所有源数据库列执行字符替换。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

您可以为具有来自以下源和目标数据库的终端节点的任意任务配置字符替换：

- 源数据库：

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- SAP Adaptive Server Enterprise ( ASE )
- IBM Db2 LUW
- 目标数据库 :
  - Oracle
  - Microsoft SQL Server
  - MySQL
  - PostgreSQL
  - SAP Adaptive Server Enterprise ( ASE )
  - Amazon Redshift

您可以在任务设置中使用 `CharacterSetSettings` 参数指定字符替换。这些字符替换发生在使用 Unicode 码位值 ( 采用十六进制表示法 ) 指定的字符处。您可以分两个阶段实施替换，在指定了这两个阶段时需按以下顺序进行：

1. 单个字符替换- AWS DMS 可以将源上所选字符的值替换为目标上相应字符的指定替换值。使用 `CharacterSetSettings` 中的 `CharacterReplacements` 数组，选择具有您指定的 Unicode 码位的所有源字符。使用此数组还可以指定目标上对应字符的替换码位。

要选择源上具有指定码位的所有字符，请将 `CharacterReplacements` 数组中 `SourceCharacterCodePoint` 的实例设置为该码位。然后通过在此数组中设置 `TargetCharacterCodePoint` 的对应实例，指定所有等同目标字符的替换码位。要删除目标字符而不是替换它们，请将 `TargetCharacterCodePoint` 的对应实例设置为零 (0)。您可以通过在 `CharacterReplacements` 数组中指定额外的 `SourceCharacterCodePoint` 和 `TargetCharacterCodePoint` 对，替换或删除所需任意数量的不同目标字符值。如果您为 `SourceCharacterCodePoint` 的多个实例指定了相同值，则 `TargetCharacterCodePoint` 的上一个对应设置的值应用到目标。

例如，假设您为 `CharacterReplacements` 指定以下值。

```
"CharacterSetSettings": {
  "CharacterReplacements": [ {
```



```

    "SourceCharCodePoint": 62,
    "TargetCharCodePoint": 61
  }, {
    "SourceCharCodePoint": 42,
    "TargetCharCodePoint": 41
  }
]
}

```

在此示例中，AWS DMS 将目标上源代码点十六进制值 62 的所有字符替换为代码点值 61 的字符。此外，AWS DMS 将目标上源代码点 42 的所有字符替换为代码点值 41 的字符。换言之，AWS DMS 在目标上将字母 'b' 的所有实例替换为字母 'a'。同样，AWS DMS 用字母替换目标 'B' 上字母的所有实例 'A'。

2. 字符集验证和替换-完成任何单个字符替换后，AWS DMS 可以确保所有目标字符在您指定的单个字符集中都有有效的 Unicode 代码点。您可使用 CharacterSetSettings 中的 CharacterSetSupport 来配置此目标字符验证和修改。要指定验证字符集，请将 CharacterSetSupport 中的 CharacterSet 设置为字符集字符串值。（CharacterSet 的可能值见下文。）您可以通过以下方式之一 AWS DMS 修改无效的目标字符：
  - 为所有无效目标字符指定单个替换 Unicode 码位，不论其当前码位如何。要配置此替换码位，请将 CharacterSetSupport 中的 ReplaceWithCharCodePoint 设置为指定值。
  - 通过将 ReplaceWithCharCodePoint 设置为零 (0) 配置删除所有无效目标字符。

例如，假设您为 CharacterSetSupport 指定以下值。

```

"CharacterSetSettings": {
  "CharacterSetSupport": {
    "CharacterSet": "UTF16_PlatformEndian",
    "ReplaceWithCharCodePoint": 0
  }
}

```

在此示例中，AWS DMS 删除目标上发现的字符集中无效的所有 "UTF16\_PlatformEndian" 字符。因此，将删除使用十六进制值 2FB6 指定的任意字符。由于这是一个 4 字节 Unicode 码位而 UTF16 字符集仅接受 2 字节码位，此值无效。

**Note**

复制任务在启动您通过表映射指定的任意全局或表级别转换之前，完成所有指定的字符替换。有关表映射的更多信息，请参阅[使用表映射指定任务设置](#)。

字符替换不支持 LOB 数据类型。这包括 DMS 认为是 LOB 数据类型的任何数据类型。例如，Oracle 中的 Extended 数据类型被视为 LOB。有关源数据类型的更多信息，请参阅下面的[Oracle 的源数据类型](#)。

AWS DMS 支持的值CharacterSet显示在下表中。

UTF-8	ibm-860_P100-1995	ibm-280_P100-1995
UTF-16	ibm-861_P100-1995	ibm-284_P100-1995
UTF-16BE	ibm-862_P100-1995	ibm-285_P100-1995
UTF-16LE	ibm-863_P100-1995	ibm-290_P100-1995
UTF-32	ibm-864_X110-1999	ibm-297_P100-1995
UTF-32BE	ibm-865_P100-1995	ibm-420_X120-1999
UTF-32LE	ibm-866_P100-1995	ibm-424_P100-1995
UTF16_PlatformEndian	ibm-867_P100-1998	ibm-500_P100-1995
UTF16_OppositeEndian	ibm-868_P100-1995	ibm-803_P100-1999
UTF32_PlatformEndian	ibm-869_P100-1995	ibm-838_P100-1995
UTF32_OppositeEndian	ibm-878_P100-1996	ibm-870_P100-1995
UTF-16BE,version=1	ibm-901_P100-1999	ibm-871_P100-1995
UTF-16LE,version=1	ibm-902_P100-1999	ibm-875_P100-1995
UTF-16,version=1	ibm-922_P100-1999	ibm-918_P100-1995
UTF-16,version=2	ibm-1168_P100-2002	ibm-930_P120-1999

UTF-7	ibm-4909_P100-1999	ibm-933_P110-1995
IMAP-mailbox-name	ibm-5346_P100-1998	ibm-935_P110-1999
SCSU	ibm-5347_P100-1998	ibm-937_P110-1999
BOCU-1	ibm-5348_P100-1997	ibm-939_P120-1999
CESU-8	ibm-5349_P100-1998	ibm-1025_P100-1995
ISO-8859-1	ibm-5350_P100-1998	ibm-1026_P100-1995
US-ASCII	ibm-9447_P100-2002	ibm-1047_P100-1995
gb18030	ibm-9448_X100-2005	ibm-1097_P100-1995
ibm-912_P100-1995	ibm-9449_P100-2002	ibm-1112_P100-1995
ibm-913_P100-2000	ibm-5354_P100-1998	ibm-1122_P100-1999
ibm-914_P100-1995	ibm-1250_P100-1995	ibm-1123_P100-1995
ibm-915_P100-1995	ibm-1251_P100-1995	ibm-1130_P100-1997
ibm-1089_P100-1995	ibm-1252_P100-2000	ibm-1132_P100-1998
ibm-9005_X110-2007	ibm-1253_P100-1995	ibm-1137_P100-1999
ibm-813_P100-1995	ibm-1254_P100-1995	ibm-4517_P100-2005
ibm-5012_P100-1999	ibm-1255_P100-1995	ibm-1140_P100-1997
ibm-916_P100-1995	ibm-5351_P100-1998	ibm-1141_P100-1997
ibm-920_P100-1995	ibm-1256_P110-1997	ibm-1142_P100-1997
iso-8859_10-1998	ibm-5352_P100-1998	ibm-1143_P100-1997
iso-8859_11-2001	ibm-1257_P100-1995	ibm-1144_P100-1997
ibm-921_P100-1995	ibm-5353_P100-1998	ibm-1145_P100-1997

iso-8859_14-1998	ibm-1258_P100-1997	ibm-1146_P100-1997
ibm-923_P100-1998	macos-0_2-10.2	ibm-1147_P100-1997
ibm-942_P12A-1999	macos-6_2-10.4	ibm-1148_P100-1997
ibm-943_P15A-2003	macos-7_3-10.2	ibm-1149_P100-1997
ibm-943_P130-1999	macos-29-10.2	ibm-1153_P100-1999
ibm-33722_P12A_P12A-2009_U2	macos-35-10.2	ibm-1154_P100-1999
ibm-33722_P120-1999	ibm-1051_P100-1995	ibm-1155_P100-1999
ibm-954_P101-2007	ibm-1276_P100-1995	ibm-1156_P100-1999
euc-jp-2007	ibm-1006_P100-1995	ibm-1157_P100-1999
ibm-1373_P100-2002	ibm-1098_P100-1995	ibm-1158_P100-1999
windows-950-2000	ibm-1124_P100-1996	ibm-1160_P100-1999
ibm-950_P110-1999	ibm-1125_P100-1997	ibm-1164_P100-1999
ibm-1375_P100-2008	ibm-1129_P100-1997	ibm-1364_P110-2007
ibm-5471_P100-2006	ibm-1131_P100-1997	ibm-1371_P100-1999
ibm-1386_P100-2001	ibm-1133_P100-1997	ibm-1388_P103-2001
windows-936-2000	ISO_2022,locale=ja,version=0	ibm-1390_P110-2003
ibm-1383_P110-1999	ISO_2022,locale=ja,version=1	ibm-1399_P110-2003
ibm-5478_P100-1995	ISO_2022,locale=ja,version=2	ibm-5123_P100-1999

euc-tw-2014	ISO_2022,locale=ja ,version=3	ibm-8482_P100-1999
ibm-964_P110-1999	ISO_2022,locale=ja ,version=4	ibm-16684_P110-2003
ibm-949_P110-1999	ISO_2022,locale=ko ,version=0	ibm-4899_P100-1998
ibm-949_P11A-1999	ISO_2022,locale=ko ,version=1	ibm-4971_P100-1999
ibm-970_P110_P110- 2006_U2	ISO_2022,locale=zh ,version=0	ibm-9067_X100-2005
ibm-971_P100-1995	ISO_2022,locale=zh ,version=1	ibm-12712_P100-1998
ibm-1363_P11B-1998	ISO_2022,locale=zh ,version=2	ibm-16804_X110-1999
ibm-1363_P110-1997	HZ	ibm-37_P100-1995,s waplfnl
windows-949-2000	x11-compound-text	ibm-1047_P100-1995 ,waplfnl
windows-874-2000	ISCII,version=0	ibm-1140_P100-1997 ,waplfnl
ibm-874_P100-1995	ISCII,version=1	ibm-1141_P100-1997 ,waplfnl
ibm-1162_P100-1999	ISCII,version=2	ibm-1142_P100-1997 ,waplfnl
ibm-437_P100-1995	ISCII,version=3	ibm-1143_P100-1997 ,waplfnl

ibm-720_P100-1997	ISCSII,version=4	ibm-1144_P100-1997 ,swaplfnl
ibm-737_P100-1997	ISCSII,version=5	ibm-1145_P100-1997 ,swaplfnl
ibm-775_P100-1996	ISCSII,version=6	ibm-1146_P100-1997 ,swaplfnl
ibm-850_P100-1995	ISCSII,version=7	ibm-1147_P100-1997 ,swaplfnl
ibm-851_P100-1995	ISCSII,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_P100-199 8,swaplfnl
ibm-857_P100-1995	ibm-277_P100-1995	ibm-16804_X110-199 9,swaplfnl
ibm-858_P100-1997	ibm-278_P100-1995	ebcdic-xml-us

## 之前映像任务设置

在将 CDC 更新写入数据流目标（如 Kinesis 或 Apache Kafka）时，可以在更新进行更改之前查看源数据库行的原始值。为此，请根据源数据库引擎提供的数据 AWS DMS 填充更新事件之前的图像。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

为此，可使用 `BeforeImageSettings` 参数，它使用从源数据库系统收集的值向每个更新操作添加一个新的 JSON 属性。

确保仅对完全加载及 CDC 任务或仅 CDC 的任务应用 BeforeImageSettings。完全加载及 CDC 任务可迁移现有数据并复制持续更改。仅 CDC 任务仅仅复制数据更改。

不将 BeforeImageSettings 应用于仅完全加载的任务。

BeforeImageSettings 的可用选项如下所示：

- EnableBeforeImage – 设置为 true 时，在映像前开启。默认值为 false。
- FieldName – 为新的 JSON 属性指定名称。当 EnableBeforeImage 为 true 时，FieldName 是必填项且不能为空。
- ColumnFilter – 指定要使用之前映像添加的列。要仅添加属于表主键一部分的列，请使用默认值 pk-only。要添加具有之前映像值的任何列，请使用 all。请注意，之前的图像不支持大型二进制对象 (LOB) 数据类型，例如 CLOB 和 BLOB。

下面的示例演示 BeforeImageSettings 的用法。

```
"BeforeImageSettings": {
  "EnableBeforeImage": true,
  "FieldName": "before-image",
  "ColumnFilter": "pk-only"
}
```

有关 Kinesis 的之前映像设置 (包括其他表映射设置) 的信息，请参阅[使用之前映像查看 Kinesis 数据流 \(作为目标\) 的 CDC 行的原始值](#)。

有关 Kafka 的之前映像设置 (包括其他表映射设置) 的信息，请参阅[使用之前映像查看 Apache Kafka \(作为目标\) 的 CDC 行的原始值](#)。

## 错误处理任务设置

可以使用以下设置来设置复制任务的错误处理行为。有关如何使用任务配置文件来设置任务设置的信息，请参阅[任务设置示例](#)。

- DataErrorPolicy— 确定在记录级别出现与数据处理相关的错误时，AWS DMS 采取的操作。数据处理错误的一些示例包括转换错误、变换中的错误和不正确的数据。默认值为 LOG\_ERROR。
  - IGNORE\_RECORD – 任务继续，并且将忽略该记录的数据。DataErrorEscalationCount 属性的错误计数器将递增。因此，如果您为表设置错误数限制，则此错误将计入限制。
  - LOG\_ERROR – 任务继续，并且错误将写入任务日志。
  - SUSPEND\_TABLE – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。

- `STOP_TASK` – 任务停止，并且需要手动干预。
- `DataTruncationErrorPolicy` – 决定在数据被截断时，AWS DMS 执行的操作。默认值为 `LOG_ERROR`。
  - `IGNORE_RECORD` – 任务继续，并且将忽略该记录的数据。`DataErrorEscalationCount` 属性的错误计数器将递增。因此，如果您为表设置错误数限制，则此错误将计入限制。
  - `LOG_ERROR` – 任务继续，并且错误将写入任务日志。
  - `SUSPEND_TABLE` – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。
  - `STOP_TASK` – 任务停止，并且需要手动干预。
- `DataErrorEscalationPolicy` – 决定在达到最大错误数（在 `DataErrorEscalationCount` 参数中设置）时，AWS DMS 执行的操作。默认值为 `SUSPEND_TABLE`。
  - `SUSPEND_TABLE` – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。
  - `STOP_TASK` – 任务停止，并且需要手动干预。
- `DataErrorEscalationCount` – 设置特定记录的数据可出现的错误的最大数目。当达到此数目时，将根据 `DataErrorEscalationPolicy` 中设置的策略来处理包含错误记录的表的数据。默认值是 0。
- `EventErrorPolicy` – 确定 AWS DMS 在发送与任务相关的事件时发生错误时采取的操作。可能的值包括
  - `IGNORE` – 任务继续，与该事件关联的所有数据都将被忽略。
  - `STOP_TASK` – 任务停止，并且需要手动干预。
- `TableErrorPolicy` – 决定在处理特定表的数据或元数据时出错的情况下，AWS DMS 执行的操作。此错误仅适用于常规表数据，而不是与特定记录相关的错误。默认值为 `SUSPEND_TABLE`。
  - `SUSPEND_TABLE` – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。
  - `STOP_TASK` – 任务停止，并且需要手动干预。
- `TableErrorEscalationPolicy` – 决定在达到最大错误数（使用 `TableErrorEscalationCount` 参数设置）时，AWS DMS 执行的操作。默认用户设置为 `STOP_TASK`，其中任务将停止，并且需要手动干预。
- `TableErrorEscalationCount` – 特定表的常规数据或元数据可出现的错误的最大数目。当达到此数目时，将根据 `TableErrorEscalationPolicy` 中设置的策略处理表的数据。默认值是 0。
- `RecoverableErrorCount` – 在出现环境错误时，可尝试重新启动任务的最大次数。在系统尝试重新启动任务指定次数时，任务将停止，并且需要手动干预。默认值为 -1，表示 AWS DMS 尝试无限期地重新启动任务。当您将此值设置为 -1 时，DMS 尝试的重试次数会根据返回的错误类型而有所不同，如下所示：



- 运行状态，可恢复错误：如果出现可恢复的错误，例如连接中断或目标应用失败，DMS 将重试任务九次。
- 启动状态，可恢复错误：DMS 重试任务六次。
- 运行状态，DMS 处理的致命错误：DMS 重试任务六次。
- 运行状态，DMS 未处理致命错误：DMS 不会重试任务。

将此值设置为 0 将永不尝试重新启动任务。

我们建议您将 `RecoverableErrorCount` 和 `RecoverableErrorInterval` 设置为值，以便在足够的时间间隔内有足够的重试次数，以使您的 DMS 任务能够正常恢复。如果发生致命错误，则在大多数情况下，DMS 会停止尝试重启。

- `RecoverableErrorInterval`— AWS DMS 在两次尝试重启任务之间等待的秒数。默认值是 5。
- `RecoverableErrorThrottling` – 启用后，尝试重启任务的间隔将根据 `RecoverableErrorInterval` 的值连续增加。例如，如果 `RecoverableErrorInterval` 设置为 5 秒，则下一次重试将在 10 秒后进行，然后是 20 秒，然后是 40 秒，依此类推。默认值为 `true`。
- `RecoverableErrorThrottlingMax`— 如果 `RecoverableErrorThrottling` 启用，AWS DMS 在尝试重启任务之间等待的最大秒数。默认值是 1800。
- `RecoverableErrorStopRetryAfterThrottlingMax`— 如果设置为 `true`，则在达到恢复尝试之间 AWS DMS 等待的最大秒数后停止重新启动任务。`RecoverableErrorThrottlingMax`
- `ApplyErrorDeletePolicy` – 决定在与 DELETE 操作发生冲突时，AWS DMS 执行的操作。默认值为 `IGNORE_RECORD`。可能的值包括：
  - `IGNORE_RECORD` – 任务继续，并且将忽略该记录的数据。`ApplyErrorEscalationCount` 属性的错误计数器将递增。因此，如果您为表设置错误数限制，则此错误将计入限制。
  - `LOG_ERROR` – 任务继续，并且错误将写入任务日志。
  - `SUSPEND_TABLE` – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。
  - `STOP_TASK` – 任务停止，并且需要手动干预。
- `ApplyErrorInsertPolicy` – 决定在与 INSERT 操作发生冲突时，AWS DMS 执行的操作。默认值为 `LOG_ERROR`。可能的值包括：
  - `IGNORE_RECORD` – 任务继续，并且将忽略该记录的数据。`ApplyErrorEscalationCount` 属性的错误计数器将递增。因此，如果您为表设置错误数限制，则此错误将计入限制。
  - `LOG_ERROR` – 任务继续，并且错误将写入任务日志。
  - `SUSPEND_TABLE` – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。

- `STOP_TASK` – 任务停止，并且需要手动干预。
- `INSERT_RECORD` – 如果存在带与插入的源记录相同的主键的现有目标记录，则更新该目标记录。
- `ApplyErrorUpdatePolicy` – 决定在与 `UPDATE` 操作发生缺少数据的冲突时，AWS DMS 执行的操作。默认值为 `LOG_ERROR`。可能的值包括：
  - `IGNORE_RECORD` – 任务继续，并且将忽略该记录的数据。`ApplyErrorEscalationCount` 属性的错误计数器将递增。因此，如果您为表设置错误数限制，则此错误将计入限制。
  - `LOG_ERROR` – 任务继续，并且错误将写入任务日志。
  - `SUSPEND_TABLE` – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。
  - `STOP_TASK` – 任务停止，并且需要手动干预。
  - `UPDATE_RECORD` – 如果缺少目标记录，则会将缺失的目标记录插入到目标表中。AWS DMS 完全禁用对任务的 LOB 列支持。选择该选项要求在 Oracle 作为源数据库时为所有源表列启用完整补充日志记录。
- `ApplyErrorEscalationPolicy` – 确定当达到最大错误数（使用 `ApplyErrorEscalationCount` 参数设置）时，AWS DMS 将采取什么操作。默认值为 `LOG_ERROR`：
  - `LOG_ERROR` – 任务继续，并且错误将写入任务日志。
  - `SUSPEND_TABLE` – 任务继续，但带错误记录的表中的数据将移至错误状态，并且不会复制数据。
  - `STOP_TASK` – 任务停止，并且需要手动干预。
- `ApplyErrorEscalationCount` – 该选项设置在更改处理操作期间，特定表可发生的 `APPLY` 冲突的最大次数。达到此数目时，将根据 `ApplyErrorEscalationPolicy` 参数中设置的策略处理表数据。默认值是 0。
- `ApplyErrorFailOnTruncationDdl` – 如果将该选项设置为 `true`，则会导致在 CDC 期间对任何跟踪的表执行截断时任务失败。默认值为 `false`。

此方式不适用于 PostgreSQL 11.x 或以下版本，以及不复制 DDL 表截断的任何其他源端点。

- `FailOnNoTablesCaptured` – 如果将该选项设置为 `true`，则会导致在任务启动时为任务定义的表映射找不到表时任务失败。默认值为 `false`。
- `FailOnTransactionConsistencyBreached` – 该选项适用于将 Oracle 作为 CDC 源的任务。默认值为 `false`。如果将该选项设置为 `true`，则会导致当事务的打开时间超出指定超时并可能删除时任务失败。

当 CDC 任务从 Oracle 开始时，在启动 CDC 之前，AWS DMS 等待最早的未平仓交易结束的有限时间。如果最早的未结交易直到达到超时后才关闭，则在大多数情况下，AWS DMS 会启动 CDC，忽略该交易。如果将该选项设置为 `true`，则任务将失败。

- `FullLoadIgnoreConflicts`— 将此选项设置为 `true` 在应用缓存事件时 AWS DMS 忽略“受影响的零行”和“重复”错误。如果设置为 `false`，则 AWS DMS 报告所有错误而不是忽略错误。默认值为 `true`。

请注意，`STL_LOAD_ERRORS` 中报告了 Redshift 作为目标的表加载错误。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 [STL\\_LOAD\\_ERRORS](#)。

## 保存任务设置

可以将任务设置另存为 JSON 文件（如果您需要将这些设置重用于另一个任务）。您可以在任务的概述详细信息部分下找到要复制到 JSON 文件的任务设置。

### Note

在将任务设置重复用于其他任务时，请移除任何 `CloudWatchLogGroup` 和 `CloudWatchLogStream` 属性。否则，将出现以下错误：系统错误消息：任务设置 `CloudWatchLogGroup` 或 `CloudWatchLogStream` 无法在创建时进行设置。

例如，以下 JSON 文件包含为任务保存的设置。

```
{
  "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
}
```

## 在任务中为源数据库设置 LOB 支持 AWS DMS

有时，大型二进制对象 (LOB) 难以在系统之间迁移。AWS DMS 提供了大量选项来帮助优化 LOB 列。要了解哪些数据类型以及何时被视为 LOB AWS DMS，请参阅 AWS DMS 文档。

在将数据从一个数据库迁移到另一个数据库时，您可能有机会重新考虑 LOB 的存储方式，特别是异构迁移。如果您需要这样做，则无需迁移 LOB 数据。

如果决定包含 LOB，您可以确定其他 LOB 设置：

- LOB 模式确定 LOB 的处理方式：
  - 完整 LOB 模式 — 在完整 LOB 模式下，无论大小如何，都会将所有 LOB 从源 AWS DMS 迁移到目标。在此配置中，AWS DMS 没有关于预期的最大 LOB 大小的信息。因此，逐个迁移 LOB，一次迁移一个。完整 LOB 模式下的速度非常慢。
  - 受限 LOB 模式 – 在受限 LOB 模式下，您设置 DMS 接受的最大 LOB 大小。该选项允许 DMS 预分配内存并批量加载 LOB 数据。将截断超过最大 LOB 大小的 LOB，并将警告发布到日志文件中。在受限 LOB 模式下，您可以获得比完整 LOB 模式高得多的性能。如果可能，建议您使用受限 LOB 模式。建议的最大值为 102400 KB ( 100 MB )。

**Note**

使用值大于 63KB 的最大 LOB 大小 ( K ) 选项会影响配置为在受限 LOB 模式下运行的完全加载的性能。在完全加载期间，DMS 通过将最大 LOB 大小 ( K ) 值乘以提交速率来分配内存，乘积乘以 LOB 列数。当 DMS 无法预先分配该内存时，DMS 就会开始消耗 SWAP 内存，这会影响完全加载的性能。因此，如果您在使用受限 LOB 模式时遇到性能问题，请考虑降低提交速率，直到达到可接受的性能水平。了解表的 LOB 分布后，也可以考虑对支持的端点使用内联 LOB 模式。

要验证受限 LOB 大小，必须将 ValidationPartialLobSize 设置为与 LobMaxSize ( K ) 相同的值。

- 内联 LOB 模式 – 在内联 LOB 模式下，您可以设置 DMS 以内联方式传输的最大 LOB 大小。小于指定大小的 LOB 将以内联方式传输。大于指定大小的 LOB 将使用完整 LOB 模式进行复制。当大多数 LOB 都很小时，您可以选择此选项来同时复制小型和大型 LOB。对于不支持完整 LOB 模式的端点，例如 S3 和 Redshift，DMS 不支持内联 LOB 模式。

**Note**

对于 Oracle，LOB 将被视为 VARCHAR 数据类型 (如果可能)。这种方法意味着从数据库中批量 AWS DMS 获取它们，这比其他方法快得多。Oracle 中的 VARCHAR 的最大大小为 32 K。因此，当 Oracle 为源数据库时，小于 32 K 的受限 LOB 大小最佳。

- 如果将任务配置为在受限 LOB 模式下运行，Max LOB size (K) (最大 LOB 大小 (K)) 选项设置 AWS DMS 接受的最大大小 LOB。大于该值的任何 LOB 将截断到该值。
- 将任务配置为使用完整 LOB 模式时，会分段 AWS DMS 检索 LOB。LOB 块大小 (K) 选项确定每个分段的大小。在设置该选项时，请特别注意网络配置允许的最大包大小。如果 LOB 块大小超出允许的最大包大小，则可能会看到连接断开错误。LobChunkSize 的建议值为 64 KB。将 LobChunkSize 的值增加到 64 KB 以上可能会导致任务失败。
- 将任务配置为在内联 LOB 模式下运行时，InlineLobMaxSize 设置将确定 DMS 以内联方式传输哪些 LOB。

**Note**

您只能将 LOB 数据类型用于包含主键的表或视图。



有关指定这些选项的任务设置的信息，请参阅[目标元数据任务设置](#)

## 创建多个任务

在某些迁移方案中，您可能必须创建多个迁移任务。任务可以独立运行，也可以并发运行。每个任务都有其自己的初始负载、CDC 和日志读取过程。通过数据操作语言 (DML) 相关的表必须属于同一任务。

为迁移创建多个任务的部分原因包括：

- 任务的目标表位于不同的数据库中，例如，当您将一个系统分散到或分成多个系统时。
- 您需要使用筛选功能将大型表的迁移分为多个任务。

### Note

由于每个任务有其自己的更改捕获和日志读取过程，因此不会跨任务协调更改。因此，当使用多个任务执行迁移时，请确保每个单独的源事务完全包含在单个任务中。如果没有将单个事务拆分到不同的任务中，则可以使用多个任务来执行迁移。

## 使用 AWS DMS 为持续复制创建任务

您可以创建一个 AWS DMS 任务来捕获源数据存储的持续更改。您可以在迁移数据时执行此捕获。您还可以创建一个任务，以便在初始（完全加载）迁移到支持的目标数据存储完成后捕获持续更改。此过程称为持续复制或更改数据捕获 (CDC)。AWS DMS 在从源数据存储复制持续更改时使用此过程。此过程的工作方式是使用数据库引擎的原生 API 来收集对数据库日志的更改。

### Note

您只能使用完全加载任务迁移视图。如果您的任务是仅 CDC 任务或在完成后启动 CDC 的完全加载任务，则迁移仅包含来自源的表。使用仅完全加载任务，您可以迁移视图或表和视图的组合。有关更多信息，请参阅[使用 JSON 指定表选择和转换规则](#)。

每个源引擎有特定的配置要求，用于将此更改流公开到指定用户账户。大部分引擎需要一些额外的配置才能让捕获流程以有意义的方式使用更改数据，而不会丢失数据。例如，Oracle 需要增加补充日志记录，MySQL 需要行级二进制日志记录 (bin logging)。

为了从源数据库读取持续更改，AWS DMS 使用引擎特定的 API 操作从源引擎的事务日志中读取更改。下面是一些 AWS DMS 如何处理的示例：

- 对于 Oracle，AWS DMS 使用 Oracle LogMiner API 或 Binary Reader API (bfile API) 读取持续更改。AWS DMS 根据系统更改号 (SCN) 从联机或存档重做日志中读取持续更改。
- 对于 Microsoft SQL Server，AWS DMS 使用 MS-Replication 或 MS-CDC 将信息写入 SQL Server 事务日志。然后，它使用 SQL Server 中的 `fn_dblog()` 或 `fn_dump_dblog()` 函数，根据日志序列号 (LSN) 读取事务日志中的更改。
- 对于 MySQL，AWS DMS 从基于行的二进制日志 (binlog) 读取更改，并将这些更改迁移到目标。
- 对于 PostgreSQL，AWS DMS 设置逻辑复制槽并使用 `test_decoding` 插件从源读取更改，并将更改迁移到目标。
- 对于将 Amazon RDS 作为源的情况，我们建议确保启用了备份以设置 CDC。我们还建议确保源数据库已配置为保留更改日志足够长的时间，通常 24 小时已足够。有关每个端点的特定设置，请参阅以下内容：
  - Amazon RDS for Oracle：[为配置由 AWS 托管的 Oracle 源 AWS DMS。](#)
  - Amazon RDS for MySQL 和 Aurora MySQL：[使用 AWS 托管的 MySQL 兼容数据库作为源 AWS DMS。](#)
  - Amazon RDS for SQL Server：[在云 SQL Server 数据库实例上设置持续复制。](#)
  - Amazon RDS for PostgreSQL 和 Aurora PostgreSQL：PostgreSQL 自动保留所需的 WAL。

有两种类型的持续复制任务：

- 完全加载加 CDC – 该任务迁移现有数据，然后根据对源数据库的更改更新目标数据库。
- 仅 CDC – 当目标数据库上存在数据之后，该任务会迁移持续更改。

## 从 CDC 开始点开始执行复制

您可以从多个点启动 AWS DMS 持续复制任务（仅更改数据捕获）。这些功能包括：

- 从自定义 CDC 开始时间 — 您可使用 AWS Management Console 或 AWS CLI 为 AWS DMS 提供您希望复制开始的时间戳。之后，AWS DMS 从此自定义 CDC 开始时间启动持续复制任务。AWS DMS 将指定的时间戳（采用 UTC 格式）转换为本机开始点，例如，对于 SQL Server 为 LSN，对于 Oracle 为 SCN。AWS DMS 使用引擎特定的方法，根据源引擎的更改流确定从哪个位置开始迁移任务。

**Note**

只有将 `StartFromContext` 连接属性设置为所需的时间戳，Db2 作为源才能提供自定义的 CDC 开始时间。

当 PostgreSQL 作为源时，不支持自定义 CDC 开始时间。这是因为 PostgreSQL 数据库引擎无法将时间戳映射到 LSN 或 SCN，而 Oracle 或 SCN Server 可以。

- 从 CDC 本机开始点 – 您也可以从源引擎的事务日志中的本机时间点开始。在某些情况下，您可能首选此方法，因为时间戳可以指出事务日志中的多个本机时间点。AWS DMS 支持此功能用于以下源端点：
  - SQL Server
  - PostgreSQL
  - Oracle
  - MySQL
  - MariaDB

创建任务时，AWS DMS 标记 CDC 开始点，并且无法更改。要使用不同的 CDC 开始点，请创建新任务。

## 确定 CDC 本机开始点

CDC 本机开始点是数据库引擎日志中的一个点，定义开始 CDC 的时间。例如，假设批量数据转储已应用于目标。您可以查找持续仅复制任务的本机开始点。为避免任何数据不一致，请谨慎选择仅复制任务的开始点。DMS 会捕获在选定 CDC 开始点之后开始的事务。

以下示例演示如何从支持的源引擎查找 CDC 本机开始点：

### SQL Server

在 SQL Server 中，日志序列号 (LSN) 分为三个部分：

- 虚拟日志文件 (VLF) 序列号
- 日志块的启动偏移量
- 槽编号

示例 LSN 如下所示：`00000014:00000061:0001`

要根据事务日志备份设置获取 SQL Server 迁移任务的开始点，请在 SQL Server 中使用 `fn_dblog()` 或 `fn_dump_dblog()` 函数。

要在 SQL Server 中使用 CDC 本机开始点，请在参与持续复制的任何表上创建发布。AWS DMS 在不使用 CDC 本机开始点的情况下使用 CDC 时会自动创建发布。

## PostgreSQL

可以对 PostgreSQL 源数据库使用 CDC 恢复检查点。在为源数据库运行持续复制任务（父任务）时，会在各个点生成此检查点值。有关一般检查点的更多信息，请参阅[使用检查点作为 CDC 开始点](#)。

要标识要用作本机开始点的检查点，请使用数据库 `pg_replication_slots` 视图或 AWS Management Console 中父任务的概述详细信息。

在控制台上查找父任务的概述详细信息

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。

如果以 IAM 用户身份登录，请确保具有适当的 AWS DMS 访问权限。有关所需权限的更多信息，请参阅[使用 AWS DMS 所需的 IAM 权限](#)。

2. 在导航窗格中，选择 Database migration tasks (数据库迁移任务)。
3. 从 Database migration tasks (数据库迁移任务) 页面上的列表中选择您的父任务。这会打开显示概述详细信息的父任务页面。
4. 在 Change data capture (CDC) (更改数据捕获 (CDC))、Change data capture (CDC) start position (更改数据捕获 (CDC) 开始位置) 和 Change data capture (CDC) recovery checkpoint (更改数据捕获 (CDC) 恢复检查点) 下找到检查点值。

该值如下面所示。

```
checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0
```

在这里，4AF/B00000D0 组件是您需要指定此本机 CDC 开始点的组件。当您创建 CDC 任务以在 PostgreSQL 源的此开始点开始复制时，将 DMS API `CdcStartPosition` 参数设置为此值。有关使用 AWS CLI 创建此 CDC 任务的信息，请参阅[使用 AWS 托管的 PostgreSQL 数据库实例启用 CDC AWS DMS](#)。

## Oracle

系统更改号 (SCN) 是由 Oracle 数据库使用的逻辑内部时间戳。SCN 对数据库中发生的事件排序，这是满足事务的 ACID 属性所必需的。Oracle 数据库使用 SCN 标记所有更改写入到磁盘中的位置，这样恢复操作就不会应用已经写入的更改。Oracle 还使用 SCN 标记数据集中不存在重做的点，这样可以停止恢复。

要获取 Oracle 数据库中的当前 SCN，请运行以下命令。

```
SELECT CURRENT_SCN FROM V$DATABASE
```

如果您使用 SCN 或时间戳启动 CDC 任务，则会错过所有未完成事务的结果，因此无法迁移这些结果。未完成事务是在任务的开始位置之前启动，并在任务开始位置之后提交的事务。您可以标识 SCN 和时间戳，以便在包含所有未完成事务的时间点启动 CDC 任务。有关更多信息，请参阅 Oracle 在线文档中的[事务](#)。在 3.5.1 及更高版本中，如果您使用 SCN 或时间戳启动任务，则 AWS DMS 使用 openTransactionWindow 端点设置支持仅 CDC 任务的未完成事务。

在使用 openTransactionWindow 设置时，您必须提供以分钟为单位的窗口来处理未完成事务。AWS DMS 移动捕获位置并找到开始数据捕获的新位置。AWS DMS 使用新的开始位置扫描所需 Oracle 重做或存档重做日志中的所有未完成事务。

## MySQL

在 MySQL 版本 5.6.3 发行之之前，MySQL 的日志序列号 (LSN) 为 4 字节的无符号整数。在 MySQL 版本 5.6.3 中，重做日志文件大小限制从 4 GB 增加到 512 GB，因此 LSN 将成为 8 字节的无符号整数。这种增长体现了存储超大信息所需的额外字节。对于在 MySQL 5.6.3 或更高版本上生成的应用程序，如果使用 LSN 值，则应该使用 64 位而非 32 位变量来存储和比较 LSN 值。有关 MySQL LSN 的更多信息，请参阅[MySQL 文档](#)。

要获取 MySQL 数据库中的当前 LSN，请运行以下命令。

```
mysql> show master status;
```

该查询返回二进制日志文件名、位置以及多个其他值。CDC 本机开始点是二进制日志文件名和位置的组合，例如 mysql-bin-changelog.000024:373。在此示例中，mysql-bin-changelog.000024 是二进制日志文件名，373 是 AWS DMS 需要开始捕获更改的位置。

## 使用检查点作为 CDC 开始点

持续复制任务将迁移更改，而 AWS DMS 将不断缓存特定于 AWS DMS 的检查点信息。AWS DMS 创建的检查点包含信息，因此复制引擎知道更改流的恢复点。您可以使用检查点返回到更改时间表中的某个点，以及恢复失败的迁移任务。您也可以使用检查点，以在任意指定时间为其他目标启动另一个持续复制任务。

您可以通过以下三种方法之一获取检查点信息：

- 运行 API 操作 `DescribeReplicationTasks` 并查看结果。您可以按任务筛选信息以及搜索检查点。当任务处于停止/失败状态时，您可以检索最新的检查点。如果任务已被删除，则此信息丢失。
- 查看目标实例上名为 `awsdms_txn_state` 的元数据表。您可以查询表来获取检查点信息。要创建元数据表，请在创建任务时将 `TaskRecoveryTableEnabled` 参数设置为 `Yes`。此设置导致 AWS DMS 连续将检查点信息写入目标元数据表。如果任务已删除，则此信息丢失。

例如，以下是元数据表中的示例检查

点：`checkpoint:V1#34#00000132/0F000E48#0#0#*#0#121`

- 在导航窗格中选择数据库迁移任务，然后从“数据库迁移任务”页面上显示的列表中选择您的父任务。会打开显示概述详细信息的父任务页面。在 `Change data capture (CDC)` (更改数据捕获 (CDC))、`Change data capture (CDC) start position` (更改数据捕获 (CDC) 开始位置) 和 `Change data capture (CDC) recovery checkpoint` (更改数据捕获 (CDC) 恢复检查点) 下找到检查点值。检查点值如下所示：

`checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0`

## 在提交时间点或服务器时间点停止任务

通过引入 CDC 本机开始点，AWS DMS 还可以在以下点停止任务：

- 源上的提交时间
- 复制实例上的服务器时间

您可以修改某个任务，并根据需要使用 UTC 格式设置停止时间。该任务根据您设置的提交时间或服务器时间自动停止。或者，如果您知道合适的停止迁移任务时间，您也可在创建任务时设置停止时间。

**Note**

首次开始新的 AWS DMS 无服务器复制时，最长可能需要 40 分钟才能初始化所有资源。请注意，`server_time` 选项仅在资源初始化完成后才适用。

## 执行双向复制

您可以使用 AWS DMS 任务在两个系统之间执行双向复制。在双向复制中，您在两个系统之间在两个方向上复制同一个表（或同一组表）中的数据。

例如，您可以将 EMPLOYEE 表从数据库 A 复制到数据库 B，并将对此表的更改从数据库 A 复制到数据库 B。您还可以将对此 EMPLOYEE 表的更改从数据库 B 复制回到 A。因此，您正在执行双向复制。

**Note**

AWS DMS 双向复制不适合用作完整的多主数据库解决方案（包括主节点、冲突解决等）。

对于对不同节点上的数据进行操作隔离的情况，使用双向复制。换句话说，假设在节点 A 上运行的应用程序更改了一个数据元素，并且节点 A 与节点 B 执行双向复制。节点 A 上的该数据元素永远不会被在节点 B 上运行的任何应用程序更改。

AWS DMS 支持在以下数据库引擎上进行双向复制：

- Oracle
- SQL Server
- MySQL
- PostgreSQL
- Amazon Aurora MySQL 兼容版
- Aurora PostgreSQL 兼容版

## 创建双向复制任务

要启用 AWS DMS 双向复制，请为两个数据库（A 和 B）配置源终端节点和目标终端节点。例如，配置数据库 A 的源终端节点、数据库 B 的源终端节点、数据库 A 的目标终端节点和数据库 B 的目标终端节点。

然后创建两个任务：一个任务用于源 A 将数据移到目标 B，另一个任务用于源 B 将数据移到目标 A。此外，请确保为每个任务配置了环回防护。这样做可以防止将完全相同的更改应用于两个任务的目标，这样会损坏其中至少一个任务的数据。有关更多信息，请参阅[防止环回](#)。

最简单的方法是从数据库 A 和数据库 B 上的相同数据集开始，然后创建两个仅 CDC 任务，一个任务将数据从 A 复制到 B，另一个任务将数据从 B 复制到 A。

要使用 AWS DMS 从节点 A 实例化节点 B 上的新数据集（数据库），请执行以下操作：

1. 使用完全加载和 CDC 任务将数据从数据库 A 移到 B。确保在此期间没有任何应用程序正在修改数据库 B 上的数据。
2. 当完全加载完成并且在允许应用程序修改数据库 B 上的数据之前，请记下数据库 B 的时间或 CDC 开始位置。有关说明，请参阅[从 CDC 开始点开始执行复制](#)。
3. 创建仅 CDC 任务，此任务使用此开始时间或 CDC 开始位置将数据从数据库 B 移回 A。

#### Note

双向对中只有一个任务可以是完全加载和 CDC。

## 防止环回

要显示防止环回，假设在任务 T1 中，AWS DMS 从源数据库 A 读取更改日志，并将更改应用于目标数据库 B。

接下来，第二个任务 T2 从源数据库 B 读取更改日志，并将其应用回目标数据库 A。在 T2 执行此操作之前，DMS 必须确保不会对源数据库 A 进行从源数据库 A 对目标数据库 B 所做的相同更改。换句话说，DMS 必须确保这些更改不会回显（环回）到目标数据库 A。否则，数据库 A 中的数据可能会被破坏。

要防止环回更改，请将以下任务设置添加到每个双向复制任务。这样做可以确保不会在任何一个方向发生环回数据损坏。

```
{  
  . . .  
  
  "LoopbackPreventionSettings": {  
    "EnableLoopbackPrevention": Boolean,
```



```

    "SourceSchema": String,
    "TargetSchema": String
  },
  . . .
}

```

LoopbackPreventionSettings 任务设置确定事务是新事务还是来自反向复制任务的回显。当 AWS DMS 将事务应用于目标数据库时，它会更新 DMS 表 (awsdms\_loopback\_prevention)，并指示更改。在将每个事务应用于目标之前，DMS 会忽略包含对此 awsdms\_loopback\_prevention 表的引用的任何事务。因此，它不应用更改。

将这些任务设置包括在双向对中的每个复制任务中。这些设置启用环回防护。它们还为包含每个终端节点的 awsdms\_loopback\_prevention 表的任务中的每个源数据库和目标数据库指定架构。

要使每个任务能够识别并丢弃回显，请将 EnableLoopbackPrevention 设置为 true。要在包含 awsdms\_loopback\_prevention 的源处指定架构，请将 SourceSchema 设置为源数据库中该架构的名称。要在包含同一个表的目标处指定架构，请将 TargetSchema 设置为目标数据库中该架构的名称。

在以下示例中，复制任务 T1 及其反向复制任务 T2 的 SourceSchema 和 TargetSchema 设置使用相反的设置指定。

任务 T1 的设置如下所示。

```

{
  . . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "LOOP-DATA",
    "TargetSchema": "loop-data"
  },
  . . .
}

```

反向任务 T2 的设置如下所示。

```

{

```

```
. . .  
  
  "LoopbackPreventionSettings": {  
    "EnableLoopbackPrevention": true,  
    "SourceSchema": "loop-data",  
    "TargetSchema": "LOOP-DATA"  
  },  
  
. . .  
}
```

### Note

使用 AWS CLI 时，请仅使用 `create-replication-task` 或 `modify-replication-task` 命令在双向复制任务中配置 `LoopbackPreventionSettings`。

## 双向复制的限制

AWS DMS 的双向复制具有以下限制：

- 环回防护仅跟踪数据操作语言 (DML) 语句。AWS DMS 不支持阻止数据定义语言 (DDL) 环回。要执行此操作，请配置双向对中的任务之一以筛选出 DDL 语句。
- 使用环回防护的任务不支持批量提交更改。要配置具有环回防护的任务，请确保 `BatchApplyEnabled` 设置为 `false`。
- DMS 双向复制不包括冲突检测或解决。要检测数据不一致性，请对这两个任务使用数据验证。

## 修改任务

如果您需要更改任务设置、表映射或其他设置，可以修改任务。您还可以先启用并运行迁移前评测，然后再运行修改后的任务。您可以在控制台中选择任务，然后选择修改以修改任务。您还可以使用 CLI 命令或 API 操作 [ModifyReplicationTask](#)。

对于修改任务，存在一些限制。这些功能包括：

- 无法修改任务的源或目标终端节点。
- 您无法更改任务的迁移类型。
- 已运行任务的状态必须为已停止或失败才能进行修改。

## 移动任务

当以下任何一种情况适用于您的使用案例时，您可以将任务移至其他复制实例。

- 您目前正在使用某种类型的实例，并且想要切换到其他实例类型。
- 您的当前实例因大量复制任务而过载，您需要在多个实例之间拆分负载。
- 您的实例存储空间已满，您希望将任务从该实例转移到性能更强大的实例，以此作为扩展存储或计算的替代方案。
- 您想使用 AWS DMS 新发布的功能，但不想创建新任务并重新启动迁移。相反，您更喜欢使用支持该功能的新 AWS DMS 版本启动复制实例，并将现有任务移至该实例。

您可以在控制台中选择任务，然后选择移动以移动任务。您也可以运行 CLI 命令或 API 操作 `MoveReplicationTask` 来移动任务。您可以移动以任何数据库引擎作为其目标端点的任务。

请确保目标复制实例有足够的存储空间来容纳正在移动的任务。否则，请在移动任务之前扩展存储空间，以便为目标复制实例腾出空间。

另请确保使用与当前复制实例相同或更高的 AWS DMS 引擎版本创建目标复制实例。

### Note

- 您无法将任务移至任务当前所在的相同复制实例。
- 任务在移动时，您无法修改设置。
- 您已运行的任务必须处于已停止、失败或移动失败状态才能进行移动。

有两种任务状态与移动 DMS 任务有关，分别是正在移动和移动失败。有关这些任务状态的更多信息，请参阅[任务状态](#)。

移动任务后，您可以启用并运行迁移前评测，以便在运行已移动的任务之前检查是否存在阻碍问题。

## 在任务期间重新加载表

当任务正在运行时，可以使用源中的数据重新加载目标数据库表。在任务期间，如果因分区操作（例如，在使用 Oracle 时）导致出错或数据更改，则可能需要重新加载表。可以从一个任务重新加载最多 10 个表。

重新加载表并不能停止任务。

要重新加载表，必须满足以下条件：

- 任务必须正在运行。
- 任务的迁移方法必须是完全加载或者完全加载加 CDC。
- 不允许重复表。
- AWS DMS 将保留之前读取的表定义并且在重新加载操作期间不会重新创建该定义。在重新加载表之前对该表所执行的任何 DDL 语句（如 ALTER TABLE ADD COLUMN 或 DROP COLUMN）都可能会导致重新加载操作失败。

#### Note

DMS 会在重新加载表之前应用 TargetTablePrepMode 设置。如果将 TargetTablePrepMode 设置为 DO\_NOTHING，必须先手动截断表。

## AWS Management Console

使用 AWS DMS 控制台重新加载表

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。

如果以 IAM 用户身份登录，请确保具有适当的 AWS DMS 访问权限。有关所需权限的更多信息，请参阅[使用 AWS DMS 所需的 IAM 权限](#)。

2. 从导航窗格中选择任务。
3. 选择具有要重新加载的表的正在运行的任务。
4. 选择表统计数据选项卡。

The screenshot shows the AWS DMS console interface. At the top, there are buttons for 'Create task', 'Assess', 'Modify', 'Start/Resume', 'Stop', and 'Delete'. Below these is a search filter. A table lists tasks, with 'move-data' selected and its status 'Running'. Below the task list, the 'Table statistics' tab is active, and the 'Reload table data' button is highlighted with a red circle. Below the tabs is another search filter and a table of table statistics.

Schema	Table	Load State	Inserts	Deletes	Updates	DDLs	Full Load
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. 选择要重新加载的表。如果任务不再运行，则您无法重新加载该表。
6. 选择 Reload table data (重新加载表数据)。

在 AWS DMS 准备重新加载表时，控制台将表状态更改为 Table is being reloaded (正在重新加载表)。

## 使用表映射指定任务设置

表映射使用多个类型的规则来指定数据源、源架构、数据以及任务期间将发生的任何转换。可以使用表映射指定数据库中要迁移的单个表和要用于迁移的架构。

使用表映射时，可以使用筛选器指定要从表复制的数据。此外，您还可以使用转换来修改选定的架构、表或视图，然后再将其写入目标数据库。

## 主题

- [通过控制台指定表选择和转换规则](#)
- [使用 JSON 指定表选择和转换规则](#)
- [选择规则和操作](#)
- [表映射中的通配符](#)
- [转换规则和操作](#)
- [使用转换规则表达式定义列内容](#)
- [表和集合设置规则和操作](#)

### Note

在将表映射用于 MongoDB 源端点时，可以使用筛选器指定要复制的数据，然后指定一个数据库名称来替代 `schema_name`。或者，可以使用默认的 "%"。

## 通过控制台指定表选择和转换规则

您可以使用 AWS Management Console 来执行表映射，包括指定表选择和转换。在控制台上，您可以使用位置部分指定架构、表和操作（包括或排除）。使用筛选器部分指定表中的列名称以及要应用于复制任务的条件。这两个操作共同创建选择规则。

在指定至少一个选择规则后，可以在表映射中包含转换。可以使用转换重命名架构或表，为架构或表添加前缀或后缀，或者删除表列。

### Note

AWS DMS 不支持每个架构级别、表级别或列级别的多个转换规则。

以下过程演示了如何根据名为 **EntertainmentAgencySample** 的架构中名为 **Customers** 的表来设置选择规则。

使用控制台指定表选择、筛选条件和转换

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。

如果以 IAM 用户身份登录，请确保具有适当的 AWS DMS 访问权限。有关所需权限的更多信息，请参阅[使用 AWS DMS 所需的 IAM 权限](#)。

2. 在控制面板页面上，选择数据库迁移任务。
3. 选择创建任务。
4. 在任务配置部分中，输入任务信息，包括任务标识符、复制实例、源数据库端点、目标数据库端点和迁移类型。

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](#)

5. 在表映射部分中，选择架构名称和表名称。在指定架构名称或表名称时，可以使用“%”作为通配符值。有关可使用的其他通配符的信息，请参阅[the section called “表映射中的通配符”](#)。指定包括或排除由筛选器定义的数据所需执行的操作。

**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. 使用添加列筛选器和添加条件链接指定筛选器信息。
  - a. 选择添加列筛选器以指定列和条件。
  - b. 选择添加条件以添加其他条件。

以下示例介绍 **Customers** 表的筛选器，包含介于 **01** 和 **85** 之间的 **AgencyIDs**。

Source filters [Info](#) Add column filter

▼ Column filter 1 ×

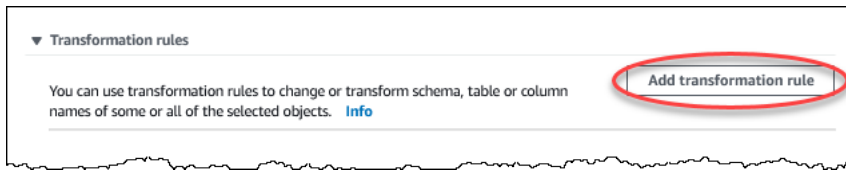
Column name  
AgencyId

Condition 1  
Equal to or between two values 01  
85

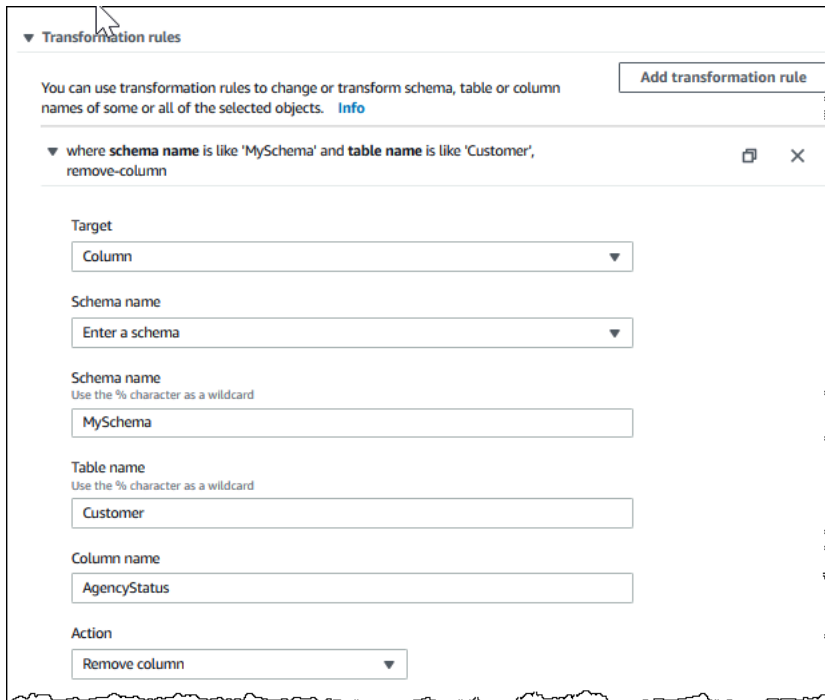
Add condition

7. 在创建所需的选择后，选择添加新的选择规则。
8. 创建至少一个选择规则后，可以向任务添加转换。选择添加转换规则。





9. 选择要转换的目标，然后输入请求的其他信息。以下示例介绍从 **Customer** 表中删除 **AgencyStatus** 列的转换。



10. 选择添加转换规则。

11. 选择创建任务。

### Note

AWS DMS 不支持每个架构级别或每个表级别的多个转换规则。

## 使用 JSON 指定表选择和转换规则

要指定在迁移过程中要应用的表映射，可以创建 JSON 文件。如果使用控制台创建迁移任务，则可浏览此 JSON 文件或在表映射框中直接输入 JSON。如果使用 CLI 或 API 执行迁移，那么可以使用 `CreateReplicationTask` 或 `ModifyReplicationTask` API 操作的 `TableMappings` 参数指定此文件。

AWS DMS 只能处理大小不超过 2 MB 的表映射 JSON 文件。我们建议您在处理 DMS 任务时，将映射规则 JSON 文件大小保持在 2 MB 限制以下。这样可防止在创建或修改任务过程中出现意外错误。如果映射规则文件超过 2 MB 限制，我们建议您将表拆分为多个任务，以减小映射规则文件的大小，使其保持在该限制以下。

您可以指定要处理哪些表、视图和架构。您还可以执行表、视图和架构转换，并指定 AWS DMS 如何加载单独的表和视图。您使用以下规则类型为这些选项创建表映射规则：

- **selection** 规则 – 确定要加载的源表、视图和架构的类型和名称。有关更多信息，请参阅 [选择规则和操作](#)。
- **transformation** 规则 – 指定在将源中的特定源表和架构加载到目标之前，对其所做某些更改或添加。有关更多信息，请参阅 [转换规则和操作](#)。

此外，要定义新列和现有列的内容，可以在转换规则中使用表达式。有关更多信息，请参阅 [使用转换规则表达式定义列内容](#)。

- **table-settings** 规则 – 指定 DMS 任务为各个表加载数据的方式。有关更多信息，请参阅 [表和集合设置规则和操作](#)。

#### Note

对于 Amazon S3 目标，也可以使用 `post-processing` 规则类型和 `add-tag` 规则操作，标记映射到所选表和架构的 S3 对象。有关更多信息，请参阅 [Amazon S3 对象标记](#)。


对于以下目标，可以使用 `object-mapping` 规则类型，指定将所选架构和表迁移到目标的方式和位置：

- Amazon DynamoDB – 有关更多信息，请参阅 [使用对象映射将数据迁移到 DynamoDB](#)。
- Amazon Kinesis – 有关更多信息，请参阅 [使用对象映射将数据迁移到 Kinesis 数据流](#)。
- Apache Kafka – 有关更多信息，请参阅 [使用对象映射将数据迁移到 Kafka 主题](#)。

## 选择规则和操作

通过使用表映射，可通过选择规则和操作来指定要使用的表、视图和架构。对于使用选择规则类型的表映射规则，您可应用以下值。

参数	可能的值	描述
rule-type	selection	一个选择规则。在指定表映射时定义至少一个选择规则。
rule-id	一个数字值。	一个用于标识规则的唯一数字值。
rule-name	一个字母数字值。	一个用于标识规则的唯一名称。
rule-action	include, exclude, explicit	一个值，包括或排除规则所选定的对象。如果指定 explicit，则可以选择并仅包含一个对应于显式指定的表和架构的对象。
object-locator	<p>一个带以下参数的对象：</p> <ul style="list-style-type: none"> <li>• schema-name – 架构的名称。</li> <li>• table-name – 表的名称。</li> <li>• ( 可选 ) table-type – table   view   all，用于指示 table-name 是仅引用表或视图，还是同时引用表和视图。默认值为 table。</li> </ul> <p>AWS DMS 仅在满载任务中加载视图。如果您只有满载和更改数据捕获 (CDC) 任务，请至少配置一项 full-load-only 任务来加载视图。</p> <p>并非所有目标终端节点都接受视图作为复制源，即使在满负荷状态下也是如此（例如 Amazon S OpenSearch ervice）。检查目标终端节点的限制。</p>	<p>规则应用于的每个架构和表或视图的名称。您还可以指定某个规则只包含表、只包含视图还是同时包含表和视图。如果 rule-action 为 include 或 exclude，您可以使用“%”百分比符号作为 schema-name 和 table-name 参数的值的全部或部分的通配符。有关可使用的其他通配符的信息，请参阅<a href="#">the section called “表映射中的通配符”</a>。因此，您可以匹配这些项目：</p> <ul style="list-style-type: none"> <li>• 单个架构中的单个表、视图或集合</li> <li>• 部分或所有架构中的单个表、视图或集合</li> <li>• 单个架构中的部分或所有表和视图，或单个数据库中的集合</li> <li>• 部分或所有架构中的部分或所有表和视图，或者部分或所有数据库中的集合</li> </ul> <p>如果 rule-action 为 explicit，则只能指定单个表或视</p>

参数	可能的值	描述
		<p>图及其架构的确切名称（不带通配符）。</p> <p>支持的视图的源包括：</p> <ul style="list-style-type: none"> <li>• Oracle</li> <li>• Microsoft SQL Server</li> <li>• PostgreSQL</li> <li>• IBM Db2 LUW</li> <li>• IBM Db2 z/OS</li> <li>• SAP Adaptive Server Enterprise ( ASE )</li> <li>• MySQL</li> <li>• Aurora</li> <li>• Aurora Serverless</li> <li>• MariaDB</li> </ul> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>AWS DMS 从不将源视图加载到目标视图。源视图加载到目标上等价的表，具有与源上视图的相同名称。</p> </div> <p>包含集合的数据库支持的源包括：</p> <ul style="list-style-type: none"> <li>• MongoDB</li> <li>• Amazon DocumentDB</li> </ul>
load-order	一个正整数。最大值为 2147483647。	加载表和视图的优先级。首先加载具有较高值的表和视图。

参数	可能的值	描述
filters	一个对象数组。	一个或多个用于筛选源的对象。您指定要在源中的单个列上筛选的对象参数。您指定要在多个列上筛选的多个对象。有关更多信息，请参阅 <a href="#">使用源筛选器</a> 。

### Example 迁移架构中的所有表

以下示例将所有表从源中名为 Test 的架构中迁移到目标端点。

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "Test",
        "table-name": "%"
      },
      "rule-action": "include"
    }
  ]
}
```

### Example 迁移架构中的一些表

以下示例将所有表（以 DMS 开头的表除外）从源中名为 Test 的架构中迁移到目标端点。

```
{
  "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 在单个架构中迁移指定的单个表

以下示例将源中 NewCust 架构中的 Customer 表迁移到目标端点。

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "NewCust",
        "table-name": "Customer"
      },
      "rule-action": "explicit"
    }
  ]
}
```

#### Note

您可以通过指定多个选择规则在多个表和架构上显式选择。

### Example 按设定顺序迁移表

以下示例迁移两个表。表loadfirst ( 优先级为 1 ) 在表loadsecond之前初始化。

```
{
  "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` 适用于表初始化。如果 `MaxFullLoadSubTasks` 大于 1，则连续表的加载不会等待上一个表的加载完成。

### Example 迁移架构中的部分视图

以下示例将部分视图从源中名为 `Test` 的架构迁移到目标中等同的表。

```
{
  "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 迁移架构中的所有表和视图

以下示例将所有表和视图从源中名为 `report` 的架构迁移到目标中等同的表。

```

{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "3",
      "rule-name": "3",
      "object-locator": {
        "schema-name": "report",
        "table-name": "%",
        "table-type": "all"
      },
      "rule-action": "include"
    }
  ]
}

```

## 表映射中的通配符

本节介绍在为表映射指定架构和表名称时可以使用的通配符。

通配符	匹配项
%	零个或多个字符
_	单个字符
[_]	字面下划线字符



通配符	匹配项
[ab]	一组字符。例如，[ab] 匹配'a'或'b'。
[a-d]	一系列字符。例如，[a-d] 匹配'a'、'b'、'c' 或 'd'。

对于 Oracle 源端点和目标端点，可以使用 `escapeCharacter` 额外的连接属性指定转义字符。转义字符允许在表达式中使用指定的通配符，就好像它不是通配符一样。例如，`escapeCharacter=#` 允许使用“#”，使通配符在表达式中充当普通字符，如本示例代码中所示。

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "542485267",
      "rule-name": "542485267",
      "object-locator": { "schema-name": "ROOT", "table-name": "TEST#_T%" },
      "rule-action": "include",
      "filters": []
    }
  ]
}
```

在这里，'#' 转义字符使 '\_' 通配符充当普通字符。AWS DMS 在名为的架构中选择表ROOT，其中每个表都有一个名字TEST\_T作为其前缀。

## 转换规则和操作

可使用转换操作指定要应用于所选架构、表或视图的任何转换。转换规则是可选的。

### 限制

- 不能对同一个对象（架构、表、列、表空间或索引表空间）应用多个转换规则操作。只要每个转换操作应用于不同的对象，那么可以在任何级别应用多个转换规则操作。
- 转换规则中的表名称和列名称是区分大小写的。例如，Oracle 或 Db2 数据库的表名称和列名称必须以大写形式提供。
- 对于采用从右到左语言的列名称，不支持转换。

- 对于名称中包含特殊字符 ( 如 #、\、/、- ) 的列，无法执行转换。
- 对于映射到 BLOB/CLOB 数据类型的列，唯一支持的转换是在目标上删除该列。
- AWS DMS 不支持将两个源表复制到单个目标表。AWS DMS 根据复制任务的转换规则，将记录从一个表复制到另一个表以及从一个列复制到另一个列。对象名称必须唯一，以防止重叠。

例如，源表包含名为 ID 的列，而对应的目标表预先存在名为 id 的列。如果规则使用 ADD-COLUMN 语句添加名为 id 的新列，并使用 SQLite 语句在该列中填充自定义值，则会创建一个名为 id 的重复、有歧义的对象，且不受支持。

## 值

对于使用转换规则类型的表映射规则，可应用以下值。

参数	可能的值	描述
rule-type	transformation	一个将规则应用于选择规则所指定的每个对象的值。使用 transformation ( 除非另有说明 )。
rule-id	一个数字值。	一个用于标识规则的唯一数字值。如果您为同一个对象 ( 架构、表、列、表间空间或索引表空间 ) 指定多个转换规则，则使用较低的规则 ID AWS DMS 应用转换规则。
rule-name	一个字母数字值。	一个用于标识规则的唯一名称。
object-locator	一个带以下参数的对象： <ul style="list-style-type: none"> <li>• schema-name – 架构的名称。对于 MongoDB 和 Amazon DocumentDB 端点，这是存放一组集合的数据库的名称。</li> <li>• table-name – 表、视图或集合的名称。</li> <li>• table-tablespace-name – 现有表表空间的名称。</li> </ul>	规则应用于的每个架构、表或视图、表的表空间、索引表空间和列的名称。您可以使用“%”百分比符号作为每个 object-locator 参数的值的全部或一部分的通配符，但 data-type 除外。因此，您可以匹配这些项目： <ul style="list-style-type: none"> <li>• 单个架构中的单个表或视图</li> <li>• 部分或所有架构中的单个表或视图</li> </ul>

参数	可能的值	描述
	<ul style="list-style-type: none"> <li>• <code>index-tablespace-name</code> – 现有索引表空间的名称。</li> <li>• <code>column-name</code> – 现有列的名称。</li> <li>• <code>data-type</code> – 现有列数据类型的名称。</li> </ul>	<ul style="list-style-type: none"> <li>• 单个架构中的部分或所有表和视图</li> <li>• 部分或所有架构中的部分或所有表和视图</li> <li>• 指定的一个或多个表、一个或多个视图以及一个或多个架构中的一个或多个列。</li> <li>• 指定多个列时，具有给定 <code>data-type</code> 的列。有关 <code>data-type</code> 的可能值，请参阅下表中说明的 <code>data-type</code>。</li> </ul> <p>此外，<code>table-tablespace-name</code> 或 <code>index-tablespace-name</code> 参数仅可用于匹配 Oracle 源端点。您可以在单个规则中指定 <code>table-tablespace-name</code> 或 <code>index-tablespace-name</code>，但不能同时指定两者。因此，您可以匹配以下任一项：</p> <ul style="list-style-type: none"> <li>• 一个、部分或所有表空间</li> <li>• 一个、部分或所有索引表空间</li> </ul>

参数	可能的值	描述
rule-action	add-column , include-column , remove-column  rename  convert-lowercase , convert- uppercase  add-prefix , remove-prefix , replace-prefix  add-suffix , remove-suffix , replace-suffix  define-primary-key  change-data-type  add-before-image-columns	<p>要应用于对象的转换。所有转换规则操作都区分大小写。</p> <p>rule-action 参数的 add-column 值会将列添加到表中。但是无法添加与同一个表中现有列同名的新列。</p> <p>与 expression 和 data-type 参数结合使用时, add-column 指定新列数据的值。</p> <p>rule-action 的 change-data-type 值仅适用于 column 规则目标。</p> <p>rule-action 参数的 include-column 值会将表的模式更改为默认删除所有列和包括指定的列。通过多次调用 include-column 规则, 可以在目标中包含多列。</p> <p>如果 define-primary-key 规则的架构或表名称中包含通配符 ( % ) , 则无法使用该规则。</p> <p>对于现有任务, 更改目标表架构的转换规则操作 ( 例如 remove-column 、 rename 或 add-prefix ) 在您重新启动任务后才会生效。如果在添加转换规则后继续执行任务, 则可能会看到更改后的列出现意外行为, 其中可能包括缺少列数据。需要重新启动任务才能确保转换规则正常工作。</p>

参数	可能的值	描述
rule-target	schema, table, column, table-tablespace , index-tablespace	<p>要转换的对象类型。</p> <p>table-tablespace 和 index-tablespace 值仅适用于 Oracle 目标端点。</p> <p>请确保为指定作为 object-locator : table-tablespace-name 或 index-tablespace-name 名称一部分的参数指定值。</p>
value	一个字母数字值，该值遵循目标类型的命名规则。	需要输入的操作的新值，例如 rename。
old-value	一个字母数字值，该值遵循目标类型的命名规则。	需要替换的操作的旧值，例如 replace-prefix 。

参数	可能的值	描述
data-type	<p>type – 如果 rule-action 为 add-column ，则为要使用的数据类型；如果 rule-action 为 change-data-type ，则为替换数据类型。</p> <p>或者为替换数据类型的名称（如果 rule-action 为 change-data-type 、column-name 的值为 "%" ），并且，用于标识现有数据类型的附加 data-type 参数包括在 object-locator 中。</p> <p>AWS DMS 支持以下 DMS 数据类型的列数据类型转换："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 – 如果添加的列或替换数据类型具有精度，则为指定精度的整数值。</p> <p>scale – 如果添加的列或替换数据类型具有小数位数，则为指定小数位数的整数值或日期时间值。</p> <p>length – 新列数据的长度（与 add-column 结合使用时）</p>	<p>以下是指定要替换的现有数据类型的 data-type 参数的示例。</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> <p>在这里，table_t 表的 col10 列更改为 string 数据类型。</p>

参数	可能的值	描述
expression	遵循 SQLite 语法的字母数字值。	<p>当与 rule-action 设置为 rename-schema 结合使用时，expression 参数指定一个新的架构。当与 rule-action 设置为 rename-table 结合使用时，expression 指定一个新表。当与 rule-action 设置为 rename-column 结合使用时，expression 指定一个新的列名值。</p> <p>当与 rule-action 设置为 add-column 结合使用时，expression 指定构成新列的数据。</p> <p>请注意，此参数仅支持表达式。不支持运算符和命令。</p> <p>有关将表达式用于转换规则的更多信息，请参阅<a href="#">使用转换规则表达式定义列内容</a>。</p> <p>有关 SQLite 表达式的更多信息，请参阅<a href="#">使用 SQLite 函数构建表达式</a>。</p>
primary-key-def	<p>一个带以下参数的对象：</p> <ul style="list-style-type: none"> <li>name – 表或视图的新主键或唯一索引的名称。</li> <li>( 可选 ) origin – 要定义的唯一键的类型：primary-key ( 默认值 ) 或 unique-index 。</li> <li>columns – 一个字符串数组，按照列在主键或唯一索引中出现的顺序列出列的名称。</li> </ul>	<p>此参数可以定义转换的表或视图上唯一键的名称、类型和内容。当 rule-action 设置为 define-primary-key 并且 rule-target 设置 table 为时，它执行此操作。默认情况下，将唯一键定义为主键。</p>

参数	可能的值	描述
before-image-def	<p>一个带以下参数的对象：</p> <ul style="list-style-type: none"> <li>column-prefix – 附加到列名称前面的值。默认值为 BI_。</li> <li>column-suffix – 附加到列名称后面的值。默认值为空。</li> <li>column-filter – 需要以下值之一：pk-only ( 默认值 )、non-lob ( 可选 ) 和 all ( 可选 )。</li> </ul>	<p>此参数定义一个命名约定以标识之前映像列，并指定一个筛选器以确定哪些源列可能在目标上具有为它们创建的之前映像列。当 rule-action 设置为 add-before-image-columns 并且 rule-target 设置为 column 时，您可以指定此参数。</p> <p>不要同时将 column-prefix 和 column-suffix 设置为空字符串。</p> <p>对于 column-filter ，选择：</p> <ul style="list-style-type: none"> <li>pk-only – 仅添加属于表主键的列。</li> <li>non-lob – 仅添加非 LOB 类型的列。</li> <li>all – 添加具有 before-image 值的任何列。</li> </ul> <p>有关 AWS DMS 目标端点的 before-image 支持的更多信息，请参阅：</p> <ul style="list-style-type: none"> <li><a href="#">使用之前映像查看 Kinesis 数据流 ( 作为目标 ) 的 CDC 行的原始值</a></li> <li><a href="#">使用之前映像查看 Apache Kafka ( 作为目标 ) 的 CDC 行的原始值</a></li> </ul>

## 示例

### Example 重命名架构

以下示例将架构从 Test ( 源中 ) 重命名为 Test1 ( 目标中 )。



```
{
  "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 重命名表

以下示例将表从 Actor (源中) 重命名为 Actor1 (目标中)。

```
{
  "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 重命名列

以下示例将列从 `first_name` 中的表 `Actor` (源中) 重命名为 `fname` (目标中)。

```
{
  "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 重命名 Oracle 表空间

以下示例将 Oracle 源中名为 Actor 的表的名为 SetSpace 的表空间重命名为 Oracle 目标端点中的 SceneTblSpace。

```

{
  "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 重命名 Oracle 索引表空间

以下示例将 Oracle 源中名为 Actor 的表的名为 SetISpace 的索引表空间重命名为 Oracle 目标端点中的 SceneIdxSpace。

```

{

```

```
"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 添加列

以下示例将 `datetime` 列添加到架构 `test` 的表 `Actor` 中。

```
{  
  "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 删除列

以下示例将转换源中名为 Actor 的表，以从目标中删除以字符 col 开头的列。

```
{
  "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 转换为小写形式

以下示例将表名从 ACTOR (源中) 转换为 actor (目标中)。

```
{
  "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 转换为大写形式

以下示例将所有表和所有架构的所有列从小写形式 (源中) 转换为大写形式 (目标中)。

```
{
  "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 添加前缀

以下示例转换源中的所有表，以便在目标中向这些表添加前缀 DMS\_。

```
{
  "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 替换前缀

以下示例将转换源中包含前缀 `Pre_` 的所有列，以在目标中将前缀替换为 `NewPre_`。

```
{
  "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 删除后缀

以下示例转换源中的所有表，以从目标中删除这些表的后缀 `_DMS`。

```
{
  "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 定义主键

以下示例在迁移到目标端点的 ITEM 表的三个列上定义一个名为 ITEM-primary-key 的主键。

```
{
"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 定义唯一索引

以下示例在迁移到目标端点的 ITEM 表的三个列上定义一个名为 ITEM-unique-idx 的唯一索引。

```
{
  "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 更改目标列的数据类型

以下示例将名为 SALE\_AMOUNT 的目标列的数据类型从现有数据类型更改为 int8。

```

{
  "rule-type": "transformation",
  "rule-id": "1",
  "rule-name": "RuleName 1",
  "rule-action": "change-data-type",
  "rule-target": "column",
  "object-locator": {
    "schema-name": "dbo",
    "table-name": "dms",
    "column-name": "SALE_AMOUNT"
  },
  "data-type": {
    "type": "int8"
  }
}

```

### Example 添加之前映像列

对于名为 emp\_no 的源列，下面示例中的转换规则会在目标中添加名为 BI\_emp\_no 的新列。

```

{
  "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"
}
}
]
}
```

在此，下面的语句用 1 填充相应行中的 BI\_emp\_no 列。

```
UPDATE employees SET emp_no = 3 WHERE BI_emp_no = 1;
```

在向支持的 AWS DMS 目标写入 CDC 更新时，该 BI\_emp\_no 列可以判断 emp\_no 列中哪些行有更新的值。

## 使用转换规则表达式定义列内容

要定义新列和现有列的内容，可以在转换规则中使用表达式。例如，使用表达式，您可以添加列或将源表标头复制到目标。您还可以使用表达式将目标表上的记录标记为在源中插入、更新或删除。

### 主题

- [使用表达式添加列](#)
- [使用表达式标记目标记录](#)
- [使用表达式复制源表标头](#)
- [使用 SQLite 函数构建表达式](#)
- [使用表达式向目标表添加元数据](#)

### 使用表达式添加列

要使用转换规则中的表达式向表添加列，请使用 add-column 规则操作和 column 规则目标。

以下示例将新列添加到 ITEM 表中。它将新列名称设置为 FULL\_NAME，数据类型为 string，长度为 50 个字符。此表达式连接两个现有列 LAST\_NAME 和 FIRST\_NAME 的值，以便计算结果为 FULL\_NAME。schema-name、table-name 和表达式参数引用的是源数据库表中的对象。Value 和 data-type 块引用的是目标数据库表中的对象。

```
{
  "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
      }
    }
  ]
}
```

## 使用表达式标记目标记录

要将目标表中的记录标记为在源表中插入、更新或删除，请在转换规则中使用表达式。表达式使用 operation\_indicator 函数来标记记录。从源中删除的记录不会从目标中删除。而是使用用户提供的值标记目标记录，以指示该记录已从源中删除。

**Note**

`operation_indicator` 函数仅适用于源数据库和目标数据库中具有主键的表。

例如，以下转换规则首先将新的 `Operation` 列添加到目标表中。然后，只要从源表中删除记录，它就会更新具有值 `D` 的列。

```
{
  "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
  }
}
```

## 使用表达式复制源表标头

默认情况下，源表的标头不会复制到目标。要指示要复制哪些标头，请将转换规则与包含表列标头的表达式结合使用。

您可以在表达式中使用以下列标头。

标题	正在进行的复制中的值	完全加载中的值	数据类型
AR_H_STRE AM_POSITION	来自源的流位置值。 此值可能是系统更改号 ( SCN ) 或日志序	空字符串。	string

标题	正在进行的复制中的值	完全加载中的值	数据类型
	列号 ( LSN ) , 具体取决于源端点。		
AR_H_TIMESTAMP	指示更改时间的时间戳。	指示数据到达目标的当前时间的时间戳。	DATETIME ( 小数位数 = 7 )
AR_H_COMMIT_TIMESTAMP	指示提交时间的时间戳。	指示当前时间的时间戳。	DATETIME ( 小数位数 = 7 )
AR_H_OPERATION	INSERT、UPDATE 或 DELETE	INSERT	string
AR_H_USER	源提供的有关进行更改的用户的用户名、ID 或任何其他信息。  此标头仅在 SQL Server 和 Oracle ( 版本 11.2.0.3 及更高版本 ) 源端点上受到支持。	要应用于对象的转换。转换规则操作区分大小写。	string
AR_H_CHANGE_SEQ	源数据库中唯一的递增数字，由时间戳和自动递增数字组成。该值取决于源数据库系统。	空字符串。	string

以下示例通过使用源中的流位置值向目标添加新列。对于 SQL Server，流位置值是源端点的 LSN。对于 Oracle，流位置值是源端点的 SCN。

```
{
  "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
    }
  }
}

```

以下示例向目标添加了一个新列，该列具有来自源的唯一递增数字。此值表示任务级别的 35 位唯一数字。前 16 位数字是时间戳的一部分，后 19 位是由 DBMS 递增的 record\_id 数字。

```

{
  "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
  }
}

```

## 使用 SQLite 函数构建表达式

您可以使用表设置指定要应用于指定操作的选定表或视图的任何设置。表设置规则是可选的。



**Note**

MongoDB 和 DocumentDB 数据库没有表和视图的概念，而是将数据记录存储为文档，这些文档聚集在集合中。因此，从 MongoDB 或 DocumentDB 源迁移时，对于所选集合（而不是表和视图），请考虑范围分段类型的并行加载设置。

**主题**

- [使用 CASE 表达式](#)
- [示例](#)

接下来，您可以找到可用于构建转换规则表达式的字符串函数。

字符串函数	描述
<code>lower(x)</code>	<code>lower(x)</code> 函数返回字符串 <code>x</code> 的副本，且所有字符都转换为小写。默认情况下，内置的 <code>lower</code> 函数仅适用于 ASCII 字符。
<code>upper(x)</code>	<code>upper(x)</code> 函数返回字符串 <code>x</code> 的副本，且所有字符都转换为大写。默认情况下，内置的 <code>upper</code> 函数仅适用于 ASCII 字符。
<code>ltrim(x,y)</code>	<code>ltrim(x,y)</code> 函数返回一个字符串，删除了 <code>y</code> 中出现的 <code>x</code> 左侧的所有字符。如果 <code>y</code> 没有值，则 <code>ltrim(x)</code> 删除 <code>x</code> 左侧的空格。
<code>replace(x,y,z)</code>	<code>replace(x,y,z)</code> 函数返回一个字符串，将字符串 <code>x</code> 中每次出现的字符串 <code>y</code> 替换为字符串 <code>z</code> 。
<code>rtrim(x,y)</code>	<code>rtrim(x,y)</code> 函数返回一个字符串，删除了 <code>y</code> 中出现的 <code>x</code> 右侧的所有字符。如果 <code>y</code> 没有值，则 <code>rtrim(x)</code> 删除 <code>x</code> 右侧的空格。
<code>substr(x,y,z)</code>	<code>substr(x,y,z)</code> 函数返回输入字符串 <code>x</code> 的子字符串，该子字符串从第 <code>y</code> 个字符开始，长度为 <code>z</code> 个字符。  如果省略 <code>z</code> ，则 <code>substr(x,y)</code> 返回字符串 <code>x</code> 中从第 <code>y</code> 个字符到结尾的全部字符。 <code>x</code> 最左侧的字符为数字 1。如果 <code>y</code> 为负数，则子字符串的第一个字符将从右侧，而不是左侧开始计数。如果 <code>z</code> 为负数，则返回第 <code>y</code> 个字符之前的 <code>abs(z)</code> 个字符。如果 <code>x</code> 是字

字符串函数	描述
	字符串，则字符索引指向实际的 UTF-8 字符。如果 $x$ 是 BLOB，则索引指向字节。
<code>trim(x,y)</code>	<code>trim(x,y)</code> 函数返回一个字符串，删除了 $y$ 中出现的 $x$ 两侧的所有字符。如果 $y$ 没有值，则 <code>trim(x)</code> 删除 $x$ 两侧的空格。

接下来，您可以找到可用于构建转换规则表达式的 LOB 函数。

LOB 函数	描述
<code>hex(x)</code>	<code>hex</code> 函数接收 BLOB 作为参数，并返回 BLOB 内容的大写十六进制字符串版本。
<code>randblob ( N)</code>	<code>randblob( N)</code> 函数返回包含伪随机字节的 $N$ 字节 BLOB。如果 $N$ 小于 1，则返回 1 字节的随机 BLOB。
<code>zeroblob(N)</code>	<code>zeroblob( N)</code> 函数返回由 $N$ 个字节 0x00 组成的 BLOB。

接下来，您可以找到可用于构建转换规则表达式的数字函数。

数字函数	描述
<code>abs(x)</code>	<code>abs(x)</code> 函数返回数字参数 $x$ 的绝对值。如果 $x$ 为 NULL，则 <code>abs(x)</code> 函数返回 NULL。如果 $x$ 是无法转换为数值的字符串或 BLOB，则 <code>abs(x)</code> 函数返回 0.0。
<code>random( )</code>	<code>random</code> 函数返回介于 -9,223,372,036,854,775,808 与 +9,223,372,036,854,775,807 之间的伪随机整数。
<code>round ( x,y)</code>	<code>round ( x,y)</code> 函数返回一个浮点值，将 $x$ 四舍五入到小数点右边第 $y$ 位。如果 $y$ 没有值，则假定为 0。
<code>max ( x,y...)</code>	多参数 <code>max</code> 函数返回具最大值的参数，如果任何参数均为 NULL，则返回 NULL。

数字函数	描述
	<p><code>max</code> 函数从左到右搜索其参数，以寻找定义排序函数的参数。如果找到一个，则使用该排序函数对所有字符串进行比较。如果 <code>max</code> 没有任何参数定义排序函数，则使用 <code>BINARY</code> 排序函数。当具有两个或更多参数时，<code>max</code> 函数是一个简单的函数，但如果只有一个参数，则该函数作为聚合函数运行。</p>
<code>min (x,y...)</code>	<p>多参数 <code>min</code> 函数返回具有最小值的参数。</p> <p><code>min</code> 函数从左到右搜索其参数，以寻找定义排序函数的参数。如果找到一个，则使用该排序函数对所有字符串进行比较。如果 <code>min</code> 没有任何参数定义排序函数，则使用 <code>BINARY</code> 排序函数。当具有两个或更多参数时，<code>min</code> 函数是一个简单的函数，但如果只有一个参数，则该函数作为聚合函数运行。</p>

接下来，您可以找到可用于构建转换规则表达式的 `NULL` 校验函数。

NULL 校验函数	描述
<code>coalesce (x,y...)</code>	<p><code>coalesce</code> 函数返回其第一个非 <code>NULL</code> 参数的副本，但如果所有参数均为 <code>NULL</code>，则返回 <code>NULL</code>。<code>coalesce</code> 函数至少具有两个参数。</p>
<code>ifnull(x,y)</code>	<p><code>ifnull</code> 函数返回其第一个非 <code>NULL</code> 参数的副本，但如果两个参数均为 <code>NULL</code>，则返回 <code>NULL</code>。<code>ifnull</code> 函数只有两个参数。<code>ifnull</code> 函数与具有两个参数的 <code>coalesce</code> 函数相同。</p>
<code>nullif(x,y)</code>	<p>如果参数不同，则 <code>nullif(x,y)</code> 函数返回其第一个参数的副本，但如果参数相同，则返回 <code>NULL</code>。</p> <p><code>nullif(x,y)</code> 函数从左到右搜索其参数，以寻找定义排序函数的参数。如果找到一个，则使用该排序函数对所有字符串进行比较。如果两个参数都没有定义排序函数，则使用 <code>BINARY</code> 排序函数。</p>

接下来，您可以找到可用于构建转换规则表达式的日期和时间函数。

日期和时间函数	描述
<code>date(<i>timestring</i> , <i>modifier</i>, <i>modifier</i>...)</code>	date 函数以 YYYY-MM-DD 格式返回日期。
<code>time(<i>timestring</i> , <i>modifier</i>, <i>modifier</i>...)</code>	time 函数以 HH:MM:SS 格式返回时间。
<code>datetime( <i>timestring</i> , <i>modifier</i>, <i>modifier</i>...)</code>	datetime 函数以 YYYY-MM-DD HH:MM:SS 格式返回日期和时间。
<code>julianday( <i>timestring</i> <i>g</i> , <i>modifier</i>, <i>modifier</i>...)</code>	julianday 函数返回自公元前 4714 年 11 月 24 日格林威治中午以来的天数。
<code>strftime( <i>format</i>, <i>timestring</i> , <i>modifier</i>, <i>modifier</i>...)</code>	<p>strftime 函数使用以下变量之一，按照第一个参数指定的格式字符串返回日期：</p> <ul style="list-style-type: none"> <li>%d：一月中的第几天</li> <li>%H：小时（00-24）</li> <li>%f:** 小数形式的秒数 SS.SSS</li> <li>%j：一年中的第几天（001-366）</li> <li>%J：** Julian 天数</li> <li>%m：月（01-12）</li> <li>%M：分钟（00-59）</li> <li>%s：自 1970 年 1 月 1 日以来的秒数</li> <li>%S：秒（00-59）</li> <li>%w：一周中的第几天（0-6，星期日 = 0）</li> <li>%W：一年中的第几周（00-53）</li> <li>%Y：年（0000-9999）</li> </ul>

日期和时间函数	描述
	%%: %

接下来，您可以找到可用于构建转换规则表达式的哈希函数。

哈希函数	描述
hash_sha256( <i>x</i> )	<p>hash 函数为输入列生成哈希值（使用 SHA-256 算法），并返回生成的哈希值的十六进制值。</p> <p>要在表达式中使用 hash 函数，可将 hash_sha256( <i>x</i> ) 添加到表达式中，并将 <i>x</i> 替换为源列名。</p>

## 使用 CASE 表达式

SQLite CASE 表达式计算条件列表，并根据结果返回一个表达式。语法如下所示。

```

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

```

## 示例

### Example 使用 case 条件向目标表添加新的字符串列

以下示例转换规则将新的字符串列 `emp_seniority` 添加到目标表 `employee`。该示例对工资列运用 SQLite `round` 函数，并采用 `case` 条件检查工资是否等于或超过 20,000。如果条件满足，则该列获得值 `SENIOR`，而其他任何列获得值 `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 向目标表添加新的日期列

以下示例将新的日期列 `createdate` 添加到目标表 `employee`。如果使用 SQLite 日期函数 `datetime`，则对于插入的每一行，都会将日期添加到新创建的表中。

```
{
  "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 向目标表添加新的数字列

以下示例将新的数字列 `rounded_emp_salary` 添加到目标表 `employee`。该示例使用 SQLite `round` 函数添加四舍五入后的工资。

```

{
  "rule-type": "transformation",
  "rule-id": "2",
  "rule-name": "2",
  "rule-action": "add-column",
  "rule-target": "column",
  "object-locator": {
    "schema-name": "public",
    "table-name": "employee"
  },
  "value": "rounded_emp_salary",
  "expression": "round($emp_salary)",
  "data-type": {
    "type": "int8"
  }
}

```

### Example 使用哈希函数向目标表添加新的字符串列

以下示例将新的字符串列 `hashed_emp_number` 添加到目标表 `employee`。SQLite `hash_sha256(x)` 函数在目标上为源列 `emp_number` 创建哈希值。

```

{
  "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
  }
}

```

## 使用表达式向目标表添加元数据

可以使用以下表达式，将元数据信息添加到目标表：

- \$AR\_M\_SOURCE\_SCHEMA – 源架构的名称。
- \$AR\_M\_SOURCE\_TABLE\_NAME – 源表的名称。
- \$AR\_M\_SOURCE\_COLUMN\_NAME – 源表中的列的名称。
- \$AR\_M\_SOURCE\_COLUMN\_DATATYPE – 源表中列的数据类型。

Example 使用源的架构名称为架构名称添加列

以下示例使用源的架构名称，将名为 schema\_name 的新列添加到目标。

```

{
  "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
  }
}

```



## 表和集合设置规则和操作

可以使用表设置，指定要应用于指定操作的选定表或视图的任何设置。表设置规则是可选的，具体取决于端点和迁移要求。

MongoDB 和 Amazon DocumentDB 数据库不使用表和视图，而是将数据记录存储为文档，这些文档聚集在集合中。任何 MongoDB 或 Amazon DocumentDB 端点的单个数据库都是由数据库名称标识的一组特定集合。

从 MongoDB 或 Amazon DocumentDB 源迁移时，使用并行加载设置的方式略有不同。在这种情况下，对于所选集合（而不是表和视图），请考虑自动分段或范围分段类型的并行加载设置。

### 主题

- [表格设置中的通配符受到限制](#)
- [对所选表、视图和集合使用并行加载](#)
- [为所选表或视图指定 LOB 设置](#)
- [表设置示例](#)

对于使用表设置规则类型的表映射规则，您可以应用以下参数。

参数	可能的值	描述
rule-type	table-settings	一个将规则应用于选定规则所指定的表、视图或集合的值。
rule-id	一个数字值。	一个用于标识规则的唯一数字值。
rule-name	一个字母数字值。	一个用于标识规则的唯一名称。
object-locator	一个带以下参数的对象： <ul style="list-style-type: none"> <li>• schema-name – 架构的名称。对于 MongoDB 和 Amazon DocumentDB 端点，这是存放一组集合的数据库的名称。</li> </ul>	特定架构和表或视图的名称，或特定数据库和集合的名称（无通配符）。

参数	可能的值	描述
	<ul style="list-style-type: none"><li>• <code>table-name</code> – 表、视图或集合的名称。</li></ul>	

参数	可能的值	描述
parallel-load	<p>一个带以下参数的对象：</p> <ul style="list-style-type: none"> <li>• <code>type</code> – 指定是否启用并行加载。</li> </ul> <p>如果是，则此参数还指定用于标识要并行加载的表或视图分区、子分区或其他段的机制。分区是已在源表或视图中按名称定义和标识的段。</p> <p>对于 MongoDB 和亚马逊 DocumentDB 终端节点，分区是分段。AWS DMS 可以根据关联的自动分段参数自动计算这些值。或者，可以使用范围分段参数手动指定这些参数。</p> <p>仅限 Oracle 端点，子分区是已在源表或视图中按名称定义和标识的其他级别的段。您可以通过在一个或多个表或视图列的值范围中指定边界，在 <code>table-settings</code> 规则中标识其他段。</p> <ul style="list-style-type: none"> <li>• <code>partitions</code> – 当 <code>type</code> 为 <code>partitions-list</code> 时，此值指定要并行加载的所有分区。</li> <li>• <code>subpartitions</code> – 仅限 Oracle 端点，当 <code>type</code> 为 <code>partitions-list</code> 时，此值指定要并行加载的所有子分区。</li> <li>• <code>columns</code> – 当 <code>type</code> 为 <code>ranges</code> 时，此值指定用于标识要并行加载的基于范围的段的列的名称。</li> <li>• <code>boundaries</code> – 当 <code>type</code> 为 <code>ranges</code> 时，此值指定用于标识</li> </ul>	<p>该值指定由 <code>object-locator</code> 选项标识的表或视图上的并行加载（多线程）操作。在这种情况下，您可以通过以下任意方式并行加载：</p> <ul style="list-style-type: none"> <li>• 按所有可用分区或子分区指定的段。</li> <li>• 按选定的分区和子分区。</li> <li>• 按您指定的自动分段或基于范围的段。</li> </ul> <p>有关并行加载的更多信息，请参阅<a href="#">对所选表、视图和集合使用并行加载</a>。</p>

参数	可能的值	描述
	要并行加载的基于范围的段的 columns 的值。	

参数	可能的值	描述
type	<p>parallel-load 的以下操作之一：</p> <ul style="list-style-type: none"> <li>• <code>partitions-auto</code> – 所有表或视图的分区均为并行加载。每个分区都分配到自己的线程。</li> </ul> <p>对于 MongoDB 和 Amazon DocumentDB 源端点而言，这是使用并行完全加载的自动分段选项所必需的设置。</p> <ul style="list-style-type: none"> <li>• <code>subpartitions-auto</code> – (仅限 Oracle 端点) 表或视图的所有子分区均为并行加载。每个子分区都分配到自己的线程。</li> <li>• <code>partitions-list</code> – 表或视图的所有指定分区均为并行加载。仅限 Oracle 端点，表或视图的所有指定子分区均为并行加载。您指定的每个分区和子分区都分配到自己的线程。您可以通过分区名称 (<code>partitions</code>) 和子分区名称 (<code>subpartitions</code>) 指定要并行加载的分区和子分区。</li> <li>• <code>ranges</code> – 表、视图或集合的所有指定范围的段均为并行加载。您标识的每个表、视图或集合段都分配到自己的线程。您可以按列名 (<code>columns</code>) 和列值 (<code>boundaries</code>) 指定这些段。</li> </ul> <p>PostgreSQL 端点仅支持此类型的并行加载。作为源端点的 MongoDB 和 Amazon DocumentDB 既支持此范围分段类型，也支</p>	<p>用于标识要并行加载的表、视图或集合分区、子分区或段的机制。</p>

参数	可能的值	描述
	<p>持并行完全加载的自动分段类型 ( <code>partitions-auto</code> )。</p> <ul style="list-style-type: none"> <li>• <code>none</code> – 表、视图或集合在单线程任务 ( 默认值 ) 中加载，而不论其分区或子分区如何。有关更多信息，请参阅 <a href="#">创建任务</a>。</li> </ul>	
<code>number-of-partitions</code>	( 可选 ) 当 MongoDB 或 Amazon DocumentDB 端点的指定集合的 <code>type</code> 为 <code>partitions-auto</code> 时，此参数指定用于迁移的分区 ( 段 ) 的总数。默认值是 16。	指定要并行加载的分区的确切数量。
<code>collection-count-from-metadata</code>	( 可选 ) 当 <code>type</code> MongoDB 或 Amazon DocumentDB 终端节点的指定集合且此参数设置为 <code>AWS DMS</code> ，则使用估计的集合计数 <code>true</code> 来确定分区的数量。 <code>partitions-auto</code> 如果此参数设置为 <code>false</code> ，则 <code>AWS DMS</code> 使用实际收集次数。默认值为 <code>true</code> 。	指定是使用估计的集合计数，还是实际的集合计数来计算要并行加载的分区数量。
<code>max-records-skip-per-page</code>	( 可选 ) 当 MongoDB 或 Amazon DocumentDB 端点的指定集合的 <code>type</code> 为 <code>partitions-auto</code> 时，此参数为确定每个分区边界时要同时跳过的记录数。 <code>AWS DMS</code> 采用分页跳过法来确定分区的最小边界。默认值是 10000。	指定在确定每个分区的边界时要同时跳过的记录数。设置一个相对默认值较大的值，可能会导致游标超时和任务失败。设置一个相对默认值较低的值，会导致每页的操作次数增加，完全加载速度降低。

参数	可能的值	描述
batch-size	( 可选 ) 当 MongoDB 或 Amazon DocumentDB 端点的指定集合的 type 为 partitions-auto 时，此整数值会限制一个往返批处理中返回的文档数。如果批处理大小为零 ( 0 )，则游标使用服务器定义的最大批处理大小。默认值是 0。	指定一个批处理中返回的最大文档数。每个批处理需要往返一次服务器。
partitions	当 type 为 partitions-list 时，这是一个字符串数组，用于指定并行加载的分区名称。	要并行加载的分区名称。
subpartitions	( 仅限 Oracle 端点 ) 当 type 为 partitions-list 时，这是一个字符串数组，用于指定并行加载的子分区名称。	要并行加载的子分区名称。
columns	当 type 为 ranges 时，这是设置为列名称的字符串数组，用于标识要并行加载的基于范围的表、视图或集合段。	用于标识要并行加载的基于范围的表、视图或集合段的列名称。
boundaries	当 type 为 ranges 时，这是列值数组的数组。每个列值数组都包含 columns 指定的数量和顺序的列值。列值数组指定表、视图或集合段的上边界。每个附加列值数组都会为一个附加表、视图或集合段添加上边界。所有此类基于范围的表、视图或集合段均为并行加载。	用于标识要并行加载的基于范围的表、视图或集合分区的列值。

参数	可能的值	描述
lob-settings	<p>一个带以下参数的对象：</p> <ul style="list-style-type: none"><li>• mode – 指定 LOB 的迁移处理模式。</li><li>• bulk-max-size – 指定 LOB 的最大大小，具体取决于 mode 设置。</li></ul>	<p>一个值，指定由 object-locator 选项标识的表或视图的 LOB 处理。指定的 LOB 处理仅覆盖此表或视图的任何任务 LOB 设置。有关使用 LOB 设置参数的更多信息，请参阅<a href="#">为所选表或视图指定 LOB 设置</a>。</p>



参数	可能的值	描述
mode	<p>使用以下值指定所选表或视图中 LOB 的迁移处理：</p> <ul style="list-style-type: none"><li>• <b>limited</b> – (默认) 此值将迁移设置为有限的 LOB 模式，所有 LOB 与表或视图中的所有其他列数据类型一起进行内联迁移。在复制大多数小 LOB (100 MB 或更小) 时使用此值。此外，指定一个 <code>bulk-max-size</code> 值 (0 无效)。所有大于 <code>bulk-max-size</code> 的已迁移 LOB 都将截断为您设置的大小。</li><li>• <b>unlimited</b> – 此值将迁移设置为完整 LOB 模式。当要复制的所有或大多数 LOB 大于 1 GB 时，请使用此值。如果您指定的 <code>bulk-max-size</code> 值为零，则所有 LOB 都在标准完整 LOB 模式中迁移。在这种形式的 <code>unlimited</code> 模式中，会使用源表或视图中的查找将所有 LOB 与其他列数据类型分开迁移。如果您指定一个大于零的 <code>bulk-max-size</code> 值，则所有 LOB 都在组合完整 LOB 模式中迁移。在这种形式的 <code>unlimited</code> 模式中，会使用源表或视图查找迁移大于 <code>bulk-max-size</code> 的 LOB，类似于标准完整 LOB 模式。否则，小于等于此大小的 LOB 将内联迁移，类似于有限 LOB 模式。无论您使用何种形式，都不会在 <code>unlimited</code> 模式下截断 LOB。</li><li>• <b>none</b> – 所有表或视图 LOB 都会根据任务 LOB 设置迁移。</li></ul>	用于迁移 LOB 的机制。

参数	可能的值	描述
	<p>有关任务 LOB 设置的更多信息，请参阅<a href="#">目标元数据任务设置</a>。</p> <p>有关如何迁移 LOB 以及如何指定这些任务 LOB 设置的更多信息，请参阅<a href="#">在任务中为源数据库设置 LOB 支持 AWS DMS</a>。</p>	
bulk-max-size	此值的影响取决于 mode。	LOB 的最大大小（以 KB 为单位）。仅当您需要复制较小 LOB 或目标端点不支持无限制的 LOB 大小时，才指定此选项。

## 表格设置中的通配符受到限制

源数据库不支持在 "table-settings" 规则中使用百分比通配符（"%"），如下所示。

```
{
  "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
  }
}
```

如果您在 "table-settings" 规则 "%" 中使用如图所示，则 AWS DMS 返回以下异常。

```
Error in mapping rules. Rule with ruleId = x failed validation. Exact schema and table name required when using table settings rule.
```

此外，AWS 建议您不要使用单个任务加载大量的大型集合parallel-load。请注意，AWS DMS 通过 MaxFullLoadSubTasks 任务设置参数值（最大值为 49），限制资源争用以及并行加载的段数。

相反，可通过分别指定每个 "schema-name" 和 "table-name"，为最大的集合指定源数据库的所有集合。此外，可适当扩大迁移规模。例如，在足够数量的复制实例上运行多个任务，以处理数据库中的大量大型集合。

## 对所选表、视图和集合使用并行加载

要加速迁移并提高其效率，可以对选定的表、视图和集合使用并行加载。换言之，可以并行使用多个线程迁移单个分段的表、视图或集合。为此，将满负荷 AWS DMS 任务拆分为线程，并将每个表段分配给自己的线程。

使用此并行加载过程，您可以首先让多个线程从源端点并行卸载多个表、视图和集合。然后，可以迁移多个线程，并将相同的表、视图和集合并行加载到目标端点。对于某些数据库引擎，可以按现有分区或子分区对表和视图进行分段。对于其他数据库引擎，您可以根据特定参数 AWS DMS 自动对集合进行细分（自动分段）。除此之外，可以按指定的列值范围，对任何表、视图或集合进行分段。

以下源端点支持并行加载：

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2 LUW
- SAP Adaptive Server Enterprise ( ASE )
- MongoDB ( 仅支持并行完全加载的自动分段和范围分段选项 )
- Amazon DocumentDB ( 仅支持并行完全加载的自动分段和范围分段选项 )

对于 MongoDB 和 Amazon DocumentDB 终端节点 AWS DMS，支持以下数据类型，这些列是并行满载的范围分割选项的分区键。

- Double

- String
- ObjectId
- 32 位整数
- 64 位整数

以下目标端点支持用于表设置规则的并行加载：

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Amazon S3
- SAP Adaptive Server Enterprise ( ASE )
- Amazon Redshift
- MongoDB ( 仅支持并行完全加载的自动分段和范围分段选项 )
- Amazon DocumentDB ( 仅支持并行完全加载的自动分段和范围分段选项 )
- Db2 LUW

要指定并行加载的表和视图的最大数，请使用 `MaxFullLoadSubTasks` 任务设置。

要为并行加载任务的支持目标指定每个表或视图的最大线程数，可使用列值边界定义更多的段。

#### Important

`MaxFullLoadSubTasks` 控制要并行加载的表或表段的数量。`ParallelLoadThreads` 控制迁移任务用于执行并行加载的线程数。这些设置为相乘关系。因此，完全加载任务期间使用的线程总数大约为 `ParallelLoadThreads` 的值乘以 `MaxFullLoadSubTasks` 的值 ( `ParallelLoadThreads * MaxFullLoadSubtasks` ) 。

如果您创建的任务包含大量完全加载子任务和大量并行加载线程，则任务可能会消耗过多的内存并失败。

要为 Amazon DynamoDB、Amazon Kinesis Data Streams、Apache Kafka 或 Amazon Elasticsearch Service 目标指定每个表的最大线程数，可使用 `ParallelLoadThreads` 目标元数据任务设置。

要指定使用 `ParallelLoadThreads` 时并行加载任务的缓冲区大小，可使用 `ParallelLoadBufferSize` 目标元数据任务设置。

`ParallelLoadThreads` 和 `ParallelLoadBufferSize` 的可用性和设置取决于目标端点。

有关 `ParallelLoadThreads` 和 `ParallelLoadBufferSize` 设置的更多信息，请参阅[目标元数据任务设置](#)。有关 `MaxFullLoadSubTasks` 设置的更多信息，请参阅[完全加载任务设置](#)。有关特定于目标端点的信息，请参阅相关主题。

要使用并行加载，您可以使用 `parallel-load` 选项创建一个类型为 `table-settings` 的表映射规则。在 `table-settings` 规则中，您可以为要并行加载的单个表、视图或集合指定分段条件。要执行此操作，请将 `parallel-load` 选项的 `type` 参数设置为多个选项之一。

如何执行此操作取决于您希望如何为并行加载分段表、视图或集合：

- 按分区（或段）– 使用 `partitions-auto` 类型加载所有现有表或视图分区（或段）。或者使用具指定分区数组的 `partitions-list` 类型，仅加载选定的分区。

仅适用于 MongoDB 和 Amazon DocumentDB 终端节点，按区段加载所有或指定的集合 AWS DMS，这些集合也使用 `partitions-auto` 类型和其他可选参数自动计算。 `table-settings`

- （仅限 Oracle 端点）按子分区 – 使用 `subpartitions-auto` 类型加载所有现有的表或视图子分区。或者使用具指定 `subpartitions` 数组的 `partitions-list` 类型，仅加载选定的子分区。
- 按您定义的段 – 加载使用列值边界定义的表、视图或集合段。为此，请使用具有指定 `columns` 和 `boundaries` 数组的 `ranges` 类型。

#### Note

PostgreSQL 端点仅支持此类型的并行加载。作为源端点的 MongoDB 和 Amazon DocumentDB 既支持此范围分段类型，也支持并行完全加载的自动分段类型（`partitions-auto`）。

要标识要并行加载的其他表、视图或集合，请使用 `parallel-load` 选项指定其他 `table-settings` 对象。

在以下过程中，您可以了解如何为每种并行加载类型编写 JSON 代码（从最简单到最复杂）。

## 指定所有表、视图或集合分区，或所有表或视图子分区

- 使用 `partitions-auto` 类型或 `subpartitions-auto` 类型（但不能同时使用二者）指定 `parallel-load`。

然后，每个表、视图或集合分区（或段）或子分区会自动分配到自己的线程。

对于某些端点，并行加载仅包括已为表或视图定义的分区或子分区。对于 MongoDB 和 Amazon DocumentDB 源终端节点，您可以 AWS DMS 根据可选的附加参数自动计算分区（或分段）。这包括 `number-of-partitions`、`collection-count-from-metadata`、`max-records-skip-per-page` 和 `batch-size`。

## 指定选定表或视图的分区和/或子分区

1. 使用 `partitions-list` 类型指定 `parallel-load`。
2. （可选）通过将分区名称数组指定为 `partitions` 的值来包含分区。

然后，每个指定的分区将分配到自己的线程。

### Important

对于 Oracle 端点，在选择分区和子分区进行并行加载时，请确保分区和子分区不重叠。如果使用重叠的分区和子分区并行加载数据，则会复制条目，或者由于主键重复冲突而失败。

3. （可选）仅限 Oracle 端点，可通过将子分区名称数组指定为 `subpartitions` 的值来包含子分区。

然后，每个指定的子分区将分配到自己的线程。

### Note

并行加载仅包括已为表或视图定义的分区或子分区。

您可以指定表或视图段作为列值范围。执行此操作时，请注意以下列特征：

- 指定索引列可显著提高性能。
- 您可以指定最多 10 个列。

- 您不能使用列来定义具有以下 AWS DMS 数据类型的分段边界：  
DOUBLE、FLOAT、BLOB、CLOB 和 NCLOB
- 不会复制具有 null 值的记录。

### 指定表、视图或集合段作为列值范围

1. 使用 `ranges` 类型指定 `parallel-load`。
2. 通过列名数组指定为 `columns` 的值来定义表或视图段之间的边界。对要为其定义表或视图段之间边界的每个列执行此操作。

列的顺序很重要。在定义每个边界时，第一列是最重要的，最后一列最不重要，如下所述。

3. 通过将边界数组指定为 `boundaries` 值来定义所有表或视图段的数据范围。边界数组是一个列值数组的数组。要实现此目的，请执行以下步骤：
  - a. 将列值数组的每个元素指定为与每个列对应的值。列值数组 代表您要定义的每个表或视图段的上边界。按照您在 `columns` 数组中指定该列的顺序指定每个列。

以源支持的格式输入 DATE 列的值。

- b. 按顺序将每个列值数组指定为从表或视图的底部到各 `next-to-top` 段的上边界。如果您指定的顶部边界上方存在任何行，则这些行将填充表或视图的顶部段。因此，基于范围的段的数量可能比边界数组中的段边界的数量多一个。每个这样的基于范围的段被分配到自己的线程。

将会复制所有非 null 数据，即使您没有为表或视图中的所有列定义数据范围也是如此。

例如，假设您为列 COL1、COL2 和 COL3 定义三个列-值数组，如下所示。

COL1	COL2	COL3
10	30	105
20	20	120
100	12	99

您就可能为总共四个段定义了三个段边界。

为了标识要为每个段复制的行范围，复制实例会对以下四个段中的每个段的这三个列应用搜索。搜索类似于以下内容：

#### 段 1

在以下情况下复制所有行：前两列值小于或等于其对应段 1 上边界值。此外，第三列的值小于其段 1 上边界值。

#### 段 2

在以下情况下复制所有行（段 1 行除外）：前两列值小于或等于其对应段 2 上边界值。此外，第三列的值小于其段 2 上边界值。

#### 段 3

在以下情况下复制所有行（段 2 行除外）：前两列值小于或等于其对应段 3 上边界值。此外，第三列的值小于其段 3 上边界值。

#### 段 4

复制所有剩余的行（除了段 1、2 和 3 行）。

在这种情况下，复制实例会创建一个 WHERE 子句来加载每个段，如下所示：

#### 段 1

```
((COL1 < 10) OR ((COL1 = 10) AND (COL2 < 30)) OR ((COL1 = 10) AND (COL2 = 30) AND (COL3 < 105)))
```

#### 段 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)))
```

#### 段 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)))
```



## 段 4

```
NOT ((COL1 < 100) OR ((COL1 = 100) AND (COL2 < 12)) OR ((COL1 = 100)
AND (COL2 = 12) AND (COL3 < 99)))
```

### 为所选表或视图指定 LOB 设置

您可以通过使用一个或多个 table-settings 对象的 lob-settings 选项创建 table-settings 类型的表映射规则来为一个或多个表设置任务 LOB 设置。

以下源端点支持为所选表或视图指定 LOB 设置：

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2，具体取决于 mode 和 bulk-max-size 设置，如下所述
- SAP Adaptive Server Enterprise (ASE)，具体取决于 mode 和 bulk-max-size 设置，如下所述

以下目标端点支持为所选表或视图指定 LOB 设置：

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- SAP ASE，具体取决于 mode 和 bulk-max-size 设置，如下所述

#### Note

您只能将 LOB 数据类型用于包含主键的表或视图。

要对选定的表或视图使用 LOB 设置，请使用 lob-settings 选项创建类型为 table-settings 的表映射规则。这样做会指定由 object-locator 选项标识的表或视图的 LOB 处理。在 table-settings 规则中，您可以指定一个具有以下参数的 lob-settings 对象：

- `mode` – 指定处理所选表或视图的 LOB 迁移的机制，如下所示：
  - `limited` – 此默认受限 LOB 模式是最快且最有效的模式。仅当所有 LOB 都较小（大小不超过 100 MB）或目标端点不支持无限制 LOB 大小时，才使用此模式。此外，如果使用 `limited`，则所有 LOB 都必须为 `bulk-max-size` 设置的大小范围内。

在完全加载任务的此模式下，复制实例将所有 LOB 与其他列数据类型一起进行内联迁移，作为主表或视图存储的一部分。但是，实例会将大于您的 `bulk-max-size` 值的任何迁移 LOB 截断为指定的大小。对于更改数据捕获（CDC）加载任务，实例使用源表查找迁移所有 LOB，就像在标准完整 LOB 模式下一样（请参阅下文）。

#### Note

您只能为完全加载任务迁移视图。

- `unlimited` – 完整 LOB 模式的迁移机制取决于您为 `bulk-max-size` 设置的值，如下所示：
  - 标准完整 LOB 模式 – 如果将 `bulk-max-size` 设置为零，则复制实例将使用标准完整 LOB 模式迁移所有 LOB。此模式需要在源表或视图中进行查找，以便迁移每个 LOB，无论大小如何都是如此。此方法通常会比对于有限 LOB 模式迁移慢得多。仅当所有或大多数 LOB 都较大（1 GB 或更大）时才使用此模式。
  - 组合完整 LOB 模式 – 如果将 `bulk-max-size` 设置为非零值，则此完整 LOB 模式使用有限 LOB 模式和标准完整 LOB 模式的组合。这适用于完全加载任务，如果 LOB 大小在您的 `bulk-max-size` 值范围内，则实例会在有限 LOB 模式下内联迁移 LOB。如果 LOB 大小大于此值，则实例使用源表或视图查找迁移 LOB，就像在标准完整 LOB 模式下一样。对于更改数据捕获（CDC）加载任务，实例使用源表查找迁移所有 LOB，就像在标准完整 LOB 模式下一样（请参阅下文）。无论 LOB 大小如何，它都会这样做。

#### Note

您只能为完全加载任务迁移视图。

此模式导致迁移速度处于更快的有限 LOB 模式和更慢的标准完整 LOB 模式之间。只有当混合使用较小和较大 LOB 并且大多数 LOB 较小时才使用此模式。

此组合完整 LOB 模式仅适用于以下端点：

- IBM Db2 作为源
- SAP ASE 作为源或目标

无论您为 unlimited 模式指定的机制如何，实例都会完全迁移所有 LOB，而不会截断。

- none – 复制实例使用任务 LOB 设置迁移所选表或视图中的 LOB。使用此选项来帮助比较对于所选表或视图，在具有和不具有 LOB 设置时的迁移结果。

如果指定的表或视图包含复制中包含的 LOB，则只能在使用 limited LOB 模式时将 BatchApplyEnabled 任务设置指定为 true。

在某些情况下，您可以将 true 设置为 BatchApplyEnabled，将 BatchApplyPreserveTransaction 设置为 false。在这些情况下，如果表或视图具有 LOB 且源和目标端点是 Oracle，实例会将 BatchApplyPreserveTransaction 设置为 true。

- bulk-max-size – 将此值设置为零或非零值（以 KB 为单位），具体取决于 mode，如之前几项所述。在 limited 模式中，您必须为此参数设置非零值。

实例将 LOB 转换为二进制格式。因此，要指定需要复制的最大 LOB，请将其大小乘以 3。例如，如果您的最大 LOB 为 2 MB，请将 bulk-max-size 设置为 6,000 (6 MB)。

## 表设置示例

下面，您可以找到一些演示如何使用表设置的示例。

### Example 加载按分区分段的表

以下示例通过基于其所有分区并行加载源中的 SALES 表来更有效地加载它。

```
{
  "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 加载按子分区分段的表

以下示例通过基于其所有子分区并行加载 Oracle 源中的 SALES 表来更有效地加载它。

```
{
  "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 加载按分区列表分段的表

以下示例通过按特定分区列表并行加载源中的 SALES 表来加载它。在此，指定的分区以用字母的部分开头的值命名，例如 ABCD、EFGH 等等。

```
{
  "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 加载按选定的分区和子分区列表分段的 Oracle 表

以下示例通过按选定分区和子分区列表并行加载 Oracle 源中的 SALES 表来加载它。在此，指定的分区以用字母的部分开头的值命名，例如 ABCD、EFGH 等等。指定的子分区用以数字开头的值命名，例如 01234 和 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 加载按列值范围分段的表

以下示例通过按 SALES\_NO 和 REGION 值的范围指定的段并行加载源中的 SALES 表来加载它。

```
{
  "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"
        ]
      ]
    }
  }
]
```

在此，为具有名称 SALES\_NO 和 REGION 的段范围指定了两列。使用两组列值 ( ["1000", "NORTH"] 和 ["3000", "WEST"] ) 指定两个边界。

因此，这两个边界标识要并行加载的以下三个表段：

### 段 1

SALES\_NO 小于或等于 1000 且 REGION 小于“NORTH”的行。换句话说，EAST 区域最高为 1,000 的销售数量。

### 段 2

段 1 以外 SALES\_NO 小于或等于 3000 且 REGION 小于“WEST”的行。换句话说，NORTH 和 SOUTH 区域超过 1,000 且不超过 3,000 的销售数量。

### 段 3

段 1 和段 2 以外的所有剩余行。换句话说，WEST 区域超过 3000 的销售数量。

Example 加载两个表：一个按范围分段，另一个按分区分段

以下示例按您标识的段边界并行加载 SALES 表。它还通过所有分区并行加载 ORDERS 表，如前面的示例所示。

```
{
  "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 使用有限 LOB 模式加载具有 LOB 的表

以下示例使用受限 LOB 模式 ( 默认 ) 加载源中包括 LOB 的 ITEMS 表，最大不截断大小为 100 MB。任何大于此大小的 LOB 都将截断为 100 MB。所有 LOB 都与所有其他列数据类型一起内联加载。

```

{
  "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 使用标准完整 LOB 模式加载具有 LOB 的表

以下示例使用标准完整 LOB 模式加载源中的 ITEMS 表，包括所有 LOB 而不截断。所有 LOB，无论大小，都通过在源表中对每个 LOB 使用查找与其他数据类型分开加载。

```

{
  "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 使用组合完整 LOB 模式加载具有 LOB 的表

以下示例使用组合完整 LOB 模式加载源中的 ITEMS 表，包括所有 LOB 而不截断。所有大小为 100 MB 的 LOB 都与其他数据类型一起内联加载，就像在有限 LOB 模式下一样。所有大小超过 100 MB 的 LOB 从其他数据类型单独加载。此单独加载为源表中的每个此类 LOB 使用一个查找，类似于标准的完整 LOB 模式中。

```

{
  "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 使用任务 LOB 设置加载具有 LOB 的表

以下示例使用其任务 LOB 设置加载源中的 ITEMS 表，包括所有 LOB。100 MB 的 bulk-max-size 设置会被忽略，仅用于快速重置为 limited 或 unlimited 模式。

```
{
  "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"
    }
  }
]
```

## 使用源筛选器

您可以使用源筛选器来限制从源传输到目标的记录的数量和类型。例如，您可以指定仅将总部员工移动到目标数据库。筛选器是选择规则的一部分。还可以对数据列应用筛选器。

源筛选器必须遵循以下约束：

- 选择规则可以不包含筛选器，也可以包含一个或多个筛选器。
- 每个筛选器可具有一个或多个筛选条件。

- 如果使用多个筛选器，则将合并一系列筛选器，就像在筛选器之间使用 AND 运算符一样。
- 如果在单个筛选器中使用多个筛选条件，则将合并一系列筛选条件，就像在筛选条件之间使用 OR 运算符一样。
- 筛选器仅在 `rule-action = 'include'` 时应用。
- 筛选器需要一个列名和一系列筛选条件。筛选条件必须具有与一个值或两个值关联或者不与值关联的筛选运算符，具体取决于运算符。
- 列名、表名、视图名和架构名都区分大小写。Oracle 和 Db2 应始终使用大写字母。
- 筛选器仅支持具有确切名称的表。筛选器不支持通配符。

以下限制适用于使用源筛选器：

- 过滤器不计算 right-to-left 语言列。
- 不要将筛选器应用于 LOB 列。
- 仅将筛选器应用于不可变列，这些列在创建后不会更新。如果将源筛选器应用于可变列（可在创建后更新），可能会导致不良行为。

例如，如果筛选器在某列中排除或包含特定行，则会始终排除或包含这些指定的行，即使该行稍后发生了更改。假设在 A 列中排除或包含第 1-10 行，之后这些行发生了更改，变为了第 11-20 行。在这种情况下，即使数据已经不同，这些行仍旧排除或包含在内。

同样，假设不在筛选器范围内的某行稍后进行了更新（或者更新并删除），此时按照筛选器的定义应当排除或包含在内。在这种情况下，它会在目标上复制。

使用源过滤器时，还需要考虑以下其他问题：

- 我们建议您使用筛选定义中包含的列和主键创建索引。

## 使用 JSON 格式创建源筛选器规则

您可以使用选择规则的 JSON `filters` 参数创建源筛选器。`filters` 参数指定包含一个或多个 JSON 对象的数组。每个对象都具有用于指定源筛选类型、列名称和筛选条件的参数。这些筛选条件包含一个或多个筛选运算符和筛选值。

下表显示了用于在 `filters` 对象中指定源筛选的参数。

参数	值
filter-type	source
column-name	要将筛选器应用于的源列的名称。该名称区分大小写。
filter-conditions	一个或多个对象的数组，其中包含一个 filter-operator 参数和 Zero 或多个关联的值参数，具体取决于 filter-operator 值。
filter-operator	<p>具有以下值之一的参数：</p> <ul style="list-style-type: none"> <li>lte – 小于等于某个值</li> <li>ste – 小于等于某个值 ( lte 别名 )</li> <li>gte – 大于等于某个值</li> <li>eq – 等于某个值</li> <li>noteq – 不等于某个值</li> <li>between – 等于两个值或介于两个值之间</li> <li>notbetween – 不等于两个值或介于两个值之间</li> <li>null – NULL 值</li> <li>notnull – 非 NULL 值</li> </ul>
value 或 start-value 和 end-value 或 无值	<p>0 或与 filter-operator 关联的多个值参数：</p> <ul style="list-style-type: none"> <li>如果 filter-operator 为 lte、ste、gte、eq 或 noteq，则使用 value 指定一个值参数。</li> <li>如果 filter-operator 为 between 或 notbetween，则使用 start-value 和 end-value 指定两个值参数。</li> <li>如果 filter-operator 为 null 或 notnull，则不指定值参数。</li> </ul>

以下示例说明了几种常见的使用源筛选器的方法。

#### Example 单一筛选器

以下筛选器将 `empid >= 100` 的所有员工复制到目标数据库中。

```
{
  "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 多个筛选运算符

以下筛选器将多个筛选运算符应用于单个数据列。筛选器将 (empid <= 10) OR (empid is between 50 and 75) OR (empid >= 100) 的所有员工复制到目标数据库中。

```
{
  "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 多个筛选器

以下筛选器将多个筛选条件应用于表中的两个列。筛选器将 (empid <= 100) AND (dept = tech) 的所有员工复制到目标数据库中。

```

{
  "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 筛选 NULL 值

以下筛选器显示如何根据空值进行筛选。它会将所有 dept = NULL 的员工复制到目标数据库。

```

{
  "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 使用 NOT 运算符筛选

有些运算符可以按否定形式使用。以下筛选器将 (empid is < 50) OR (empid is > 75) 的所有员工复制到目标数据库中。

```

{
  "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 使用混合筛选运算符

从 3.5.0 AWS DMS 版开始，您可以混合使用包含运算符和负运算符。

以下筛选器将 (empid != 50) AND (dept is not NULL) 的所有员工复制到目标数据库中。

```
{
  "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"  
    }]  
  }]  
}]  
}
```

将 null 与其他筛选运算符一起使用时，请注意以下几点：

- 如果在同一个筛选器中同时使用包含、否定和 null 筛选条件，则不会复制具有 NULL 值的记录。
- 如果在同一个筛选器中同时使用否定和 null 筛选条件，但没有包含筛选条件，则不会复制任何数据。
- 如果使用否定筛选条件但未明确设置 null 筛选条件，则不会复制具有 NULL 值的记录。

## 按时间和日期筛选

在选择要导入的数据时，您可以指定日期或时间作为筛选条件的一部分。AWS DMS 使用日期格式 YYYY-MM-DD 和时间格式 YYYY-MM-DD HH: MM: SS 进行过滤。AWS DMS 比较函数遵循 SQLite 的约定。有关 SQLite 数据类型和日期比较的更多信息，请参阅 SQLite 文档中的 [SQLite 版本 3 中的数据类型](#)。

以下筛选器显示了如何基于日期进行筛选。它会将所有 empstartdate >= January 1, 2002 的员工复制到目标数据库。

### Example 单一日期筛选器

```
{  
  "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"
      }]
    }]
  }]
}
```

## 为任务启用和使用迁移前评估

迁移前评估会评估数据库迁移任务的指定组件，以帮助确定可能导致迁移任务无法按预期运行的任何问题。此评估使您有机会在运行新的或修改后的任务之前识别和修复问题。这使您可以避免因缺少要求或已知限制而导致的任务失败相关的延迟。

AWS DMS 提供两种不同的移民前评估选项：

- 数据类型评估：提供有限评估范围的传统报告。
- 迁移前评估运行：包含各种类型的个人评估，包括数据类型评估结果。

### Note

如果您选择进行迁移前评估，则无需单独选择数据类型评估。

以下主题对这些选项进行了介绍：

- [指定、启动和查看迁移前评估运行](#)：迁移前（推荐）评估运行指定根据新的或现有的迁移任务配置运行一项或多项单独评估。每项评估都会从迁移类型、支持的对象、索引配置和其他任务设置（例如标识要迁移的架构和表的表映射）等标准的角度评估支持的源数据库和/或目标数据库的特定元素。

例如，个人评估可能会根据 AWS DMS 引擎版本评估哪些源数据类型或主键格式可以迁移或不能迁移。您可以使用 AWS DMS 管理控制台或使用和软件开发工具包访问 AWS DMS API 来启动和查看最新评估运行的结果，AWS CLI 并查看任务之前所有评估运行的结果。您还可以在 Amazon S3 存储桶中查看先前对任务运行的评估结果，该存储桶是您选择 AWS DMS 用来存储这些结果的。

**Note**

可用单独评估的数量和类型可能会随着时间的推移而增加。有关定期更新的更多信息，请参阅[指定单独评估](#)。

- [启动和查看数据类型评估 \(旧版\)](#)：数据类型 (旧版) 评估以单个 JSON 结构返回单一类型的迁移前评估的结果：在支持的关系源数据库实例中可能无法正确迁移的数据类型。此报告返回源数据库中选定进行迁移的每个架构和表中发现的所有有问题的数据类型的结果。

## 为迁移前评估创建先决条件

本节介绍创建迁移前评估所需的 Amazon S3 和 IAM 资源。

### 创建 S3 存储桶

AWS DMS 将迁移前评估报告存储在 S3 存储桶中。要创建 S3 存储桶，请执行以下操作：

1. 登录 AWS Management Console 并打开亚马逊 S3 控制台，[网址为 https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/)。
2. 选择创建存储桶。
3. `##### dms-bucket-yoursignin#`
4. AWS 区域 为 DMS 迁移任务选择。
5. 保持其余设置不变，然后选择创建存储桶。

### 创建 IAM 资源

DMS 使用 IAM 角色和策略访问 S3 存储桶以存储迁移前评估结果。

要创建 IAM 策略，请执行以下操作：

1. 登录 AWS Management Console 并打开 IAM 控制台，[网址为 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/)。
2. 在导航窗格中，选择策略。
3. 选择创建策略。
4. 在创建策略页面中，选择 JSON 选项卡。

5. 将以下 JSON 代码粘贴到编辑器中，替换示例代码。将 *my-bucket* 替换为您在上一节中创建的 Amazon S3 存储桶的名称。

```
{
  "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. 选择“下一步：标签”，然后选择“下一步：查看”。
7. 在名称\* 中输入 **DMSPremigrationAssessmentS3Policy**，然后选择创建策略。

要创建 IAM 角色，请执行以下操作：

1. 在 IAM 控制台的导航窗格中，选择角色。
2. 选择创建角色。
3. 在选择可信实体页面中，在可信实体类型下选择 AWS 服务。对于其他 AWS 服务的用例，请选择 DMS。

4. 选中 DMS 复选框，然后选择下一步。
5. 在添加权限页面上，选择 DMS PremigrationAssessment S 3Policy。选择下一步。
6. 在命名、检查并创建页面上，在角色名称中输入 **DMSPremigrationAssessmentS3Role**，然后选择创建角色。
7. 在角色页面上，在角色名称中输入 **DMSPremigrationAssessmentS3Role**。选择 DMS PremigrationAssessment S 3Role。
8. 在 DMS PremigrationAssessment S3Role 页面上，选择信任关系选项卡。选择编辑信任策略。
9. 在编辑信任策略页面上，将以下 JSON 粘贴到编辑器中，替换现有文本。

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

此策略授予 DMS 将迁移前评估运行结果放入 S3 存储桶的 `sts:AssumeRole` 权限。

10. 选择更新策略。

## 指定、启动和查看迁移前评估运行

迁移前评估根据新的或现有的迁移任务配置指定要运行的一项或多项个人评估。每项单独评估都会根据迁移类型、支持的对象、索引配置和其他任务设置（例如标识要迁移的架构和表的表映射）等考虑因素来评估源或目标数据库的特定元素。例如，个人评估可能会评估哪些源数据类型或主键格式可以迁移，哪些主键格式可以迁移，哪些不能迁移。

### 指定单独评估

在创建新的评估时，您可以选择运行适用于您的任务配置的部分或全部个人评估。

AWS DMS 支持对以下关系源和目标数据库引擎运行迁移前评估：

- [甲骨文评估](#)
- [Sql 服务器评估](#)
- [MySQL 评估](#) ( 包括兼容 MariaDB 和亚马逊 Aurora MySQL 的版本 )
- [PostgreSQL 评估](#) ( 包括兼容亚马逊 Aurora PostgreSQL 的版本 )

## 启动和查看迁移前评估运行

您可以使用 AWS DMS 管理控制台、和 AWS DMS API 开始为新的或现有的迁移任务运行迁移前评估。AWS CLI

要为新任务或现有任务启动迁移前评估运行

1. 在 AWS DMS 管理控制台的数据库迁移任务页面上，执行下列操作之一：
    - 要创建新任务并对其进行评估，请选择创建任务。创建数据库迁移任务页面打开：
      1. 输入创建任务所需的任务设置，包括表格映射。
      2. 在“迁移前评估”部分，选中“运行迁移前评估”复选框。此页面包含用于为新任务指定评估运行的选项。
-  **Note**

创建新任务时，启用迁移前评估运行会禁用在创建任务时自动启动任务的选项。评估运行完成后，您可以手动启动该任务。
- 要评估现有任务，请在数据库迁移任务页面上选择现有任务的标识符。所选现有任务的任务页面打开：
    1. 选择操作，然后选择创建迁移前评估。创建预迁移评估页面打开，其中包含用于为现有任务指定评估运行的选项。
2. 为您的评估运行输入一个唯一的名称，或保留默认值。
  3. 选择要包含在该评估运行中的可用单独评估。您只能根据当前的任务设置选择可用的单独评估。在默认情况下，已启用并选中所有可用的单独评估。
  4. 在您的账户中搜索并选择一个 Amazon S3 存储桶和文件夹，以存储您的评估结果报告。有关为评估运行设置资源的信息，请参阅[为迁移前评估创建先决条件](#)。
  5. 选择或输入对您选择的 Amazon S3 存储桶和文件夹具有完全账户访问权限的 IAM 角色。有关为评估运行设置资源的信息，请参阅[为迁移前评估创建先决条件](#)。



6. 也可以选择相应设置，以对 Amazon S3 存储桶中的评估结果报告进行加密。有关 S3 存储桶加密的信息，请参阅为 [Amazon S3 存储桶设置默认服务器端加密行为](#)。
7. 为新任务选择创建任务或为现有任务选择创建。

数据库迁移任务页面打开，其中列出您的新任务或修改后的任务（其状态为正在创建...）以及一条横幅消息，表示创建任务后将开始您的迁移前评估运行。

AWS DMS 允许访问使用 AWS DMS 管理控制台、或 AWS DMS API 运行的最新和所有之前的 AWS CLI 迁移前评估。

### 要查看评估运行的结果

1. 在 AWS DMS 管理控制台中，在数据库迁移任务页面上选择现有任务的标识符。将打开现有任务的任务页面。
2. 在现有任务页面上选择预迁移评估选项卡。这将打开该页面上的“移民前评估”部分，按时间倒序按名称列出的评估结果。最新结果显示在列表顶部。选择要查看其结果的评估运行的名称。

这些评估运行结果先列出最新评估运行的名称及其状态概述，然后列出指定的单独评估及其状态。然后，可以在列表中选择每项评估的名称以浏览其状态的详细信息，可用结果显示在表格的列中。

评估运行的状态概述和每项单独评估都显示一个状态值。该值指示评估运行的总体状态以及每项单独评估的类似状态。以下是评估运行的状态值列表：

- "cancelling" – 评估运行已取消。
- "deleting" – 评估运行已删除。
- "failed" – 至少一项单独评估已完成，状态为 failed。
- "error-provisioning" – 预置资源时（provisioning 状态期间）发生内部错误。
- "error-executing" – 运行单独评估时（running 状态期间）发生内部错误。
- "invalid state" – 评估运行处于未知状态。
- "passed" – 所有单独评估均已完成，没有一个评估处于 failed 状态。
- "provisioning" – 正在预置运行单独评测所需的资源。
- "running" – 正在运行单独评估。
- "starting" – 评测运行已开始，但尚未为单独评测预置资源。
- "warning" – 至少一项单独评估已完成，状态为 warning。

以下是评估运行中每一项单独评估的状态值列表：

- "cancelled" – 作为取消评估运行的一部分，单独评估被取消。
- "error" – 单独评估未成功完成。
- "failed" – 单独评估成功完成，但验证结果未通过：查看结果的详细信息以获取更多信息。
- "invalid state" – 单独评估处于未知状态。
- "passed" – 单独评估已完成，验证结果成功。
- "pending" – 单独评估正在等待运行。
- "running" – 单独评估正在运行中。
- "warning" – 单独评估成功完成，但验证结果为警告：查看结果的详细信息以获取更多信息。

您还可以在 Amazon S3 上查看评估运行结果的 JSON 文件。

在 Amazon S3 上查看评估运行的 JSON 文件

1. 在 AWS DMS 管理控制台中，选择评估运行状态概述中显示的 Amazon S3 存储桶链接。这将显示存储桶文件夹和存储在存储桶中的其他 Amazon S3 对象的列表。如果您的结果存储在存储桶文件夹中，请打开该文件夹。
2. 您可以在多个 JSON 文件中查找您的评估运行结果。summary.json 文件包含评估运行的总体结果。其余文件均以为评估运行指定的单独评估命名，例如 unsupported-data-types-in-source.json。这些文件都包含所选评估运行的相应单独评估的结果。

要启动并查看现有迁移任务的迁移前评估运行结果，您可以运行以下 CLI 命令和 AWS DMS API 操作：

- CLI : [describe-applicable-individual-assessments](#) , API : [DescribeApplicableIndividualAssessments](#) – 提供您可以为新的迁移前评估运行指定的单独评估列表，并给出一个或多个任务配置参数。
- CLI : [start-replication-task-assessment-run](#) , API : [StartReplicationTaskAssessmentRun](#) – 为现有迁移任务的一个或多个单独评估启动新的迁移前评估运行。
- CLI : [describe-replication-task-assessment-runs](#) , API : [DescribeReplicationTaskAssessmentRuns](#) – 根据筛选器设置返回迁移前评估运行的分页列表。

- CLI : [describe-replication-task-individual-assessments](#) , API : [DescribeReplicationTaskIndividualAssessments](#) – 根据筛选器设置返回单独评估的分页列表。
- CLI : [cancel-replication-task-assessment-run](#) , API : [CancelReplicationTaskAssessmentRun](#) – 取消但不删除单个迁移前评估运行。
- CLI : [delete-replication-task-assessment-run](#) , API : [DeleteReplicationTaskAssessmentRun](#) – 删除单次迁移前评估运行的记录。

## 个人评估

本节介绍个人移民前评估。

要使用 AWS DMS API 创建个人预迁移评估，请使用列出的 API 密钥作为 [StartReplicationTaskAssessmentRun](#) 操作 `IncludeOnly` 参数。

### 主题

- [针对所有终端节点类型的评估](#)
- [甲骨文评估](#)
- [Sql 服务器评估](#)
- [MySQL 评估](#)
- [MariaDB 评估](#)
- [PostgreSQL 评估](#)

## 针对所有终端节点类型的评估

本节介绍所有端点类型的单独迁移前评估。

### 主题

- [不支持的数据类型](#)
- [使用大型对象 \(LOB\) 但目标 LOB 列不可为空](#)
- [包含大型对象 \(LOB\) 但没有主键或唯一约束的源表](#)
- [不带主键的源表，仅适用于 CDC 或满负荷和 CDC 任务](#)
- [不带主键的目标表仅适用于 CDC 任务](#)
- [不支持的源主键类型-复合主键](#)

## 不支持的数据类型

API 密钥：`unsupported-data-types-in-source`

检查 DMS 不支持的源端点中的数据类型。并非所有数据类型都可以在引擎之间迁移。

使用大型对象 (LOB) 但目标 LOB 列不可为空

API 密钥：`full-lob-not-nullable-at-target`

当复制使用完整 LOB 模式或内联 LOB 模式时，检查目标中 LOB 列是否可为空。使用这些 LOB 模式时，DMS 要求 LOB 列为空。此评估要求源数据库和目标数据库必须是关联的。

包含大型对象 (LOB) 但没有主键或唯一约束的源表

API 密钥：`table-with-lob-but-without-primary-key-or-unique-constraint`

检查是否存在含 LOB 但没有主键或唯一键的源表。表必须具有主键或唯一键，DMS 才能迁移 LOB。此评估要求源数据库必须是关系数据库。

不带主键的源表，仅适用于 CDC 或满负荷和 CDC 任务

API 密钥：`table-with-no-primary-key-or-unique-constraint`

检查源表中是否存在主键或唯一键，以进行满载和更改数据捕获 (CDC) 迁移或仅限 CDC 的迁移。缺少主键或唯一密钥可能会导致 CDC 迁移期间出现性能问题。此评估要求源数据库为关系数据库，迁移类型必须包含 CDC。

不带主键的目标表仅适用于 CDC 任务

API 密钥：`target-table-has-unique-key-or-primary-key-for-cdc`

检查已创建的目标表是否具有主键或唯一键以进行仅 CDC 的迁移。当 DMS 应用更新和删除时，缺少主键或唯一键可能会导致目标系统中出现全表扫描。这可能会导致 CDC 迁移期间出现性能问题。此评估要求目标数据库为关系数据库，迁移类型必须包含 CDC。

不支持的源主键类型-复合主键

API 密钥：`unsupported-source-pk-type-for-elasticsearch-target`

迁移到 Amazon OpenSearch 服务时，会检查源表中是否存在复合主键。源表的主键必须由单个列组成。此评估要求源数据库为关系数据库，目标数据库必须为 DynamoDB。

**Note**

DMS 支持将源数据库迁移到源主键由多列组成的 OpenSearch 服务目标。

## 甲骨文评估

本节介绍使用 Oracle 源端点的迁移任务的单独迁移前评估。

**Note**

要使用本节中的迁移前评估，您必须向 `dms_user` 添加下列权限：

```
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;
```

有关将 Oracle 作为源时的权限的更多信息，请参阅 [在自行管理的 Oracle 源上需要用户帐户权限 AWS DMS](#)。

## 主题

- [选中数据库级别补充日志记录](#)
- [验证是否为备用数据库创建了所需的数据库链接](#)
- [Oracle 验证 LOB 数据类型以及是否已配置二进制读取器](#)
- [验证数据库是否为 CDB](#)
- [查看 Oracle 数据库版本](#)
- [DMS 验证 Oracle CDC 的方法](#)
- [DMS 验证 Oracle RAC 配置](#)

- [验证 DMS 用户是否拥有目标权限](#)
- [验证是否所有列都需要补充日志](#)
- [验证是否对具有主键或唯一键的表启用了补充日志记录](#)
- [验证是否有 SecureFile LOB 并且任务已配置为完整 LOB 模式](#)
- [验证任务范围中包含的表中是否使用基于函数的索引。](#)
- [验证是否正在对任务范围中包含的表使用全局临时表。](#)
- [验证任务范围中包含的表上是否使用了按索引组织的带有溢出段的表。](#)
- [验证任务范围中包含的表是否使用了多级嵌套表。](#)
- [验证任务范围中包含的表上是否使用了不可见列。](#)
- [验证任务范围中包含的表是否使用了基于 ROWID 列的实例化视图。](#)
- [验证是否使用了 Active Data Guard DML 重定向功能。](#)
- [验证是否使用混合分区表。](#)
- [验证是否使用了仅限架构的 Oracle 帐户](#)
- [验证是否使用了虚拟列](#)
- [验证在任务作用域中定义的表名是否包含撇号。](#)
- [验证任务作用域中定义的列是否具有 XMLTypeLong、或 Long Raw 数据类型，并在任务设置中验证 LOB 模式配置。](#)
- [验证源 Oracle 版本是否受支持 AWS DMS。](#)
- [验证目标 Oracle 版本是否受支持 AWS DMS。](#)
- [验证目标 Oracle 版本是否受支持 AWS DMS。](#)
- [验证 DMS 用户是否具有使用数据验证所需的权限。](#)
- [验证 DMS 用户是否有权在 Oracle ASM 中使用二进制读取器](#)
- [验证 DMS 用户是否有权在 Oracle 非 ASM 中使用二进制读取器](#)
- [验证 DMS 用户是否有权使用带 CopyToTempFolder 方法的二进制读取器](#)
- [验证 DMS 用户是否有权使用 Oracle 备用服务器作为来源](#)
- [验证 DMS 源是否已连接到应用程序容器 PDB](#)
- [验证表的范围中是否包含 XML 数据类型。](#)
- [验证源数据库上是否启用了归档日志模式。](#)
- [验证 RDS Oracle 的存档日志保留期。](#)
- [验证表的范围中是否包含扩展数据类型。](#)
- [验证任务范围中包含的对象名称的长度。](#)

- [验证 DMS 源是否已连接到 Oracle PDB](#)
- [验证表的任务范围中是否包含空间列。](#)
- [验证 DMS 源是否已连接到 Oracle 备用服务器。](#)
- [验证源数据库表空间是否使用 TDE 加密。](#)
- [验证源数据库是否为 Oracle ASM](#)

### 选中数据库级别补充日志记录

API 密钥 : oracle-supplemental-db-level

此迁移前评估将验证是否在数据库级别启用了最低限度的补充日志记录。必须启用补充日志记录才能使用 Oracle 数据库作为迁移源。

要启用补充日志记录，使用以下查询：

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
```

有关更多信息，请参阅 [设置补充日志记录](#)。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

### 验证是否为备用数据库创建了所需的数据库链接

API 密钥 : oracle-validate-standby-dblink

此迁移前评估将验证是否为 Oracle 备用数据库源创建 Dblink。AWSDMS\_DBLINK 是使用备用数据库作为源的先决条件。使用 Oracle Standby 作为源时，默认情况下 AWS DMS 不验证未完成事务。

有关更多信息，请参阅 [使用自行管理的 Oracle 数据库作为数据源 AWS DMS](#)。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

### Oracle 验证 LOB 数据类型以及是否已配置二进制读取器

API 密钥 : oracle-binary-lob-source-validation

此迁移前评估将验证 Oracle LogMiner 是否用于版本 12c 或更高版本的 Oracle 数据库终端节点。AWS DMS 不支持 Oracle LogMiner 从 Oracle 数据库版本 12c 迁移 LOB 列。该评估还会检查 LOB 列是否存在，并提供适当的建议。

要将您的迁移配置为不使用 Oracle LogMiner，请将以下配置添加到您的源终端节点：

```
useLogMinerReader=N;useBfile=Y;
```


有关更多信息，请参阅 [在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器](#)。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

验证数据库是否为 CDB

API 密钥：oracle-validate-cdb

此迁移前评估会验证数据库是否为容器数据库。AWS DMS 不支持多租户容器根数据库（CDB \$ROOT）。

 Note

只有 Oracle 版本 12.1.0.1 或更高版本需要进行此评估。此评估不适用于 12.1.0.1 之前的 Oracle 版本。

有关更多信息，请参阅 [使用 Oracle 作为来源的限制 AWS DMS](#)。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

查看 Oracle 数据库版本

API 密钥：oracle-check-cdc-support-express-edition

此迁移前评估会验证 Oracle 源数据库是否为 Express Edition。对于 Oracle Express Edition（Oracle Database XE）版本 18.0 及更高版本，AWS DMS 不支持 CDC。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

DMS 验证 Oracle CDC 的方法

API 密钥：oracle-recommendation-cdc-method

此迁移前评估验证了过去七天的重做日志生成，并建议是使用 AWS DMS Binary Reader 还是 Oracle LogMiner for CDC。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。



有关决定使用哪种 CDC 方法的更多信息，请参阅[在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器](#)。

## DMS 验证 Oracle RAC 配置

API 密钥：`oracle-check-rac`

此迁移前评估会验证 Oracle 数据库是否为 Real Application Cluster。必须正确配置 Real Application Cluster 数据库。如果数据库基于 RAC，我们建议您使用适用于 CDC 的 AWS DMS 二进制阅读器，而不是 Oracle LogMiner。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅[在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器](#)。

## 验证 DMS 用户是否拥有目标权限

API 密钥：`oracle-validate-permissions-on-target`

此迁移前评估将验证 DMS 用户是否拥有目标数据库的所有必需权限。

## 验证是否所有列都需要补充日志

API 密钥：`oracle-validate-supplemental-logging-all-columns`

对于任务范围中提到的表，此迁移前评估可以验证是否已向没有主键或唯一键的表的所有列添加了补充日志记录。如果不对缺少主键或唯一键的表的所有列进行补充日志，则重做日志中将无法显示数据 before-and-after 图像。DMS 需要对没有主键或唯一键的表进行补充日志才能生成 DML 语句。

## 验证是否对具有主键或唯一键的表启用了补充日志记录

API 密钥：`oracle-validate-supplemental-logging-for-pk`

此迁移前评估将验证是否为具有主键或唯一索引的表启用了补充日志记录，还会检查 `AddSupplementalLogging` 是否在端点级别启用了补充日志记录。为确保 DMS 可以复制更改，您可以根据主键或唯一键在表级别上手动添加补充日志记录，也可以使用终端节点设置 `AddSupplementalLogging = true`，让 DMS 用户对任何复制的表拥有 ALTER 权限。

## 验证是否有 SecureFile LOB 并且任务已配置为完整 LOB 模式

API 密钥：`oracle-validate-securefile-lob`

此迁移前评估检查任务范围内的表中是否存在 SecureFile LOB，并验证其 LOB 设置。需要注意的是，目前仅在 F SecureFile ULL LOB 模式下支持 LOB。考虑将 LOB 表分配给单独的任务以提高性能，因为在完整 LOB 模式下运行任务可能会导致性能降低。

验证任务范围中包含的表中是否使用基于函数的索引。

API 密钥 : `oracle-validate-function-based-indexes`

此迁移前评估检查任务范围内的表上是否存在基于函数的索引。请注意，AWS DMS 这不支持复制基于函数的索引。考虑在迁移目标数据库后创建索引。

验证是否正在对任务范围中包含的表使用全局临时表。

API 密钥 : `oracle-validate-global-temporary-tables`

此迁移前评估检查任务表映射范围内是否使用了全局临时表。请注意，AWS DMS 这不支持迁移或复制全局临时表。

验证任务范围中包含的表上是否使用了按索引组织的带有溢出段的表。

API 密钥 : `oracle-validate-iot-overflow-segments`

验证任务范围中包含的表上是否使用了按索引组织的带有溢出段的表。AWS DMS 不支持 CDC 用于按索引组织的带有溢出段的表。

验证任务范围中包含的表是否使用了多级嵌套表。

API 密钥 : `oracle-validate-more-than-one-nesting-table-level`

此迁移前评估检查任务范围上使用的嵌套表的嵌套级别。AWS DMS 仅支持一层表嵌套。

验证任务范围中包含的表上是否使用了不可见列。

API 密钥 : `oracle-validate-invisible-columns`

此迁移前评估可验证任务范围中使用的表是否有不可见的列。AWS DMS 不会迁移源数据库中不可见列的数据。要迁移不可见的列，您需要将其修改为可见。

验证任务范围中包含的表是否使用了基于 ROWID 列的实例化视图。

API 密钥 : `oracle-validate-rowid-based-materialized-views`

此迁移前评估可验证迁移中使用的实例化视图是否是基于 ROWID 列创建的。AWS DMS 不支持 ROWID 数据类型或基于 ROWID 列的实例化视图。

验证是否使用了 Active Data Guard DML 重定向功能。

API 密钥 : `oracle-validate-adg-redirect-dml`

此迁移前评估可验证是否使用了 Active Data Guard DML 重定向功能。使用 Oracle 19.0 作为源代码时，AWS DMS 不支持 Data Guard DML 重定向功能。

验证是否使用混合分区表。

API 密钥 : `oracle-validate-hybrid-partitioned-tables`

此迁移前评估可验证混合分区表是否用于任务范围中定义的表。

验证是否使用了仅限架构的 Oracle 帐户

API 密钥 : `oracle-validate-schema-only-accounts`

此迁移前评估可验证任务范围内是否存在仅限架构的帐户。

验证是否使用了虚拟列

API 密钥 : `oracle-validate-virtual-columns`

此迁移前评估将验证 Oracle 实例在任务范围内的表中是否有虚拟列。

验证在任务作用域中定义的表名是否包含撇号。

API 密钥 : `oracle-validate-names-with-apostrophes`

此迁移前评估可验证任务范围中使用的表是否包含撇号。AWS DMS 不会复制名称中包含撇号的表。如果确定了这些表，可以考虑重命名这些表。或者，您可以创建不带撇号的视图或实例化视图来加载这些表。

验证任务作用域中定义的列是否具有 `XMLTypeLong`、或 `Long Raw` 数据类型，并在任务设置中验证 LOB 模式配置。

API 密钥 : `oracle-validate-limited-lob-mode-for-longs`

此迁移前评估将验证任务范围内定义的表是否具有数据类型 `XMLTypeLongLong Raw`、或，并检查任务设置是否配置为使用有限大小 LOB 模式。AWS DMS 不支持使用 FULL LOB 模式复制这些数据类型。在识别具有此类数据类型的表时，可以考虑将任务设置更改为使用有限大小 LOB 模式。

验证源 Oracle 版本是否受支持 AWS DMS。

API 密钥 : `oracle-validate-supported-versions-of-source`

此迁移前评估将验证源 Oracle 实例版本是否受支持。AWS DMS

验证目标 Oracle 版本是否受支持 AWS DMS。

API 密钥 : `oracle-validate-supported-versions-of-target`

此迁移前评估将验证目标 Oracle 实例版本是否受支持。AWS DMS

验证目标 Oracle 版本是否受支持 AWS DMS。

API 密钥 : `oracle-validate-supported-versions-of-target`

此迁移前评估将验证目标 Oracle 实例版本是否受支持。AWS DMS

验证 DMS 用户是否具有使用数据验证所需的权限。

API 密钥 : `oracle-prerequisites-privileges-of-validation-feature`

此迁移前评估将验证 DMS 用户是否具有使用 DMS 数据验证的必要权限。如果您不打算使用数据验证，则可以忽略启用此验证。

验证 DMS 用户是否有权在 Oracle ASM 中使用二进制读取器

API 密钥 : `oracle-prerequisites-privileges-of-binary-reader-asm`

此迁移前评估将验证 DMS 用户是否具有在 Oracle ASM 实例上使用 Binary Reader 的必要权限。如果您的来源不是 Oracle ASM 实例，或者您没有使用适用于 CDC 的二进制阅读器，则可以忽略启用此评估。

验证 DMS 用户是否有权在 Oracle 非 ASM 中使用二进制读取器

API 密钥 : `oracle-prerequisites-privileges-of-binary-reader-non-asm`

此迁移前评估将验证 DMS 用户是否具有在 Oracle 非 ASM 实例上使用 Binary Reader 的必要权限。只有当您拥有 Oracle 非 ASM 实例时，此评估才有效。

验证 DMS 用户是否有权使用带 CopyToTempFolder 方法的二进制读取器

API 密钥 : `oracle-prerequisites-privileges-of-binary-reader-copy-to-temp-folder`

此迁移前评估将验证 DMS 用户是否具有使用“复制到临时文件夹”方法的 Binary Reader 所需的权限。只有当您计划在使用二进制读取器时使用 CopyToTempFolder 读取 CDC 更改并且将 ASM 实例连接

到源代码时，此评估才有意义。如果您不打算使用该 CopyToTempFolder 功能，则可以忽略启用此评估。

我们建议不要使用该 CopyToTempFolder 功能，因为它已被弃用。

验证 DMS 用户是否有权使用 Oracle 备用服务器作为来源

API 密钥：`oracle-prerequisites-privileges-of-standby-as-source`

此迁移前评估将验证 DMS 用户是否具有使用 StandBy Oracle 实例作为源的必要权限。如果您不打算使用 StandBy Oracle 实例作为来源，则可以忽略启用此评估。

验证 DMS 源是否已连接到应用程序容器 PDB

API 密钥：`oracle-check-app-pdb`

此迁移前评估可验证 DMS 源是否已连接到应用程序容器 PDB。DMS 不支持从应用程序容器 PDB 进行复制。

验证表的范围中是否包含 XML 数据类型。

API 密钥：`oracle-check-xml-columns`

此迁移前评估可验证任务范围内使用的表是否具有 XML 数据类型。当表包含 XML 数据类型时，它还会检查任务是否配置为受限 LOB 模式。DMS 仅支持有限的 LOB 模式来迁移 Oracle XML 列。

验证源数据库上是否启用了归档日志模式。

API 密钥：`oracle-check-archivelog-mode`

此迁移前评估可验证源数据库上是否启用了归档日志模式。DMS 需要在源数据库上启用存档日志模式才能复制更改。

验证 RDS Oracle 的存档日志保留期。

API 密钥：`oracle-check-archivelog-retention-rds`

此迁移前评估将验证您的 RDS Oracle 数据库上的归档日志保留期是否配置为至少 24 小时。

验证表的范围中是否包含扩展数据类型。

API 密钥：`oracle-check-extended-columns`

此迁移前评估可验证任务范围内使用的表是否具有扩展的数据类型。请注意，只有 DMS 版本 3.5 及更高版本支持扩展数据类型。

验证任务范围中包含的对象名称的长度。

API 密钥：`oracle-check-object-30-bytes-limit`

此迁移前评估可验证对象名称的长度是否超过 30 字节。DMS 不支持长对象名（超过 30 字节）。

验证 DMS 源是否已连接到 Oracle PDB

API 密钥：`oracle-check-pdb-enabled`

此迁移前评估可验证 DMS 源是否已连接到 PDB。仅当使用以 Oracle PDB 作为源的二进制读取器时，DMS 才支持 CDC。评估还会评估任务是否配置为在 DMS 连接到 Oracle PDB 时使用二进制读取器。

验证表的范围中是否包含空间列。

API 密钥：`oracle-check-spatial-columns`

此迁移前评估可验证表的范围中是否包含空间列。DMS 仅支持使用完整 LOB 模式的空间数据类型。评估还会评估 DMS 识别空间列时任务是否配置为使用 Full LOB 模式。

验证 DMS 源是否已连接到 Oracle 备用服务器。

API 密钥：`oracle-check-standby-db`

此迁移前评估可验证源是否已连接到 Oracle 备用服务器。仅当使用以 Oracle 待机为源的二进制读取器时，DMS 才支持 CDC。评估还会评估任务是否配置为在 DMS 连接到 Oracle 待机时使用二进制读取器。

验证源数据库表空间是否使用 TDE 加密。

API 密钥：`oracle-check-tde-enabled`

此迁移前评估将验证源是否在表空间上启用了 TDE 加密。使用 Oracle for RDS Oracle 时，DMS 仅支持加密表空间 LogMiner 的 TDE。

验证源数据库是否为 Oracle ASM

API 密钥：`oracle-check-asm`

此迁移前评估可验证源是否使用 ASM。为了提高 ASM 配置的性能，请考虑在源端点设置中添加 `parallelASMRReadThreads` 和 `readAheadBlocks`。

## Sql 服务器评估

本节介绍使用 Microsoft SQL Server 源端点的迁移任务的单独迁移前评估。

### 主题

- [检查数据库的恢复模式是否简单](#)
- [检查任务范围内的表是否包含计算的列](#)
- [检查任务范围内的表是否包含列存储索引](#)
- [检查内存优化表是否属于任务范围](#)
- [检查时态表是否属于任务范围](#)
- [检查是否在数据库级别启用了延迟耐久性](#)
- [检查是否在数据库级别启用了加速数据恢复](#)
- [检查表映射中是否有超过 1 万个带主键的表](#)
- [检查源数据库中是否有带有特殊字符的表或架构名称。](#)
- [检查源数据库是否有包含掩码数据的列名](#)
- [检查源数据库是否有加密的备份](#)
- [检查源数据库是否有存储在 URL 或 Windows Azure 上的备份。](#)
- [检查源数据库是否在多个磁盘上有备份](#)
- [检查源数据库是否至少有一个完整备份](#)
- [检查源数据库是否具有稀疏列和列结构压缩。](#)
- [检查源数据库实例是否有 SQL Server 2008 或 SQL Server 2008 R2 的服务器级别审计](#)
- [检查源数据库是否有适用于完整 LOB 模式的几何列](#)
- [检查源数据库是否有带有身份属性的列。](#)
- [检查 DMS 用户是否具有“满载”权限](#)
- [检查 DMS 用户是否拥有满载权限以及仅限 CDC 或 CDC 的权限](#)
- [检查将 MS-CDC 与本地或 `ignoreMsReplicationEnablement EC2` 数据库配合使用时是否设置了 ECA](#)
- [检查 DMS 用户是否具有“查看定义”权限。](#)
- [对于没有 Sysadmin 角色的用户，请检查 DMS 用户是否拥有主数据库的“查看数据库状态”权限。](#)
- [检查 DMS 用户是否具有“查看服务器状态”权限。](#)

## 检查数据库的恢复模式是否简单

API 密钥：`sqlserver-check-for-recovery-model`

此迁移前评估验证了源端点恢复模型。AWS DMS 要求将恢复模式设置为Bulk logged或Full用于持续复制。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅 [在 SQL Server 源中使用持续复制 \( CDC \) 的先决条件](#)。

## 检查任务范围内的表是否包含计算的列

API 密钥：`sqlserver-check-for-computed-fields`

此迁移前评估检查是否存在计算列。AWS DMS 不支持从 SQL Server 计算列中复制更改。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

## 检查任务范围内的表是否包含列存储索引

API 密钥：`sqlserver-check-for-columnstore-indexes`

此迁移前评估检查是否存在带有列存储索引的表。AWS DMS 不支持从带有列存储索引的 SQL Server 表中复制更改。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

## 检查内存优化表是否属于任务范围

API 密钥：`sqlserver-check-for-memory-optimized-tables`

此迁移前评估会检查是否存在经过内存优化的表。AWS DMS 不支持从内存优化的表中复制更改。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

## 检查时态表是否属于任务范围

API 密钥：`sqlserver-check-for-temporal-tables`



此迁移前评估检查是否存在时态表。AWS DMS 不支持复制时态表中的更改。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查是否在数据库级别启用了延迟耐久性

API 密钥：`sqlserver-check-for-delayed-durability`

此迁移前评估检查是否存在延迟的耐久性。AWS DMS 不支持从使用延迟耐久性的交易中复制更改。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查是否在数据库级别启用了加速数据恢复

API 密钥：`sqlserver-check-for-accelerated-data-recovery`

此迁移前评估会检查是否存在加速的数据恢复。AWS DMS 不支持通过加速数据恢复功能从数据库中复制更改。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查表映射中是否有超过 1 万个带主键的表

API 密钥：`sqlserver-large-number-of-tables`

此迁移前评估检查是否存在超过 1 万个带主键的表。如果带有主键的表过多，则使用 MS-Replication 配置的数据库可能会遇到任务失败。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关配置 MS-Replication 的更多信息，请参阅[捕获本地或 Amazon EC2 上的自管理 SQL Server 的数据更改](#)。

检查源数据库中是否有带有特殊字符的表或架构名称。

API 密钥：`sqlserver-check-for-special-characters`

此迁移前评估可验证源数据库的表名或架构名是否包含以下集合中的字符：

```
\\ -- \n \" \b \r ' \t ;
```

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否有包含掩码数据的列名

API 密钥：sqlserver-check-for-masked-data

此迁移前评估可验证源数据库是否包含屏蔽的数据。AWS DMS 在不进行屏蔽的情况下迁移被屏蔽的数据。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否有加密的备份

API 密钥：sqlserver-check-for-encrypted-backups

此迁移前评估可验证源数据库是否有加密的备份。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否有存储在 URL 或 Windows Azure 上的备份。

API 密钥：sqlserver-check-for-backup-url

此迁移前评估将验证源数据库的备份存储在网址上还是存储在 Windows Azure 上。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否在多个磁盘上有备份

API 密钥：sqlserver-check-for-backup-multiple-stripes

此迁移前评估可验证源数据库是否在多个磁盘上有备份。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否至少有一个完整备份

API 密钥：sqlserver-check-for-full-backup

此迁移前评估可验证源数据库是否至少有一个完整备份。必须将 SQL Server 配置为完全备份，并且在复制数据之前必须运行备份。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否具有稀疏列和列结构压缩。

API 密钥：`sqlserver-check-for-sparse-columns`

此迁移前评估可验证源数据库是否具有稀疏列和列结构压缩。DMS 不支持稀疏列和列结构压缩。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库实例是否有 SQL Server 2008 或 SQL Server 2008 R2 的服务器级别审计

API 密钥：`sqlserver-check-for-audit-2008`

此迁移前评估验证源数据库是否已为 SQL Server 2008 或 SQL Server 2008 R2 启用服务器级审计。DMS 在 SQL Server 2008 和 2008 R2 中存在一个相关的已知问题。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否有适用于完整 LOB 模式的几何列

API 密钥：`sqlserver-check-for-geometry-columns`

此迁移前评估用于验证在使用 SQL Server 作为源数据库时，源数据库是否具有适用于完整大型对象 (LOB) 模式的几何列。当数据库包含几何列时，我们建议使用受限 LOB 模式或将 `InlineLobMaxSize` 任务设置设置为使用内联 LOB 模式。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查源数据库是否有带有身份属性的列。

API 密钥：`sqlserver-check-for-identity-columns`

此迁移前评估验证源数据库中是否有带有该 IDENTITY 属性的列。DMS 不会将此属性迁移到相应的目标数据库列。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查 DMS 用户是否具有“满载”权限

API 密钥：`sqlserver-check-user-permission-for-full-load-only`

此迁移前评估将验证 DMS 任务的用户是否有权在满负荷模式下运行任务。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查 DMS 用户是否拥有满载权限以及仅限 CDC 或 CDC 的权限

API 密钥：`sqlserver-check-user-permission-for-cdc`

此迁移前评估将验证 DMS 用户是否有权在或模式下运行任务。FULL LOAD and CDC CDC only

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查将 MS-CDC 与本地或 **ignoreMsReplicationEnablement** EC2 数据库配合使用时是否设置了 ECA

API 密钥：`sqlserver-check-attribute-for-enable-ms-cdc-onprem`

检查将 MS-CDC 与本地或 EC2 数据库配合使用时是否设置了 **ignoreMsReplicationEnablement** 额外连接属性 (ECA)。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查 DMS 用户是否具有“查看定义”权限。

API 密钥：`sqlserver-check-user-permission-on-view-definition`

此迁移前评估可验证终端节点设置中指定的用户是否拥有该VIEW DEFINITION权限。DMS 需要查看对象定义的VIEW DEFINITION权限。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

对于没有 Sysadmin 角色的用户，请检查 DMS 用户是否拥有主数据库的“查看数据库状态”权限。

API 密钥：`sqlserver-check-user-permission-on-view-database-state`

此迁移前评估可验证终端节点设置中指定的用户是否拥有该VIEW DATABASE STATE权限。DMS 需要此权限才能访问 MASTER 数据库中的数据库对象。当用户没有系统管理员权限时，DMS 也需要此权限。DMS 需要此权限才能创建函数、证书和登录名以及授予证书。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

检查 DMS 用户是否具有“查看服务器状态”权限。

API 密钥：`sqlserver-check-user-permission-on-view-server-state`

此迁移前评估将检查在额外连接属性 (ECA) 中指定的用户是否具有VIEW SERVER STATE权限。VIEW SERVER STATE是一种服务器级权限，允许用户查看服务器范围的信息和状态。此权限提供对动态管理视图 (DMV) 和动态管理功能 (DMF) 的访问权限，这些视图和动态管理功能 (DMF) 可公

开有关 SQL Server 实例的信息。DMS 用户需要此权限才能访问 CDC 资源。在 FULL LOAD and CDC或CDC only模式下运行 DMS 任务需要此权限。

有关更多信息，请参阅 [使用 SQL Server 作为源代码的限制 AWS DMS](#)。

## MySQL 评估

本节介绍使用 MySQL 源端点的迁移任务的单独迁移前评估。

### 主题

- [验证表是否使用 InnoDB 以外的存储引擎](#)
- [验证是否对任何用于迁移的表启用了自动增量](#)
- [验证数据库二进制日志图像是否设置 FULL 为支持 DMS CDC](#)
- [验证源数据库是否为 MySQL 只读副本](#)
- [验证表是否有分区，并建议 target\\_table\\_prep\\_mode 用于满载任务设置](#)
- [验证 DMS 是否支持数据库版本](#)
- [验证目标数据库是否配置为设置 local\\_infile 为 1](#)
- [验证目标数据库是否有带外键的表](#)
- [验证任务范围内的源表是否具有级联约束](#)
- [验证超时值是否适用于 MySQL 源或目标](#)

验证表是否使用 InnoDB 以外的存储引擎

API 密钥：`mysql-check-table-storage-engine`

此迁移前评估可验证用于源 MySQL 数据库中任何表的存储引擎是否是 InnoDB 以外的引擎。默认情况下，DMS 使用 InnoDB 存储引擎创建目标表。如果您需要使用 InnoDB 以外的存储引擎，则必须在目标数据库上手动创建表，并将您的 DMS 任务配置为使用 TRUNCATE\_BEFORE\_LOAD 或 DO\_NOTHING 作为满载任务设置。有关满载任务设置的更多信息，请参阅 [完全加载任务设置](#)。

有关 MySQL 终端节点限制的更多信息，请参阅 [使用 MySQL 数据库作为源代码的限制 AWS DMS](#)。

验证是否对任何用于迁移的表启用了自动增量

API 密钥：`mysql-check-auto-increment`

此迁移前评估可验证任务中使用的源表是否启用了自动增量。DMS 不会将列上的 AUTO\_INCREMENT 属性迁移到目标数据库。

有关 MySQL 终端节点限制的更多信息，请参阅[使用 MySQL 数据库作为源代码的限制 AWS DMS](#)。有关在 MySQL 中处理标识列的信息，请参见[AWS DMS：第 2 部分中的处理身份列](#)。

验证数据库二进制日志图像是否设置**FULL**为支持 DMS CDC

API 密钥：`mysql-check-binlog-image`

此迁移前评估检查源数据库的二进制日志图像是否设置为。FULL在 MySQL 中，该`binlog_row_image`变量决定了使用该ROW格式时如何写入二进制日志事件。为确保与 DMS 兼容并支持 CDC，请FULL将`binlog_row_image`变量设置为。此设置可确保 DMS 收到足够的信息，以便在迁移期间为目标数据库构建完整的数据操作语言 (DML)。

要将二进制日志图像设置为FULL，请执行以下操作：

- 对于 Amazon RDS，此值是FULL默认值。
- 对于托管在本地或亚马逊 EC2 上的数据库，请在 `my.ini` (Microsoft Windows) 或 `my.cnf` (UNIX) 中设置`binlog_row_image`值。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

验证源数据库是否为 MySQL 只读副本

API 密钥：`mysql-check-database-role`

此迁移前评估可验证源数据库是否为只读副本。要在连接到只读副本时启用 DMS 的 CDC 支持，请将`log_slave_updates`参数设置为。True有关使用自行管理的 MySQL 数据库的更多信息，请参阅[使用自管理的 MySQL 兼容数据库作为源 AWS DMS](#)。

要将该`log_slave_updates`值设置为True，请执行以下操作：

- 对于 Amazon RDS，请使用数据库的参数组。有关使用 RDS 数据库参数组的信息，请参阅 Amazon RDS 用户指南中的使用[参数组](#)。
- 对于托管在本地或亚马逊 EC2 上的数据库，请在 `my.ini` (Microsoft Windows) 或 `my.cnf` (UNIX) 中设置`log_slave_updates`值。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

验证表是否有分区，并建议**`target_table_prep_mode`**用于满载任务设置

API 密钥：`mysql-check-table-partition`

此迁移前评估检查源数据库中是否存在带有分区的表。DMS 在 MySQL 目标上创建没有分区的表。要将分区表迁移到目标系统上的分区表，必须执行以下操作：

- 在目标 MySQL 数据库中预先创建分区表。
- 将您的 DMS 任务配置为使用 TRUNCATE\_BEFORE\_LOAD 或 DO\_NOTHING 作为满载任务设置。

有关 MySQL 终端节点限制的更多信息，请参阅[使用 MySQL 数据库作为源代码的限制 AWS DMS](#)。

验证 DMS 是否支持数据库版本

API 密钥：`mysql-check-supported-version`

此迁移前评估验证源数据库版本是否与 DMS 兼容。Amazon RDS MySQL 5.5 或更低版本或 8.0.x 以上的 MySQL 版本不支持 CDC。只有 MySQL 版本 5.6、5.7 或 8.0 支持 CDC。有关支持的 MySQL 版本的更多信息，请参阅[数据迁移的源端点](#)。

验证目标数据库是否配置为设置 `local_infile` 为 1

API 密钥：`mysql-check-target-localinfile-set`

此迁移前评估检查目标数据库中的 `local_infile` 参数是否设置为 1。在目标数据库满负荷期间，DMS 要求将 “`local_infile`” 参数设置为 1。有关更多信息，请参阅[使用 AWS DMS 从 MySQL 迁移到 MySQL](#)。

此评估仅对满负荷或满载以及 CDC 任务有效。

验证目标数据库是否有带外键的表

API 密钥：`mysql-check-fk-target`

此迁移前评估检查迁移到 MySQL 数据库的 CDC 任务是满负荷还是满负荷以及迁移到 MySQL 数据库的表中是否有带外键的表。DMS 中的默认设置是按字母顺序加载表。具有外键和参照完整性约束的表可能会导致加载失败，因为父表和子表可能无法同时加载。

有关 DMS 中引用完整性的更多信息，请参阅主题中的使用索引、触发器和参照完整性约束。[改进 AWS DMS 迁移的性能](#)

验证任务范围内的源表是否具有级联约束

API 密钥：`mysql-check-cascade-constraints`

此迁移前评估检查是否有任何 MySQL 源表具有级联限制。DMS 任务不会迁移或复制级联约束，因为 MySQL 不会在二进制日志中记录这些事件的更改。虽然 AWS DMS 不支持这些限制，但你可以对关系数据库目标使用变通办法。

有关支持 cascade 限制和其他限制的信息，请参阅主题[索引、外键或级联更新内容或删除内容未迁移](#)中的迁移任务疑难解答。AWS DMS

验证超时值是否适用于 MySQL 源或目标

API 密钥：`mysql-check-network-parameter`

此迁移前评估检查任务的 MySQL 终端节点是否

将`net_read_timeout`、`net_wait_timeout`和`wait_timeout`设置设置为至少 300 秒。这是防止迁移期间断开连接所必需的。

有关更多信息，请参阅 [与目标 MySQL 实例的连接在任务期间中断](#)。

## MariaDB 评估

本节介绍使用 MariaDB 源端点的迁移任务的单独迁移前评估。

要使用 AWS DMS API 创建个人预迁移评估，请使用列出的 API 密钥作为 `StartReplicationTaskAssessmentRun` 操作 `Include` 参数。

### 主题

- [验证表是否使用 InnoDB 以外的存储引擎](#)
- [验证是否对任何用于迁移的表启用了自动增量](#)
- [验证数据库二进制日志格式是否设置 ROW 为支持 DMS CDC](#)
- [验证数据库二进制日志图像是否设置 FULL 为支持 DMS CDC](#)
- [验证源数据库是否为 MariaDB 只读副本](#)
- [验证表是否有分区，并建议 TRUNCATE\\_BEFORE\\_LOAD 或 DO\\_NOTHING 用于满载任务设置](#)
- [验证 DMS 是否支持数据库版本](#)
- [验证目标数据库是否配置为设置 local\\_infile 为 1](#)
- [验证目标数据库是否有带外键的表](#)
- [验证任务范围内的源表是否具有级联约束](#)
- [验证任务范围内的源表是否已生成列](#)
- [验证超时值是否适合于 MariaDB 源](#)
- [验证超时值是否适合于 MariaDB 目标](#)



## 验证表是否使用 InnoDB 以外的存储引擎

API 密钥 : mariadb-check-table-storage-engine

此迁移前评估可验证用于源 MariaDB 数据库中任何表的存储引擎是否是 InnoDB 以外的引擎。默认情况下，DMS 使用 InnoDB 存储引擎创建目标表。如果您需要使用 InnoDB 以外的存储引擎，则必须在目标数据库上手动创建表，并将您的 DMS 任务配置为使用 TRUNCATE\_BEFORE\_LOAD 或 DO\_NOTHING 作为满载任务设置。有关满载任务设置的更多信息，请参阅 [完全加载任务设置](#)。

有关 MariaDB 端点限制的更多信息，请参阅 [使用 MySQL 数据库作为源代码的限制 AWS DMS](#)

## 验证是否对任何用于迁移的表启用了自动增量

API 密钥 : mariadb-check-auto-increment

此迁移前评估可验证任务中使用的源表是否启用了自动增量。DMS 不会将列上的 AUTO\_INCREMENT 属性迁移到目标数据库。

有关 MariaDB 端点限制的更多信息，请参阅 [使用 MySQL 数据库作为源代码的限制 AWS DMS](#) 有关在 MariaDB 中处理标识列的信息，请参见 [AWS DMS：第 2 部分中的处理身份列](#)。

## 验证数据库二进制日志格式是否设置 ROW 为支持 DMS CDC

API 密钥 : mariadb-check-binlog-format

此迁移前评估将验证源数据库二进制日志格式是否设置为 ROW 支持 DMS 更改数据捕获 (CDC)。

要将二进制日志格式设置为 ROW，请执行以下操作：

- 对于 Amazon RDS，请使用数据库的参数组。有关使用 RDS 参数组的信息，请参阅 Amazon RDS 用户指南中的 [配置 MySQL 二进制日志记录](#)。
- 对于托管在本地或亚马逊 EC2 上的数据库，请在 my.ini (Microsoft Windows) 或 my.cnf (UNIX) 中设置 binlog\_format 值。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关自托管 MariaDB 服务器的更多信息，请参阅 [使用自管理的 MySQL 兼容数据库作为源 AWS DMS](#)

## 验证数据库二进制日志图像是否设置 FULL 为支持 DMS CDC

API 密钥 : mariadb-check-binlog-image

此迁移前评估检查源数据库的二进制日志图像是否设置为 FULL。在 MariaDB 中，该 `binlog_row_image` 变量决定了使用该格式时如何写入二进制日志事件。ROW 为确保与 DMS 兼容并支持 CDC，请 FULL 将 `binlog_row_image` 变量设置为 FULL。此设置可确保 DMS 收到足够的信息，以便在迁移期间为目标数据库构建完整的数据操作语言 (DML)。

要将二进制日志图像设置为 FULL，请执行以下操作：

- 对于 Amazon RDS，此值是 FULL 默认值。
- 对于托管在本地或亚马逊 EC2 上的数据库，请在 `my.ini` (Microsoft Windows) 或 `my.cnf` (UNIX) 中设置 `binlog_row_image` 值。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

有关自托管 MariaDB 服务器的更多信息，请参阅 [使用自管理的 MySQL 兼容数据库作为源 AWS DMS](#)

验证源数据库是否为 MariaDB 只读副本

API 密钥：`mariadb-check-database-role`

此迁移前评估可验证源数据库是否为只读副本。要在连接到只读副本时启用 DMS 的 CDC 支持，请将 `log_slave_updates` 参数设置为 `True`。有关使用自行管理的 MySQL 数据库的更多信息，请参阅 [使用自管理的 MySQL 兼容数据库作为源 AWS DMS](#)。

要将该 `log_slave_updates` 值设置为 `True`，请执行以下操作：

- 对于 Amazon RDS，请使用数据库的参数组。有关使用 RDS 数据库参数组的信息，请参阅 Amazon RDS 用户指南中的使用 [参数组](#)。
- 对于托管在本地或亚马逊 EC2 上的数据库，请在 `my.ini` (Microsoft Windows) 或 `my.cnf` (UNIX) 中设置 `log_slave_updates` 值。

此评估仅对完全加载迁移和 CDC 迁移或仅 CDC 迁移有效。此评估对仅完全加载迁移无效。

验证表是否有分区，并建议 `TRUNCATE_BEFORE_LOAD` 或 `DO_NOTHING` 用于满载任务设置

API 密钥：`mariadb-check-table-partition`

此迁移前评估检查源数据库中是否存在带有分区的表。DMS 在 MariaDB 目标上创建没有分区的表。要将分区表迁移到目标系统上的分区表，必须执行以下操作：

- 在目标 MariaDB 数据库中预先创建分区表。
- 将您的 DMS 任务配置为使用 TRUNCATE\_BEFORE\_LOAD 或 DO\_NOTHING 作为满载任务设置。

有关 MariaDB 端点限制的更多信息，请参阅 [使用 MySQL 数据库作为源代码的限制 AWS DMS](#)

验证 DMS 是否支持数据库版本

API 密钥：mariadb-check-supported-version

此迁移前评估验证源数据库版本是否与 DMS 兼容。亚马逊 RDS MariaDB 10.4 或更低版本或 10.11 以上的 MySQL 版本不支持 CDC。有关支持的 MariaDB 版本的更多信息，请参阅 [数据迁移的源端点](#)

验证目标数据库是否配置为设置 `local_infile` 为 1

API 密钥：mariadb-check-target-localinfile-set

此迁移前评估检查目标数据库中的 `local_infile` 参数是否设置为 1。在目标数据库满负荷期间，DMS 要求将“local\_infile”参数设置为 1。有关更多信息，请参阅 [使用 AWS DMS 从 MySQL 迁移到 MySQL。](#)

此评估仅对满负荷任务有效。

验证目标数据库是否有带外键的表

API 密钥：mariadb-check-fk-target

此迁移前评估检查迁移到 MariaDB 数据库的 CDC 任务是否包含带有外键的表。DMS 中的默认设置是按字母顺序加载表。具有外键和参照完整性约束的表可能会导致加载失败，因为父表和子表可能无法同时加载。

有关 DMS 中引用完整性的更多信息，请参阅主题中的使用索引、触发器和参照完整性约束。 [改进 AWS DMS 迁移的性能](#)

验证任务范围内的源表是否具有级联约束

API 密钥：mariadb-check-cascade-constraints

此迁移前评估检查是否有任何 MariaDB 源表具有级联约束。DMS 任务不会迁移或复制级联约束，因为 MariaDB 不会在二进制日志中记录这些事件的更改。虽然 AWS DMS 不支持这些限制，但你可以对关系数据库目标使用变通办法。

有关支持 `casca` 限制和其他限制的信息，请参阅主题[索引、外键或级联更新内容或删除内容未迁移](#)中的迁移任务疑难解答。AWS DMS

验证任务范围内的源表是否已生成列

API 密钥：`mariadb-check-generated-columns`

此迁移前评估检查是否有任何MariaDB源表已生成列。DMS 任务不会迁移或复制生成的列。

有关如何迁移生成的列的信息，请参见[???](#)。

验证超时值是否适合于 MariaDB 源

API 密钥：`mariadb-check-source-network-parameter`

此迁移前评估检查任务的 MariaDB 源端点是

否`net_read_timeout`将`net_wait_timeoutwait_timeout`和设置设置为至少 300 秒。这是防止迁移期间断开连接所必需的。

有关更多信息，请参阅[与目标 MySQL 实例的连接在任务期间中断](#)。

验证超时值是否适合于 MariaDB 目标

API 密钥：`mariadb-check-target-network-parameter`

此迁移前评估检查任务的 MariaDB 目标端点是

否`net_read_timeout`将`net_wait_timeoutwait_timeout`和设置设置为至少 300 秒。这是防止迁移期间断开连接所必需的。

有关更多信息，请参阅[与目标 MySQL 实例的连接在任务期间中断](#)。

## PostgreSQL 评估

本节介绍使用 PostgreSQL 源端点的迁移任务的单独迁移前评估。

主题

- [验证 DMS 是否支持源数据库版本进行迁移](#)
- [校验源数据库上的`logical\_decoding\_work\_mem`参数](#)
- [验证源数据库是否有长时间运行的事务](#)
- [验证源数据库参数`max\_slot\_wal\_keep\_size`](#)
- [检查源数据库参数`postgres-check-maxwalsenders`是否设置为支持 CDC。](#)

- [检查源数据库是否配置为 PGLOGICAL](#)
- [验证源表主键是否为 LOB 数据类型](#)
- [验证源表是否有主键](#)
- [验证源数据库中是否存在准备好的事务](#)
- [验证wal\\_sender\\_timeout是否设置为支持 DMS CDC 所需的最小值](#)
- [验证源数据库上wal\\_level是否设置为逻辑](#)

验证 DMS 是否支持源数据库版本进行迁移

API 密钥：postgres-check-dbversion

此迁移前评估可验证源数据库版本是否与兼容。AWS DMS

校验源数据库上的**logical\_decoding\_work\_mem**参数

API 密钥：postgres-check-for-logical-decoding-work-mem

此迁移前评估建议调整源数据库上的**logical\_decoding\_work\_mem**参数。在事务性很强的数据库中，您可能有长时间运行的事务或许多子事务，这可能会导致逻辑解码内存消耗增加，并且需要溢出到磁盘。这会导致复制期间的 DMS 源延迟过高。在这种情况下，您可能需要进行调整**logical\_decoding\_work\_mem**。PostgreSQL 版本 13 及更高版本支持此参数。

验证源数据库是否有长时间运行的事务

API 密钥：postgres-check-longrunningtxn

此迁移前评估可验证源数据库是否存在持续时间超过 10 分钟的长时间运行的事务。启动任务可能会失败，因为默认情况下，DMS 会在启动任务时检查是否存在任何未完成的交易。

验证源数据库参数 **max\_slot\_wal\_keep\_size**

API 密钥：postgres-check-maxslot-wal-keep-size

此迁移前评估验证了为配置的值。**max\_slot\_wal\_keep\_size**当设置**max\_slot\_wal\_keep\_size**为非默认值时，DMS 任务可能会因为删除所需的 WAL 文件而失败。

检查源数据库参数**postgres-check-maxwalsenders**是否设置为支持 CDC。

API 密钥：postgres-check-maxwalsenders

此迁移前评估会验证源数据库max\_wal\_senders上为配置的值。DMS 需要设置max\_wal\_senders为大于 1 才能支持变更数据捕获 (CDC)。

检查源数据库是否配置为 **PGLLOGICAL**

API 密钥 : postgres-check-pglogical

此迁移前评估可验证该shared\_preload\_libraries值是否设置pglogicalPGLLOGICAL为支持 CDC。请注意，如果您计划使用测试解码进行逻辑复制，则可以忽略此评估。

验证源表主键是否为 LOB 数据类型

API 密钥 : postgres-check-pk-lob

此迁移前评估验证表的主键是否为大型对象 (LOB) 数据类型。如果源表将 LOB 列作为主键，则 DMS 不支持复制。

验证源表是否有主键

API 密钥 : postgres-check-pk

此迁移前评估验证任务范围内使用的表是否存在主键。除非在源表上将副本标识设置为，否则 DMS 不支持对没有主键的表full进行复制。

验证源数据库中是否存在准备好的事务

API 密钥 : postgres-check-preparedtxn

此迁移前评估可验证源数据库中是否存在任何准备好的事务。如果源数据库上存在任何准备好的事务，则创建复制槽可能会停止响应。

验证wal\_sender\_timeout是否设置为支持 DMS CDC 所需的最小值

API 密钥 : postgres-check-walsendersttimeout

此迁移前评估将验证wal\_sender\_timeout是否设置为至少 10000 毫秒 ( 10 秒 )。使用 CDC 执行的 DMS 任务至少需要 10000 毫秒 ( 10 秒 )，如果该值小于 10000，则会失败。

验证源数据库上wal\_level是否设置为逻辑

API 密钥 : postgres-check-wallevel

此迁移前评估将验证是否设置wal\_level为逻辑。要让 DMS CDC 起作用，需要在源数据库上启用此参数。

## 启动和查看数据类型评估 (旧版)

### Note

本节介绍旧版内容。我们建议您使用迁移前评估运行，如前所[指定、启动和查看迁移前评估运行](#)行述。

控制台中不提供数据类型评估。您只能使用 API 或 CLI 运行数据类型评估，并且只能在任务的 S3 存储桶中查看数据类型评估的结果。

数据类型评估可识别源数据库中由于目标数据库不支持而可能无法正确迁移的数据类型。在此评估期间，AWS DMS 读取迁移任务的源数据库架构并创建列数据类型列表。然后，它将此列表与支持的预定义数据类型列表进行比较 AWS DMS。如果您的迁移任务包含不支持的数据类型，请 AWS DMS 创建一份报告，您可以查看该报告以查看迁移任务中是否有任何不受支持的数据类型。AWS DMS 如果您的迁移任务没有任何不受支持的数据类型，则不会创建报告。

AWS DMS 支持为以下关系数据库创建数据类型评估报告：

- Oracle
- SQL Server
- PostgreSQL
- MySQL
- MariaDB
- Amazon Aurora

您可以使用 CLI 和 SDK 启动和查看数据类型评估报告以访问 AWS DMS API：

- CLI 使用 [start-replication-task-assessment](#) 命令启动数据类型评估，并使用 [describe-replication-task-assessment-results](#) 命令查看 JSON 格式的最新数据类型评估报告。
- AWS DMS API 使用该 [StartReplicationTaskAssessment](#) 操作启动数据类型评估，并使用该 [DescribeReplicationTaskAssessmentResults](#) 操作查看 JSON 格式的最新数据类型评估报告。

数据类型评估报告是一个 JSON 文件，其中列出不支持的数据类型以及每种数据类型的列数。它包括每种不支持的数据类型的数据结构列表，包括包含不支持数据类型的架构、表和列。您可以使用该报告修改源数据类型并提高迁移成功率。

共有两种级别的不支持的数据类型。报告中显示为“not supported (不支持)”的数据类型无法迁移。报告中显示为部分支持的数据类型可以转换为其他数据类型，但不会按预期方式进行迁移。

以下示例显示您可以查看的数据类型评估报告示例。

```
{
  "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 将最新和以前的所有数据类型评估存储在您的账户中创建的 Amazon S3 存储桶 AWS DMS 中。Amazon S3 存储桶名称采用以下格式，其中 *customerId* 是您的客户 ID，*customerDNS* 是内部标识符。

```
dms-customerId-customerDNS
```

#### Note

默认情况下，您可以在每个 AWS 账户中创建多达 100 个 Amazon S3 存储桶。由于在您的账户中 AWS DMS 创建了存储桶，因此请确保其不超过您的存储桶限制。否则，数据类型评估将失败。

给定迁移任务的所有数据类型评估报告都存储在以任务标识符命名的存储桶文件夹中。每个报告的文件名都是数据类型评估的日期，格式为 yyyy-mm-dd-hh-mm。您可以在 Amazon S3 管理控制台中查看和比较以前的任务评估报告。

AWS DMS 还会创建一个 AWS Identity and Access Management (IAM) 角色以允许访问为这些报告创建的 S3 存储桶。角色名称为 `dms-access-for-tasks`。该角色使用 `AmazonDMSRedshiftS3Role` 策略。如果运行时出现 `ResourceNotFoundFault` 错误 `StartReplicationTaskAssessment`，请参阅 [ResourceNotFoundFault](#) “疑难解答” 部分，了解有关手动创建 `dms-access-for-tasks` 角色的信息。

## 对评估运行进行故障排除

接下来，您可以找到有关使用运行评估报告时遇到的问题进行故障排除的主题 AWS Database Migration Service。这些主题可以帮助您解决常见问题。

### 主题

- [ResourceNotFoundFault 跑步时 StartReplicationTaskAssessment](#)

## ResourceNotFoundFault 跑步时 StartReplicationTaskAssessment

运行 [StartReplicationTaskAssessment](#) 操作时可能会遇到以下异常。

```
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
```

如果遇到此异常，请执行以下操作来创建 dms-access-for-tasks 角色：

1. 通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 在导航窗格中，选择 Roles（角色）。
3. 选择 创建角色。
4. 在选择可信实体中的可信实体类型下，选择自定义信任策略。
5. 将以下 JSON 粘贴到编辑器中，替换现有文本。

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

前面的策略向授 `sts:AssumeRole` 予权限 AWS DMS。当您添加 `AmazonDMSRedshifts3Role` 策略时，DMS 可以在您的账户中创建 S3 存储桶，并将数据类型评估结果放入此 S3 存储桶。

6. 选择下一步。
7. 在“添加权限”页面上，搜索并添加 `AmazonDMSRedshifts3Role` 策略。选择下一步。
8. 在“名称、查看和创建”页面上，为角色命名 `dms-access-for-tasks`。选择 创建角色。

## 指定任务设置的补充数据

为某些 AWS DMS 终端节点创建或修改复制任务时，该任务可能需要附加信息才能执行迁移。可以使用 DMS 控制台中的选项指定此附加信息。或者，可以使用 DMS API 操作 `CreateReplicationTask` 或 `ModifyReplicationTask` 的 `TaskData` 参数进行指定。

如果目标端点为 Amazon Neptune，则需要指定映射数据，作为表映射的补充。此补充映射数据指定如何将源关系数据转换为 Neptune 数据库可以使用的目标图形数据。在这种情况下，可以使用两种可能的格式之一。有关更多信息，请参阅[将 Amazon Neptune 作为目标使用时使用 Gremlin 和 R2RML 指定图形映射规则](#)。

## 监控 AWS DMS 任务

监控是保持 AWS DMS 和您的 AWS 解决方案的可靠性、可用性和性能的重要方面。您应该从 AWS 解决方案的各个部分收集监控数据，以便您可以更轻松地调试多点故障（如果发生）。AWS 提供了多种工具来监控您的 AWS DMS 任务和资源并对潜在事件做出响应。

### AWS DMS 事件和通知

AWS DMS 使用 Amazon Simple Notification Service (Amazon SNS) 在发生 AWS DMS 事件时提供通知，例如，创建或删除复制实例。AWS DMS 将事件分组为各个类别，您可以订阅这些类别以便在发生该类别中的事件时收到通知。例如，如果您订阅了指定复制实例的“创建”类别，则只要发生了与创建相关的事件影响到您的复制实例，您就会收到通知。您可以通过 AWS 区域的 Amazon SNS 支持的任何形式使用这些通知，如电子邮件、短信或者调用 HTTP 端点。有关更多信息，请参阅[在 AWS Database Migration Service 中使用 Amazon SNS 事件和通知](#)。

### 任务状态

您可以查看任务状态和监控任务的控制表，以便监控任务的进度。任务状态指示 AWS DMS 任务及其关联资源的状况。它包含此类指示，如同正在创建、启动、运行或停止任务一样。此外，它还包含任务正在迁移的表的当前状态，例如，表已开始或正在进行完全加载时，并包含表已执行的插入、删除和更新次数等详细信息。有关监控任务和任务资源条件的更多信息，请参阅[任务状态](#)和[任务期间的表状态](#)。有关控制表的更多信息，请参阅[控制表任务设置](#)。

### Amazon CloudWatch 警报和日志

使用 Amazon CloudWatch 警报，您可以监控指定时间段内的一个或多个任务指标。如果指标超过给定阈值，则会向 Amazon SNS 主题发送通知。CloudWatch 警报将不会调用操作，因为这些操作处于特定状态。而是必须在状态已改变并在指定的若干个时间段内保持不变后才调用。AWS DMS 还使用 CloudWatch 在迁移过程中记录任务信息。您可以使用 AWS CLI 或 AWS DMS API 查看有关任务日志的信息。有关将 CloudWatch 与 AWS DMS 搭配使用的更多信息，请参阅[使用 Amazon CloudWatch 监控复制任务](#)。有关监控 AWS DMS 指标的更多信息，请参阅[AWS Database Migration Service 指标](#)。有关使用 AWS DMS 任务日志的更多信息，请参阅[查看和管理 AWS DMS 任务日志](#)。

### Time Travel 日志

要记录和调试复制任务，可以使用 AWS DMS Time Travel。在此方法中，您可以使用 Amazon S3 来存储日志，并使用加密密钥对其进行加密。您可以使用日期时间筛选器检索 S3 日志，然后根据需要查看、下载日志以及对日志进行模糊处理。由此，您可以安全地“回到过去”来调查数据库活动。

您可以将 Time Travel 与 DMS 支持的 PostgreSQL 源端点以及 DMS 支持的 PostgreSQL 和 MySQL 目标端点配合使用。您只能为完全加载和 CDC 任务以及仅 CDC 任务启用 Time Travel。要打开 Time Travel 或修改任何现有的 Time Travel 设置，请确保您的任务已停止。

有关 Time Travel 日志的更多信息，请参阅[Time Travel 任务设置](#)。有关使用 Time Travel 日志的最佳实践，请参阅[使用 Time Travel 对复制任务进行故障排除](#)。

## AWS CloudTrail 日志

AWS DMS 与 AWS CloudTrail 集成，后者是一项服务，可用于记录 AWS DMS 中由用户、IAM 角色或 AWS 服务所执行的操作。CloudTrail 将对 AWS DMS 的所有 API 调用均作为事件捕获，包括来自 AWS DMS 控制台的调用和对 AWS DMS API 操作的代码调用。如果您创建跟踪记录，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 AWS DMS 的事件）。如果您不配置跟踪记录，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 AWS DMS 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。有关更多信息，请参阅[使用 AWS DMS 记录 AWS CloudTrail API 调用](#)。

## 数据库日志

您可以使用 AWS 数据库服务的 AWS Management Console、AWS CLI 或 API 查看、下载和监控任务终端节点的数据库日志。有关更多信息，请参阅[AWS 文档](#)中的数据库服务文档。

有关更多信息，请参阅以下主题。

### 主题

- [任务状态](#)
- [任务期间的表状态](#)
- [使用 Amazon CloudWatch 监控复制任务](#)
- [AWS Database Migration Service 指标](#)
- [查看和管理 AWS DMS 任务日志](#)
- [使用 AWS DMS 记录 AWS CloudTrail API 调用](#)
- [AWS DMS 上下文日志记录](#)

## 任务状态

任务状态指示任务的情况。下表显示了任务可能具有的状态：

任务状态	描述
Creating	AWS DMS 在创建任务。
正在运行	任务正在执行指定的迁移工作。
Stopped	任务已停止。
Stopping	正在停止任务。这通常表明任务中有用户干预。
Deleting	正在删除任务，通常源于用户的干预请求。
Failed	任务已失败。有关更多信息，请参阅任务日志文件。
Error	任务因错误而停止。在概述选项卡的最后一条失败消息部分中，提供了任务错误的简要描述。
运行时出错	任务正在运行，状态为出错。这通常表示任务中的一个或多个表无法迁移。该任务继续根据选择规则加载其他表。
Starting	任务正在连接到复制实例以及源和目标终端节点。应用所有筛选器和转换。
Ready (就绪)	任务已准备好运行。此状态通常在“creating”状态之后。
Modifying	正在修改任务，通常是由于修改了任务设置的用户操作造成。
正在移动	任务正在移动到另一个复制实例。复制将保持此状态，直到移动完成。复制任务正在移动时，唯一允许执行的操作是删除任务。
失败的移动	任务移动由于任何原因而失败，例如目标复制实例上没有足够的存储空间。当复制任务处于这种状态时，可以启动、修改、移动或删除该任务。
测试	为此任务指定的数据库迁移正在进行测试，以响应 <a href="#">StartReplicationTaskAssessmentRun</a> 或 <a href="#">StartReplicationTaskAssessment</a> 操作的运行。

任务状态栏提供了任务进度的估计值。此估计值的质量取决于源数据库的表统计数据的质量；表统计数据的质量越好，估计值就越准确。对于只有一个表而且该表没有估算行统计数据的任务，我们无法提供任何类型的完成百分比估算值。在此情况下，可使用任务状态和加载行的指示来确认任务是否确实在运行且取得进展。

请注意，对于“上次更新时间”列，DMS 控制台仅指示 AWS DMS 上次更新表的表统计数据记录的时间。它不指示上次更新表的时间。

除了使用 DMS 控制台外，您还可以使用 [AWS CLI](#) 中的 `aws dms describe-replication-tasks` 命令，输出当前复制任务的描述，包括任务状态，如以下示例所示。

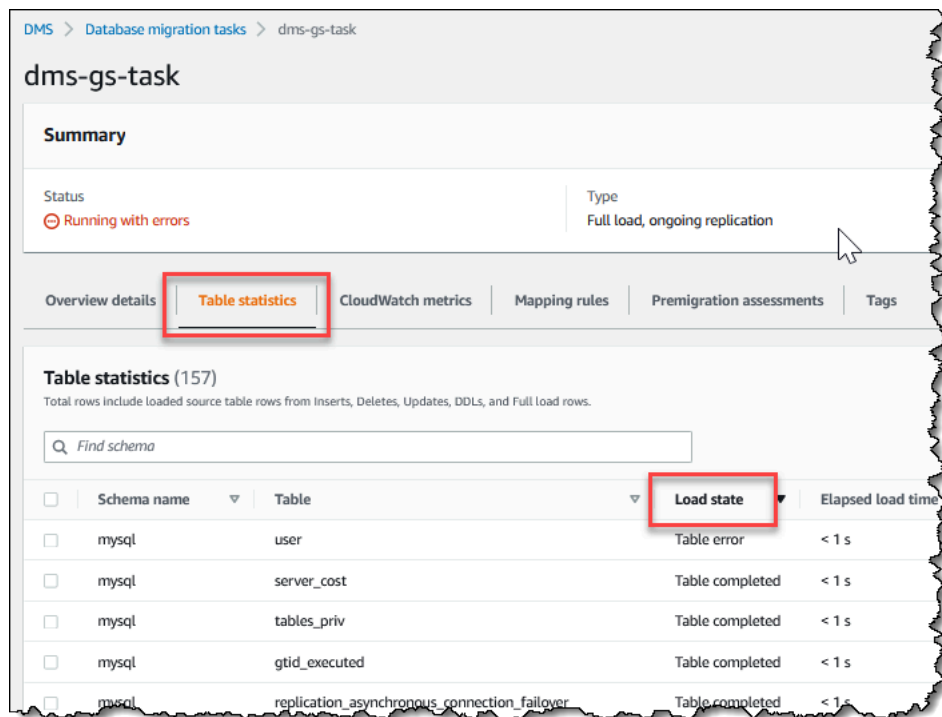
```
{
  "ReplicationTasks": [
    {
      "ReplicationTaskIdentifier": "moveit2",
      "SourceEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",
      "TargetEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
      "ReplicationInstanceArn": "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
      }
    }
  ]
}
```



}

## 任务期间的表状态

AWS DMS 控制台更新表在迁移期间相关状态的信息。下表显示了可能的状态值：



状态	描述
表不存在	AWS DMS 在源端点上找不到表。
加载前	完全加载过程已启用，但尚未启动。
完全加载	完全加载过程正在进行。
表已完成	完全加载已完成。
表已取消	已取消加载表。
表错误	加载表时出错。

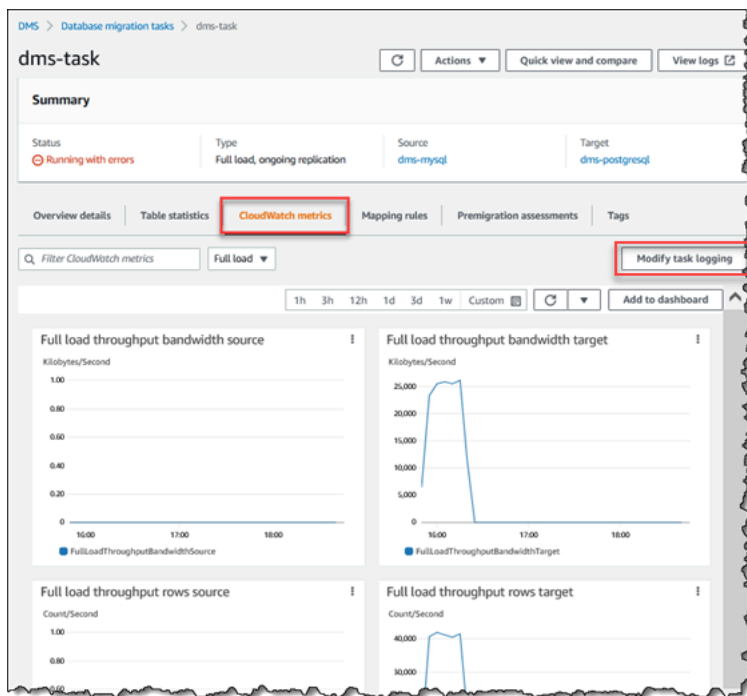
## 使用 Amazon CloudWatch 监控复制任务

您可以使用 Amazon CloudWatch 警报或事件，以便更密切地跟踪您的迁移。有关 Amazon CloudWatch 的更多信息，请参阅《Amazon CloudWatch 用户指南》中的[什么是 Amazon CloudWatch](#)、[Amazon CloudWatch Events](#) 和 [Amazon CloudWatch Logs](#)？。请注意，在使用 Amazon CloudWatch 时，需要支付相应的费用。

如果您的复制任务未创建 CloudWatch 日志，请参阅故障排除指南中的[AWS DMS不创建 CloudWatch 日志](#)。

AWS DMS 控制台显示了每个任务的基本 CloudWatch 统计信息，例如：任务状态、完成百分比、已用时间、表统计信息等，如下所示。选择复制任务，然后选择 CloudWatch 指标选项卡。

要查看和修改 CloudWatch 任务日志设置，请选择修改任务日志记录。有关更多信息，请参阅[日志记录任务设置](#)。



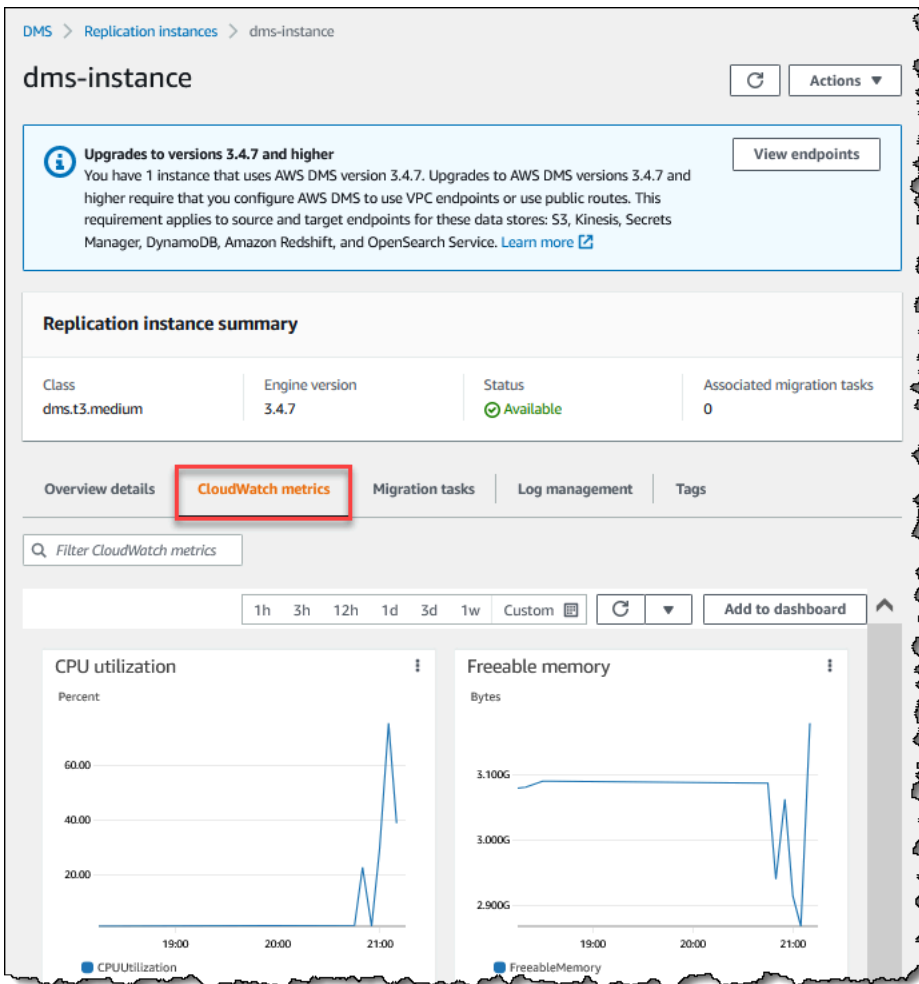
当您选择表统计数据选项卡时，AWS DMS 控制台显示每个表的性能统计数据，包括插入数、删除数和更新数。

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

此外，如果您从复制实例页面中选择一个复制实例，则可以通过选择 CloudWatch 指标选项卡来查看该实例的性能指标。



## AWS Database Migration Service 指标

AWS DMS 提供以下指标的统计数据：

- 主机指标 – 复制主机的性能和利用率统计数据，由 Amazon CloudWatch 提供。有关可用指标的完整列表，请参阅[复制实例指标](#)。
- 复制任务指标 – 复制任务的统计数据，包括传入和已提交更改，以及复制主机与源数据库和目标数据库之间的延迟。有关可用指标的完整列表，请参阅[复制任务指标](#)。
- 表指标 – 迁移过程中的表的统计数据，包括插入数、更新数、删除数和完成的 DDL 语句。

任务指标划分为复制主机与源终端节点之间的统计数据，以及复制主机与目标终端节点之间的统计数据。您可以将两个相关的统计数据加在一起，确定某个任务的统计数据总计。例如，您可以合并 CDCLatencySource 和 CDCLatencyTarget 值以确定总延迟时间或副本滞后。

任务指标值受您源数据库上当前活动的影响。例如，如果事务已开始但尚未提交，CDCLatencySource 指标将继续增长，直至提交该事务。

对于复制实例，需要澄清一下 FreeableMemory 指标。FreeableMemory 并不指示实际的可用内存。它是当前使用的可释放内存 (以用于其他用途)；它是复制实例上使用的缓冲区和缓存的组合。

虽然 FreeableMemory 指标不反映实际的可用内存，但 FreeableMemory 和 SwapUsage 指标组合可以指示复制实例是否过载。

请监控这两个指标是否出现以下情况：

- FreeableMemory 指标接近于零。
- SwapUsage 指标增加或发生波动。

如果出现上述任一情况，则表示应考虑移动到更大的复制实例。还应考虑减少在复制实例上运行的任务数量和类型。完全加载任务所需的内存比仅复制更改的任务需要的内存多。

要粗略估计 AWS DMS 迁移任务的实际内存需求，您可以使用以下参数。

## LOB 列

迁移范围内每个表中平均的 LOB 列数。

可并行加载的表数量上限

AWS DMS 在一个任务中并行加载的最大表数量。

默认值是 8。

## LOB 块大小

AWS DMS 将数据复制到目标数据库时使用的 LOB 块大小，以 KB 为单位。

## 完全加载时的提交率

AWS DMS 可并行传输的最大记录数。

默认值是 10000。

## LOB 大小

单个 LOB 的最大大小，以 KB 为单位。

## 批量数组大小

端点驱动程序提取或处理的最大行数。此值取决于驱动程序设置。

默认值是 1,000。

确定这些值后，您可以使用以下方法之一来估算迁移任务所需的内存量。这些方法取决于您在迁移任务中为 LOB 列设置选择的选项。

- 对于完整 LOB 模式，请使用以下公式。

$$\text{Required memory} = (\text{LOB columns}) * (\text{Maximum number of tables to load in parallel}) * (\text{LOB chunk size}) * (\text{Commit rate during full load})$$

举一个例子，您的源表平均包含 2 个 LOB 列，而 LOB 块的大小为 64 KB。如果您使用 `Maximum number of tables to load in parallel` 和 `Commit rate during full load` 的默认值，则任务所需的内存量如下所示。

$$\text{Required memory} = 2 * 8 * 64 * 10,000 = 10,240,000 \text{ KB}$$

### Note

要降低完全加载时的提交率的值，请打开 AWS DMS 控制台，选择数据库迁移任务，然后创建或修改任务。展开高级设置，然后输入完全加载时的提交率的值。

- 对于受限 LOB 模式，请使用以下公式。

$$\text{Required memory} = (\text{LOB columns}) * (\text{Maximum number of tables to load in parallel}) * (\text{LOB size}) * (\text{Bulk array size})$$

举一个例子，您的源表平均包含 2 个 LOB 列，而单个 LOB 的最大大小为 4096 KB。如果您使用 Maximum number of tables to load in parallel 和 Bulk array size 的默认值，则任务所需的内存量如下所示。

$$\text{Required memory} = 2 * 8 * 4,096 * 1,000 = 65,536,000 \text{ KB}$$

要让 AWS DMS 以最佳性能执行转换，进行转换时必须有 CPU 可用。CPU 过载和没有足够的 CPU 资源可能会导致迁移速度缓慢。AWS DMS 可能是 CPU 密集型任务，尤其是在执行异构迁移和复制（例如从 Oracle 迁移到 PostgreSQL）时。对于这些情况，C4 复制实例类是理想选择。有关更多信息，请参阅[为您的迁移选择正确的 AWS DMS 复制实例](#)。

## 复制实例指标

复制实例监控包括针对以下统计数据的 Amazon CloudWatch 指标。

指标	描述
AvailableMemory	<p>估计可用于启动新应用程序的内存量（没有交换）。有关更多信息，请参阅 <a href="#">Linux man-pages</a> 的 /proc/memInfo 一节中的 MemAvailable 值。</p> <p>单位：字节</p>
CPUAllocated	<p>为任务分配的 CPU 最大百分比（0 表示没有限制）。</p> <p>AWS DMS 根据 CloudWatch 控制台中 ReplicationTaskIdentifier 和 ReplicationInstanceIdentifier 维度的组合提出这个指标。使用 ReplicationInstanceIdentifier, ReplicationTaskIdentifier 类别查看此指标。</p> <p>单位：百分比</p>
CPU 利用率	<p>当前实例上正在使用的已分配 vCPU（虚拟 CPU）的百分比。</p> <p>单位：百分比</p>
DiskQueueDepth	<p>等待访问磁盘的未完成 I/O（读取/写入请求）的数量。</p> <p>单位：计数</p>

指标	描述
可用存储空间	可用存储空间的大小。  单位：字节
FreeMemory	可供应用程序、页面缓存和内核自身数据结构使用的物理内存量。有关更多信息，请参阅 <a href="#">Linux man-pages</a> 的 /proc/memInfo 一节中的 MemFree 值。  单位：字节
FreeableMemory	随机存取内存的可用大小。  单位：字节
MemoryAllocated	为任务分配的最大内存量（0 表示没有限制）。  AWS DMS 根据 CloudWatch 控制台中 ReplicationTaskIdentifier 和 ReplicationInstanceIdentifier 维度的组合提出这个指标。使用 ReplicationInstanceIdentifier, ReplicationTaskIdentifier 类别查看此指标。  单位：MiB
写入 IOPS	每秒平均磁盘写入 I/O 操作数。  单位：计数/秒
读取 IOPS	每秒平均磁盘读取 I/O 操作数。  单位：计数/秒
写入吞吐量	每秒写入磁盘的平均字节数。  单位：字节/秒
读取吞吐量	每秒从磁盘读取的平均字节数。  单位：字节/秒

指标	描述
WriteLatency	每个磁盘 I/O ( 输出 ) 操作所需的平均时间。  单位 : 毫秒
ReadLatency	每个磁盘 I/O ( 输入 ) 操作所需的平均时间。  单位 : 毫秒
SwapUsage	复制实例上使用的交换空间的大小。  单位 : 字节
NetworkTransmitThroughput	复制实例的传出 ( 传输 ) 网络流量 , 包括客户数据库流量以及用于监控和复制的 AWS DMS 流量。  单位 : 字节/秒
NetworkReceiveThroughput	复制实例的传入 ( 接收 ) 网络流量 , 包括客户数据库流量以及用于监控和复制的 AWS DMS 流量。  单位 : 字节/秒

## 复制任务指标

复制任务监控包括针对以下统计数据的指标。

指标	描述
FullLoadThroughputBandwidthTarget	目标在满负载情况下的传出数据传输速率 , 以每秒 KB 数为单位。
FullLoadThroughputRowsTarget	目标上的完全加载的传出更改 , 以行/秒为单位。
CDCIncomingChanges	在某个时间点等待应用于目标的总更改事件数。请注意 , 这与源终端节点的事务更改速率指标不同。对于该指标 , 较大的值通常表示 AWS DMS 无法及时应用捕获的更改 , 从而导致较高的目标延迟。



指标	描述
CDCChange sMemorySource	内存中累积并等待从源提交的行数。您可以将此指标与 CDCChange sDiskSource 一起查看。
CDCChange sMemoryTarget	内存中累积并等待提交到目标的行数。您可以将此指标与 CDCChange sDiskTarget 一起查看。
CDCChangesDiskSour ce	磁盘上累积并等待从源提交的行数。您可以将此指标与 CDCChange sMemorySource 一起查看。
CDCChangesDiskTarg et	磁盘上累积并等待提交到目标的行数。您可以将此指标与 CDCChange sMemoryTarget 一起查看。
CDCThroughputBandw idthTarget	目标的传出数据传输速率，以每秒 KB 数为单位。CDCThroughput Bandwidth 记录在采样点上传输的传出数据。如果未找到任务网络流 量，则该值为零。由于 CDC 不发布长时间运行的事务，可能不会记录 网络流量。
CDCThroughputRowsS ource	源的传入任务更改，以行/秒为单位。
CDCThroughputRowsT arget	目标的传出任务更改，以行/秒为单位。
CDCLatencySource	<p>从源端点中捕获的最后一个事件与 AWS DMS 实例的当前系统时间戳之间的间隔（秒）。CDCLatencySource 表示源实例和复制实例之间的延迟。高 CDCLatencySource 意味着从源捕获更改的过程会延迟。要确定正在进行的复制中的延迟，您可以将此指标与 CDCLatencyTarget 一起查看。如果 CDCLatencySource 和 CDCLatencyTarget 都很高，请先调查 CDCLatencySource。</p> <p>当源和复制实例之间没有复制延迟时，CDCSourceLatency 可以为 0。当复制任务尝试读取源事务日志中的下一个事件，并且与上次读取源相比没有新事件时，CDCSourceLatency 也可能变为零。出现这种情况时，任务会将 CDCSourceLatency 重置为 0。</p>

指标	描述
CDCLatencyTarget	<p>在目标上等待提交的第一个事件时间戳与 AWS DMS 实例的当前系统时间戳之间的间隔（秒）。目标延迟是复制实例服务器时间，与转发到目标组件的最早未经确认的事件 ID 时间之间的差异。换句话说，目标延迟是复制实例的时间戳，与已应用但未经 TRG 端点确认的最早事件之间的时间戳之差 (99%)。当 CDCLatencyTarget 很高时，它表示将更改事件应用于目标的过程已延迟。要确定正在进行的复制中的延迟，您可以将此指标与 CDCLatencySource 一起查看。如果 CDCLatencyTarget 很高但是 CDCLatencySource 不高，请调查是否：</p> <ul style="list-style-type: none"> <li>• 目标中没有主键或索引</li> <li>• 目标实例或复制实例中出现资源瓶颈</li> <li>• 复制实例和目标之间存在网络问题</li> </ul>
CPU 利用率	<p>任务在多个内核上使用的 CPU 百分比。任务 CPUUtilization 的语义与复制 CPUUtilization 的语义略有不同。如果 1 个 vCPU 已完全使用，则表示 100%；但如果有多于一个 vCPU 在使用中，则值可能高于 100%。</p> <p>单位：百分比</p>
SwapUsage	<p>任务使用的交换区数量。</p> <p>单位：字节</p>
MemoryUsage	<p>任务使用的控制组 (cgroup) memory.usage_in_bytes。DMS 使用 cgroup 来控制内存和 CPU 等系统资源的使用。该指标表示在为任务分配的 cgroup 中，该任务的内存使用量，以 MB 为单位。cgroup 限制基于可供您 DMS 复制实例类使用的资源。memory.usage_in_bytes 由驻留集大小 (RSS)、缓存和内存的交换组件组成。如果需要，操作系统可以回收缓存。我们建议您同时监控复制实例指标 AvailableMemory。</p> <p>AWS DMS 根据 CloudWatch 控制台中 ReplicationTaskIdentifier 和 ReplicationInstanceIdentifier 维度的组合提出这个指标。使用 ReplicationInstanceIdentifier, ReplicationTaskIdentifier 类别查看此指标。</p>

## 查看和管理 AWS DMS 任务日志

在 AWS DMS 迁移过程中，您可以使用 Amazon CloudWatch 记录任务信息。您可以在选择任务设置时启用日志记录。有关更多信息，请参阅[日志记录任务设置](#)。

要查看运行的任务的日志，请按照下列步骤操作：

1. 打开 AWS DMS 控制台，然后从导航窗格中选择数据库迁移任务。此时将显示“数据库迁移任务”对话框。
2. 选择角色的名称。此时将显示“概述详细信息”对话框。
3. 找到迁移任务日志部分，然后选择查看 CloudWatch 日志。

此外，您可以使用 AWS CLI 或 AWS DMS API 查看有关任务日志的信息。为此，请使用 `describe-replication-instance-task-logs` AWS CLI 命令或 AWS DMS API 操作 `DescribeReplicationInstanceTaskLogs`。

例如，以下 AWS CLI 命令显示采用 JSON 格式的任务日志元数据。

```
$ aws dms describe-replication-instance-task-logs \  
  --replication-instance-arn arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAY
```

命令中的示例响应如下所示。

```
{  
  "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:CDSFSFSFFFSSUFCAY"  
}
```

在此响应中，只有一个与复制实例关联的任务日志 (mysql-to-ddb)。此日志的大小为 3,726,124 字节。

您可以使用由 `describe-replication-instance-task-logs` 返回的信息来诊断和排查任务日志的问题。例如，如果您对任务启用详细的调试日志记录，则任务日志将快速增大，可能会消耗复制实例上的所有可用存储空间，并且导致实例状态更改为 `storage-full`。通过描述任务日志，您可以确定不再需要的日志；然后可以将其删除，从而释放存储空间。

要删除任务的任务日志，请将任务设置 `DeleteTaskLogs` 设置为 `true`。例如，以下 JSON 使用 AWS CLI `modify-replication-task` 命令或 AWS DMS API `ModifyReplicationTask` 操作在修改任务时删除任务日志。

```
{
  "Logging": {
    "DeleteTaskLogs": true
  }
}
```

## 使用 AWS DMS 记录 AWS CloudTrail API 调用

AWS DMS 与 AWS CloudTrail 集成，后者是在 AWS 中记录用户、角色或 AWS DMS 服务所执行操作的服务。CloudTrail 将对 AWS DMS 的所有 API 调用均作为事件捕获，包括来自 AWS DMS 控制台的调用和对 AWS DMS API 操作的代码调用。如果您创建跟踪记录，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 AWS DMS 的事件）。如果您不配置跟踪记录，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 AWS DMS 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，请参阅 [《AWS CloudTrail 用户指南》](#)。

### CloudTrail 中的 AWS DMS 信息

在您创建 AWS 账户时，将在该账户上启用 CloudTrail。当 AWS DMS 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history（事件历史记录）中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅 [使用 CloudTrail 事件历史记录查看事件](#)。

要持续记录 AWS 账户中的事件（包括 AWS DMS 的事件），请创建跟踪。通过跟踪记录，CloudTrail 可将日志文件传送至 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪记录来自 AWS 分区中的所有 AWS 区域的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅：

- [创建跟踪概览](#)
- [CloudTrail 支持的服务和集成](#)
- [为 CloudTrail 配置 Amazon SNS 通知](#)
- [从多个 AWS 区域接收 CloudTrail 日志文件](#)和[从多个账户接收 CloudTrail 日志文件](#)

CloudTrail 记录所有 AWS DMS 操作，[AWS Database Migration Service API 参考](#)中介绍了这些操作。例如，对 `CreateReplicationInstance`、`TestConnection` 和 `StartReplicationTask` 操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 IAM 用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其它 AWS 服务发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

## 了解 AWS DMS 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 `RebootReplicationInstance` 操作。

```
{
  "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 上下文日志记录

AWS DMS 使用上下文日志记录为您提供有关正在进行的迁移的信息。上下文日志记录将诸如以下内容的信息写入任务的 CloudWatch 日志：



- 有关任务与源数据库和目标数据库连接的信息。
- 复制任务行为。您可以使用任务日志来诊断复制问题。
- AWS DMS 在源数据库和目标数据库上执行的不带数据的 SQL 语句。您可以使用 SQL 日志来诊断意外的迁移行为。
- 流式传输每个 CDC 事件的位置详细信息。

上下文日志记录仅在 AWS DMS 版本 3.5.0 或更高版本中可用。

默认情况下 AWS DMS 开启上下文日志记录。要控制上下文日志记录，请将 `EnableLogContext` 任务设置设为 `true` 或 `false`，或者在控制台中修改任务。

AWS DMS 每三分钟将上下文日志信息写入 CloudWatch 日志的复制任务。请确保您的复制实例有足够的空间来存储其应用程序日志。有关使用管理任务日志的更多信息，请参阅[查看和管理 AWS DMS 任务日志](#)。

## 主题

- [对象类型](#)
- [日志记录示例](#)
- [限制](#)

## 对象类型

AWS DMS 为以下对象类型在 CloudWatch 中生成上下文日志。

对象类型	描述
TABLE_NAME	这些日志条目包含当前任务映射规则范围内表的相关信息。您可以使用这些条目来检查迁移期间特定时间段内的表事件。
SCHEMA_NAME	这些日志条目包含当前任务映射规则所使用的架构的相关信息。您可以使用这些条目来确定在迁移期间，AWS DMS 在特定时间段内使用了哪个架构。

对象类型	描述
TRANSACTION_ID	这些条目包含从源数据库捕获的每个 DML/DDL 更改的事务 ID。您可以使用这些日志条目来确定在给定事务期间发生了哪些更改。
CONNECTION_ID	这些条目包含连接 ID。您可以使用这些日志条目来确定 AWS DMS 在每个迁移步骤中使用哪个连接。
STATEMENT	这些条目包含用于获取、处理和应用每项迁移更改的 SQL 代码。
STREAM_POSITION	这些条目包含源数据库上的每个迁移操作在事务日志文件中的位置。这些条目的格式因源数据库引擎类型而异。在配置仅 CDC 的复制时，您还可以使用此信息来确定恢复检查点的起始位置。

## 日志记录示例

本节提供了可用于监控复制和诊断复制问题的日志记录示例。

### 连接日志示例

本节提供了包含连接 ID 的日志示例。

```
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)
```

## 任务行为日志示例

本节提供了有关复制任务日志行为的日志示例。您可以使用此信息来诊断复制问题，例如状态为 IDLE 的任务。

以下 SOURCE\_CAPTURE 日志表明没有可从源数据库日志文件中读取的事件，并且包含的 TARGET\_APPLY 记录表明，没有从 AWS DMS CDC 组件收到任何可应用到目标数据库的事件。这些事件还包含先前应用的与事件相关的上下文详细信息。

```
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 语句日志示例

本节提供了在源数据库和目标数据库上运行 SQL 语句的日志示例。您在日志中看到的 SQL 语句仅显示 SQL 语句，不显示数据。以下 TARGET\_APPLY 日志显示了在目标上运行的 INSERT 语句。

```
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
(?,?,?,?/?),
```

## 限制

以下限制适用于 AWS DMS 上下文日志记录：

- AWS DMS 为所有端点类型创建极少的日志记录，仅以下端点类型有特定于引擎的大量上下文日志记录可用。我们建议在使用这些端点类型时开启上下文日志记录。
  - MySQL
  - PostgreSQL
  - Oracle
  - Microsoft SQL Server

- MongoDB/Amazon DocumentDB
- Amazon S3

# 在 AWS Database Migration Service 中使用 Amazon EventBridge 事件和通知

您可以使用 Amazon EventBridge，在发生 AWS DMS 事件（例如创建或删除复制实例）时提供通知。根据事件规则的定义，EventBridge 接收事件并路由事件的通知。您可以按照 Amazon EventBridge 在 AWS 区域中支持的任何形式使用通知。有关使用 EventBridge 的更多信息，请参阅《Amazon EventBridge 用户指南》中的[什么是 Amazon EventBridge？](#)。

## Note

AWS DMS 3.4.5 及更高版本支持处理 Amazon EventBridge 事件。

EventBridge 收到一个事件，指示 AWS DMS 环境中出现更改，并应用规则将该事件路由到通知机制。根据事件的结构（称为事件模式），规则将事件与通知机制匹配。

AWS DMS 将这些事件分组为您可以应用事件规则的类别，以便您在出现该类事件时收到通知。例如，对于给定复制实例，假设您将 EventBridge 事件规则应用于“创建”类别。然后，每当发生与创建相关的事件会影响您的复制实例时，您都会收到通知。如果您对复制实例的“配置更改”类别应用规则，则您将在复制实例的配置更改时收到通知。有关 AWS DMS 提供的事件类别的列表，请参阅下面的 AWS DMS 事件类别和事件消息。

## Note

要允许从 [events.amazonaws.com](https://events.amazonaws.com) 发布内容，请确保更新您的 Amazon SNS 主题的访问策略。有关更多信息，请参阅《Amazon EventBridge 用户指南》中的[为 Amazon EventBridge 使用基于资源的策略](#)。

有关将事件订阅移至 Amazon EventBridge 的更多信息，请参阅[将活跃的活动订阅从 DMS 迁移到 Amazon EventBridge](#)。

有关在 Amazon SNS 上使用文本消息的更多信息，请参阅[使用 Amazon SNS 发送和接收 SMS 通知](#)。

## 将 Amazon EventBridge 事件规则用于 AWS DMS

Amazon EventBridge 会将事件通知发送到您在创建 EventBridge 事件规则时提供的地址。您可能需要创建多个不同的规则。例如，您可能会创建一个规则来接收所有事件通知，创建另一个规则仅接收生产 DMS 资源的关键事件。您还可以在 EventBridge 中打开或关闭事件通知。

### 创建对 AWS DMS 事件作出反应的 Amazon EventBridge 规则

- 按照《Amazon EventBridge 用户指南》的[创建对事件作出反应的 Amazon EventBridge 规则](#)中的说明执行步骤，并为 AWS DMS 事件创建规则：
  - a. 指定 EventBridge 收到与规则中的事件匹配的事件时要采取的通知操作。当事件匹配时，EventBridge 会发送事件并调用规则中定义的操作。
  - b. 对于 Service provider ( 服务提供商 )，选择 AWS。
  - c. 对于服务名称，选择 Database Migration Service (DMS)。

然后您将开始接收事件通知。

以下 JSON 示例显示 AWS DMS 服务的 EventBridge 事件模型。

```
{
  "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:AAAABBBB0CCCCDDDEEEEEE1FFFF2GGG3FFFFFFF3"
  ],
  "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"
  }
}
```

有关您可以接收通知的类型和事件的列表，请参阅下面的章节。

## AWS DMS 事件类别和事件消息

AWS DMS 会在您可以标识的类别中生成大量事件。每个类别应用到一个复制实例或复制任务源类型。

### 主题

- [ReplicationInstance 事件消息](#)
- [ReplicationTask 事件消息](#)
- [复制事件消息](#)

### ReplicationInstance 事件消息

下表显示了 ReplicationInstance 源类型可能的类别和事件。

类别	事件 ID	描述
创建	DMS-EVENT-0067	正在创建复制实例。
删除	DMS-EVENT-0066	正在删除复制实例。
配置更改	DMS-EVENT-0012	正在更改此复制实例的复制实例类。
配置更改	DMS-EVENT-0018	正在增加复制实例的存储。
配置更改	DMS-EVENT-0024	复制实例正在过渡到多可用区配置。
配置更改	DMS-EVENT-0030	复制实例正在过渡到单可用区配置。
维护	DMS-EVENT-0026	复制实例正在进行脱机维护。复制实例当前不可用。
创建	DMS-EVENT-0005	复制实例已创建。
删除	DMS-EVENT-0003	复制实例已删除。

类别	事件 ID	描述
配置更改	DMS-EVENT-0014	此复制实例的复制实例类已更改。
配置更改	DMS-EVENT-0017	复制实例的存储已增加。
配置更改	DMS-EVENT-0025	复制实例已过渡到多可用区配置。
配置更改	DMS-EVENT-0029	复制实例已过渡到单可用区配置。
维护	DMS-EVENT-0047	复制实例上的管理软件已更新。
维护	DMS-EVENT-0027	复制实例的脱机维护已完成。现在可以使用复制实例。
维护	DMS-EVENT-0068	复制实例处于无法升级的状态。
失效转移	DMS-EVENT-0034	如果过于频繁地请求失效转移，则会发生此事件而不是常规失效转移事件。
失败	DMS-EVENT-0031	复制实例置于 %s 状态。
失败	DMS-EVENT-0036	由于不兼容的网络，复制实例失败。
失败	DMS-EVENT-0037	当服务无法访问用于加密数据卷的 KMS 密钥时。
失败		复制实例使用了不兼容的参数
失效转移		等待进入能够安全地启动用户所请求失效转移的状态时超时
失效转移	DMS-EVENT-0013	为多可用区复制实例启用了失效转移。
失效转移	DMS-EVENT-0049	多可用区复制实例的失效转移已完成。
失效转移	DMS-EVENT-0050	多可用区激活已开始。
失效转移	DMS-EVENT-0051	多可用区激活已完成。



类别	事件 ID	描述
StateChange		常规和慢速查询日志已自动轮换为 %s
StateChange		AWS DMS 无法访问应用程序实例 %s 的 KMS 加密密钥。这可能是由于密钥被禁用或 AWS DMS 无法访问密钥所致。如果这种情况继续，应用程序将进入无法访问的状态。有关更多详细信息，请参阅 AWS DMS 文档中的故障排除部分。
StateChange		AWS DMS 现在可以成功访问应用程序实例 %s 的 KMS 加密密钥。
StateChange		Amazon DMS 无法访问应用程序实例 %s 的 KMS 加密密钥。此应用程序将进入无法访问的状态。有关更多详细信息，请参阅 Amazon DMS 文档中的故障排除部分。
StateChange		作为复制实例创建的一部分，应用程序在 HM 上重新启动
StateChange		作为复制实例删除的一部分，应用程序在 HM 上关闭
失效转移	DMS-EVENT-0015	多可用区失效转移到备用状态已完成。
LowStorage	DMS-EVENT-0007	复制实例的免费存储不足。
LowStorage		已分配的 inode 已用尽 – 扩展存储空间以解决问题

## ReplicationTask 事件消息

下表显示了 ReplicationTask 源类型可能的类别和事件。

类别	事件 ID	描述
失败	DMS-EVENT-0078	复制任务已失败。
失败	DMS-EVENT-0082	清理任务数据的调用失败。
状态更改	DMS-EVENT-0081	已请求重新加载表详细信息。
状态更改		复制任务已复制。
状态更改		复制任务的复制已失败。
状态更改		复制任务已移动。
状态更改		复制任务移动已失败。
状态更改		目标任务的创建失败。
状态更改		复制任务评测运行已开始。
状态更改		复制任务评测运行已成功完成。
状态更改		复制任务评测运行已完成，但出现故障。
StateChange		复制任务评测运行已完成，但出现警告。
StateChange		复制任务评测运行已完成，但出现错误。
StateChange		复制任务评测运行 %s 已取消。
StateChange		复制任务评测运行 %s 已删除。
StateChange		复制任务评测预置资源失败。

类别	事件 ID	描述
StateChange		复制任务已失败。
创建		复制任务已创建。
ConfigurationChange		复制任务已修改。
失败		复制任务已失败。
StateChange	DMS-EVENT-0091	读取暂停，已达到交换文件限制。
StateChange	DMS-EVENT-0092	读取暂停，已达到磁盘使用量限制。
StateChange	DMS-EVENT-0093	读取暂停，已达到磁盘使用量限制。
StateChange	DMS-EVENT-0093	读取已恢复。
StateChange	DMS-EVENT-0069	复制任务已启动，taskType：%s，startType：%s
StateChange	DMS-EVENT-0079	复制任务已停止
删除	DMS-EVENT-0073	复制任务已删除。

## 复制事件消息

下表显示了复制源类型可能的类别和事件。

类别	描述
状态更改	DMS 复制纵向扩展事件。

类别	描述
状态更改	DMS 复制纵向缩减事件。
状态更改	DMS 复制扩展活动已完成。
状态更改	DMS 复制已创建。
状态更改	DMS 复制正在初始化。
状态更改	DMS 复制正在为元数据集合准备资源。
状态更改	正在测试与 DMS 复制相关的连接。
状态更改	DMS 复制正在提取元数据
状态更改	DMS 复制正在计算容量
状态更改	DMS 复制正在预置其容量
状态更改	DMS 复制已预置。
状态更改	DMS 复制已开始
状态更改	DMS 复制正在运行。
状态更改	DMS 复制正在停止。
状态更改	DMS 复制已停止。
状态更改	正在修改 DMS 复制。
状态更改	正在删除 DMS 复制。
状态更改	DMS 复制正在取消预置其容量
状态更改	DMS 复制已取消预置。
失败	DMS 复制已失败。

# 在 AWS Database Migration Service 中使用 Amazon SNS 事件和通知

从 AWS DMS 版本 3.4.5 及更高版本开始，我们建议您使用 Amazon EventBridge，在 AWS DMS 事件发生时提供通知。有关将 AWS DMS 与 EventBridge 事件结合使用的更多信息，请参阅[在 AWS Database Migration Service 中使用 Amazon EventBridge 事件和通知](#)。

## 将事件订阅移至 Amazon EventBridge

您可以使用以下 AWS CLI 命令，将活跃的事件订阅从 DMS 迁移到 Amazon EventBridge，一次最多可迁移 10 个订阅。

```
update-subscriptions-to-event-bridge [--force-move | --no-force-move]
```

默认情况下，只有当您的复制实例当前版本为 AWS DMS 3.4.5 及更高版本时，AWS DMS 才会迁移活跃的事件订阅。要覆盖此默认行为，请使用 `--force-move` 选项。但是，如果您的复制实例未升级，则使用 Amazon EventBridge 时某些类型的事件可能不可用。

要运行 `update-subscriptions-to-event-bridge` CLI 命令，AWS Identity and Access Management (IAM) 用户必须具有以下策略权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "events:PutTargets",
        "events:EnableRule",
        "events:PutRule"
      ],
      "Resource": "*"
    }
  ]
}
```

有关将订阅移至 EventBridge 的更多信息，请参阅《AWS Database Migration Service API 参考》中的 [UpdateSubscriptionsToEventBridge](#)。

## 使用 Amazon SNS 事件和通知

AWS DMS 版本 3.4.5 及更早版本支持使用事件和通知，如下所述。

AWS Database Migration Service (AWS DMS) 可以使用 Amazon Simple Notification Service (Amazon SNS)，在发生 AWS DMS 事件（例如创建或删除复制实例）时提供通知。您可以通过 AWS 区域的 Amazon SNS 支持的任何形式使用这些通知，如电子邮件、短信或者调用 HTTP 端点。

AWS DMS 将这些事件分组为您可以订阅的类别，以便您在出现该类事件时收到通知。例如，如果您订阅了指定复制实例的“创建”类别，则只要发生了与创建相关的事件影响到您的复制实例，您就会收到通知。如果您订阅了复制实例的“配置更改”类别，则您将在复制实例的配置更改时收到通知。您还将在事件通知订阅更改时收到通知。有关 AWS DMS 提供的事件类别的列表，请参阅后文中的 [AWS DMS 事件类别和 SNS 通知的事件消息](#)。

AWS DMS 将事件通知发送到您在创建事件订阅时提供的地址。您可能希望创建多个不同的订阅，如使用一个订阅接收所有事件通知，并使用另一个订阅接收仅针对生产 DMS 资源的关键事件。您可以在 AWS DMS 控制台中取消选择已启用选项，或者使用 AWS DMS API 将 Enabled 参数设置为 false，轻松地禁用通知而不删除订阅。

### Note

目前，在支持 Amazon SNS 的所有 AWS 区域中，使用 SMS 文本消息的 AWS 事件通知可用于 AWS DMS 资源。有关 Amazon SNS 支持 SMS 消息的 AWS 区域和国家/地区列表，请参阅 [支持的区域和国家/地区](#)。

有关在 SNS 上使用文本消息的更多信息，请参阅 [使用 Amazon SNS 发送和接收 SMS 通知](#)。AWS DMS 事件通知不同于 CloudWatch 或 EventBridge 中的 CloudTrail 事件。任何 API 调用都可以生成 CloudTrail 事件通知。只有在发生了 DMS 事件时，DMS 才会发送通知。

AWS DMS 使用订阅标识符来标识各个订阅。您可以将多个 AWS DMS 事件订阅发布到同一个 Amazon SNS 主题。在您使用事件通知时，将会收取 Amazon SNS 费用；有关 Amazon SNS 计费的信息，请参阅 [Amazon SNS 定价](#)。

要订阅 Amazon SNS 上的 AWS DMS 事件，您可以使用以下过程：

1. 创建 Amazon SNS 主题。在主题中，指定您要接收的通知类型，以及要将通知发送到的地址或号码。
2. 使用 AWS Management Console、AWS CLI 或 AWS DMS API 创建 AWS DMS 事件通知订阅。
3. AWS DMS 将批准电子邮件或者 SMS 消息发送到您在订阅时提交的地址。要确认订阅，请单击批准电子邮件或 SMS 消息中的链接。
4. 确认订阅后，订阅状态即会在 AWS DMS 控制台的事件订阅部分中更新。
5. 您将开始接收事件通知。

有关您可以接收通知的类型和事件的列表，请参阅下面的章节。有关订阅和使用 AWS DMS 事件通知的更多详情，请参阅[使用 SNS 订阅 AWS DMS 事件通知](#)。

## AWS DMS 事件类别和 SNS 通知的事件消息

### Important

从 AWS DMS 版本 3.4.5 及更高版本开始，我们建议您使用 Amazon EventBridge，在 AWS DMS 事件发生时提供通知。有关将 AWS DMS 与 EventBridge 事件结合使用的更多信息，请参阅[在 AWS Database Migration Service 中使用 Amazon EventBridge 事件和通知](#)。

AWS DMS 会生成各种类别的大量事件，您可以使用 AWS DMS 控制台或者 AWS DMS API 对它们进行订阅。每个类别应用于一种源类型；当前 AWS DMS 支持复制实例和复制任务源类型。

下表显示了复制实例源类型可能的类别和事件。

类别	DMS 事件 ID	描述
配置更改	DMS-EVENT-0012	正在更改此复制实例的复制实例类。
配置更改	DMS-EVENT-0014	此复制实例的复制实例类已更改。
配置更改	DMS-EVENT-0018	正在增加复制实例的存储。
配置更改	DMS-EVENT-0017	复制实例的存储已增加。
配置更改	DMS-EVENT-0024	复制实例正在过渡到多可用区配置。

类别	DMS 事件 ID	描述
配置更改	DMS-EVENT-0025	复制实例已完成过渡到多可用区配置。
配置更改	DMS-EVENT-0030	复制实例正在过渡到单可用区配置。
配置更改	DMS-EVENT-0029	复制实例已过渡到单可用区配置。
创建	DMS-EVENT-0067	正在创建复制实例。
创建	DMS-EVENT-0005	复制实例已创建。
删除	DMS-EVENT-0066	正在删除复制实例。
删除	DMS-EVENT-0003	复制实例已删除。
维护	DMS-EVENT-0047	复制实例上的管理软件已更新。
维护	DMS-EVENT-0026	复制实例正在进行脱机维护。复制实例当前不可用。
维护	DMS-EVENT-0027	复制实例的脱机维护已完成。现在可以使用复制实例。
维护	DMS-EVENT-0068	复制实例处于无法升级的状态。
LowStorage	DMS-EVENT-0007	复制实例已使用其分配的存储空间的 90% 以上。您可以使用可用存储空间指标监控复制实例的存储空间。
失效转移	DMS-EVENT-0013	为多可用区复制实例启用了失效转移。
失效转移	DMS-EVENT-0049	多可用区复制实例的失效转移已完成。
失效转移	DMS-EVENT-0015	多可用区失效转移到备用状态已完成。
失效转移	DMS-EVENT-0050	多可用区激活已开始。
失效转移	DMS-EVENT-0051	多可用区激活已完成。



类别	DMS 事件 ID	描述
失效转移	DMS-EVENT-0034	如果过于频繁地请求失效转移，则会发生此事件而不是常规失效转移事件。
失败	DMS-EVENT-0031	复制实例遇到了存储故障。
失败	DMS-EVENT-0036	由于不兼容的网络，复制实例失败。
失败	DMS-EVENT-0037	服务无法访问用于加密数据卷的 AWS KMS 密钥。

下表显示了复制任务源类型可能的类别和事件。

类别	DMS 事件 ID	描述
状态更改	DMS-EVENT-0069	复制任务已开始。
状态更改	DMS-EVENT-0081	已请求重新加载表详细信息。
状态更改	DMS-EVENT-0079	复制任务已停止。
状态更改	DMS-EVENT-0091	读取暂停，已达到交换文件限制。
状态更改	DMS-EVENT-0092	读取暂停，已达到磁盘使用量限制。
状态更改	DMS-EVENT-0093	读取已恢复。
失败	DMS-EVENT-0078	复制任务已失败。
失败	DMS-EVENT-0082	删除任务的调用未能清理任务数据。
配置更改	DMS-EVENT-0080	复制任务已修改。
删除	DMS-EVENT-0073	复制任务已删除。
创建	DMS-EVENT-0074	复制任务已创建。

以下示例显示了具有“状态更改”类别的 AWS DMS 事件订阅。

```
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
```

## 使用 SNS 订阅 AWS DMS 事件通知

### Important

从 AWS DMS 版本 3.4.5 及更高版本开始，我们建议您使用 Amazon EventBridge，在 AWS DMS 事件发生时提供通知。有关将 AWS DMS 与 EventBridge 事件结合使用的更多信息，请参阅[在 AWS Database Migration Service 中使用 Amazon EventBridge 事件和通知](#)。

您可以创建 AWS DMS 事件通知订阅，这样就能在发生 AWS DMS 事件时收到通知。创建订阅最简单的方式是使用 AWS DMS 控制台。在通知订阅中，您可以选择发送通知的方式和位置。您可以指定希望收到其通知的源类型；当前 AWS DMS 支持复制实例和复制任务源类型。并且，根据选择的源类型，您需要选择希望接收其事件通知的事件类别并标识源。

## 使用 AWS Management Console

### Important

从 AWS DMS 版本 3.4.5 及更高版本开始，我们建议您使用 Amazon EventBridge，在 AWS DMS 事件发生时提供通知。有关将 AWS DMS 与 EventBridge 事件结合使用的更多信息，请参阅[在 AWS Database Migration Service 中使用 Amazon EventBridge 事件和通知](#)。

### 使用控制台订阅 Amazon SNS 上的 AWS DMS 事件通知

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。

如果以 IAM 用户身份登录，请确保具有适当的 AWS DMS 访问权限。

2. 在导航窗格中，选择事件订阅。
3. 在事件订阅页面中，选择创建事件订阅。
4. 在创建事件订阅页面中，执行以下操作：
  - a. 在详细信息下，对于名称，输入事件通知订阅的名称。
  - b. 选择已启用以启用订阅。如果您希望创建订阅但尚不发送通知，可选择已禁用。
  - c. 在目标下，选择现有主题、创建新电子邮件主题或新建 SMS 主题以发送通知。请确保您有可以发送通知的现有 Amazon SNS 主题，否则必须创建主题。如果您创建主题，则可以输入将通知发送到的电子邮件地址。
  - d. 在事件源下，对于源类型，请选择一种源类型。仅有的选项是复制实例和复制任务。
  - e. 根据选定源类型的情况，选择您希望接收其事件通知的事件类别和源。

# Create event subscription

## Details

### Name

The name for your event subscription

EventSubscriptionExample

Enabled

## Target

Send notification to

- Existing topics
- Create new email topic
- Create new SMS topic

### Topic name

ProdEventSubscription

### With these recipients

Email addresses or phone numbers of SMS enabled devices to send the notifications to

user@domain.com

## Event source

### Source type

Source Type of resource this subscription will consume events from

replication-instance

### Event categories

- All event categories
- Select specific event categories

### Replication instance

- All instances
- Select specific instances

f. 选择创建事件订阅。

AWS DMS 控制台会表明正在创建订阅。

**Note**

您也可以使用 AWS DMS API 和 CLI 创建 Amazon SNS 事件通知订阅。有关更多信息，请参阅《AWS DMS API 参考》中的 [CreateEventSubscription](#) 和《AWS DMS CLI 参考》文档中的 [create-event-subscription](#)。

## 验证您的 SNS 主题的访问策略

您的 SNS 访问策略需要允许 AWS DMS 将事件发布到 SNS 主题的权限。您可以按照以下过程中的说明验证和更新您的访问策略。

### 验证访问策略

1. 打开 Amazon SNS 控制台。
2. 在导航面板中，选择主题，然后选择您要接收 DMS 通知的主题。
3. 选择访问策略选项卡。

如果您的 SNS 访问策略不允许 AWS DMS 向 SNS 主题发布事件，则可以更新策略。

### 更新访问策略

1. 在主题页面的详细信息部分，选择编辑。
2. 展开访问策略部分，然后将以下策略附加到 JSON 编辑器中。

```
{
  "Sid": "dms-allow-publish",
  "Effect": "Allow",
  "Principal": {
    "Service": "dms.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "your-SNS-topic-ARN"
}
```

我们建议您通过指定 `aws:SourceArn` 条件来进一步限制对您 SNS 主题的访问权限，即发布事件到主题的 DMS EventSubscription ARN。

```
...
"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. 选择保存更改。

# AWS DMS 数据验证

## 主题

- [复制任务统计数据](#)
- [使用 Amazon CloudWatch 的复制任务统计数据](#)
- [在任务期间重新验证表](#)
- [使用 JSON 编辑器修改验证规则](#)
- [仅验证任务](#)
- [故障排除](#)
- [Redshift 验证性能](#)
- [限制](#)
- [Amazon S3 目标数据验证](#)

AWS DMS 提供数据验证支持，以确保将数据准确地从源迁移到目标。如果启用，则在对表执行完全加载后立即开始验证。对于启用 CDC 的任务，验证会在发生更改时比较增量更改。

在数据验证期间，AWS DMS 将源中的每一行与目标中的相应行进行比较，确认这些行是否包含相同的数据并报告任何不匹配的情况。为完成此操作，AWS DMS 发出相应的查询以检索数据。请注意，这些查询将占用源和目标中的额外资源以及额外的网络资源。

对于启用了验证的仅 CDC 任务，在开始验证新数据之前，将对表中所有已存在的数据进行验证。

数据验证可用于以下 AWS DMS 支持作为源端点的数据库：

- Oracle
- 兼容 PostgreSQL 的数据库 ( PostgreSQL、Aurora PostgreSQL 或 Aurora Serverless for PostgreSQL )
- 兼容 MySQL 的数据库 ( MySQL、MariaDB、Aurora MySQL 或 Aurora Serverless for MySQL )
- Microsoft SQL Server
- IBM Db2 LUW

数据验证可用于以下 AWS DMS 支持作为目标端点的目标数据库：

- Oracle

- 兼容 PostgreSQL 的数据库 ( PostgreSQL、Aurora PostgreSQL 或 Aurora Serverless for PostgreSQL )
- 兼容 MySQL 的数据库 ( MySQL、MariaDB、Aurora MySQL 或 Aurora Serverless for MySQL )
- Microsoft SQL Server
- IBM Db2 LUW
- Amazon Redshift
- Amazon S3。有关验证 Amazon S3 目标数据的信息，请参阅 [Amazon S3 目标数据验证](#)。

有关支持的终端节点的更多信息，请参阅[使用 AWS DMS 端点](#)。

除了迁移本身所需的时间以外，数据验证还需要占用额外的时间。所需的额外时间取决于迁移的数据量。

有关这些设置的更多信息，请参阅 [数据验证任务设置](#)。

有关 JSON 文件中的 ValidationSettings 任务设置的示例，请参阅[任务设置示例](#)。

## 复制任务统计数据

如果启用了数据验证，AWS DMS 将在表级别提供以下统计数据：

- ValidationState – 表的验证状态。参数可能具有以下值：
  - Not enabled – 没有在迁移任务中为表启用验证。
  - Pending records – 表中的某些记录正在等待验证。
  - 不匹配的记录 – 表中的某些记录在源和目标之间不匹配。可能会因多种原因而发生不匹配；有关更多信息，请参阅目标终端节点上的 `awsdms_control.awsdms_validation_failures_v1` 表。
  - Suspended records – 无法验证表中的某些记录。
  - No primary key – 无法验证表，因为该表没有主键。
  - Table error – 未验证表，因为该表处于错误状态并且未迁移某些数据。
  - 已验证 – 表中的所有行已验证。如果更新表，则可能会变为非 Validated 状态。
  - Error – 无法验证表，因为出现意外错误。
  - 等待验证 – 表正在等待验证。
  - 准备表 – 准备迁移任务中启用的表以进行验证。
  - 等待重新验证 – 表更新后，表中的所有行在等待验证。



- ValidationPending – 已迁移到目标但尚未验证的记录数。
- ValidationSuspended – AWS DMS 无法比较的记录数。例如，如果持续更新源中的记录，AWS DMS 将无法比较源和目标。
- ValidationFailed – 未通过数据验证阶段的记录数。

有关 JSON 文件中的 ValidationSettings 任务设置的示例，请参阅[任务设置示例](#)。

您可以使用控制台、AWS CLI 或 AWS DMS API 查看数据验证信息。

- 在控制台中，您可以选择在创建或修改任务时验证该任务。要使用控制台查看数据验证报告，请在任务页中选择任务，然后在详细信息部分中选择表统计数据选项卡。
- 在创建或修改任务以开始数据验证时，使用 CLI 将 EnableValidation 参数设置为 true。以下示例创建一个任务并启用数据验证。

```
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"}]}'
```

可以使用 describe-table-statistics 命令接收 JSON 格式的数据验证报告。以下命令显示数据验证报告。

```
aws dms describe-table-statistics --replication-task-arn arn:aws:dms:us-
east-1:5731014:
rep:36KWVMB7Q
```

该报告类似于以下内容。

```
{
  "ReplicationTaskArn": "arn:aws:dms:us-west-2:5731014:task:VFPPTYKK2RYSI",
```

```
"TableStatistics": [
  {
    "ValidationPendingRecords": 2,
    "Inserts": 25,
    "ValidationState": "Pending records",
    "ValidationSuspendedRecords": 0,
    "LastUpdateTime": 1510181065.349,
    "FullLoadErrorRows": 0,
    "FullLoadCondtnlChkFailedRows": 0,
    "Ddls": 0,
    "TableName": "t_binary",
    "ValidationFailedRecords": 0,
    "Updates": 0,
    "FullLoadRows": 10,
    "TableState": "Table completed",
    "SchemaName": "d_types_s_sqlserver",
    "Deletes": 0
  }
]
```

- 使用 AWS DMS API，使用 `CreateReplicationTask` 操作创建任务，然后将 `EnableValidation` 参数设置为 `true`，以便按任务验证迁移的数据。可以使用 `DescribeTableStatistics` 操作接收 JSON 格式的数据验证报告。

## 使用 Amazon CloudWatch 的复制任务统计数据

启用 Amazon CloudWatch 后，AWS DMS 提供以下复制任务统计数据：

- `ValidationSucceededRecordCount` – AWS DMS 每分钟验证的行数。
- `ValidationAttemptedRecordCount` – 每分钟尝试验证的行数。
- `ValidationFailedOverallCount` – 验证失败的行数。
- `ValidationSuspendedOverallCount` – 验证已暂停的行数。
- `ValidationPendingOverallCount` – 验证待处理的行数。
- `ValidationBulkQuerySourceLatency` – AWS DMS 可以批量验证数据，特别是在完全加载和持续复制期间有许多更改的特定场景中。此指标指示从源终端节点读取批量数据所需的延迟。
- `ValidationBulkQueryTargetLatency` – AWS DMS 可以批量验证数据，特别是在完全加载和持续复制期间有许多更改的特定场景中。此指标指示从目标终端节点读取批量数据所需的延迟。

- `ValidationItemQuerySourceLatency` – 在持续复制期间，数据验证可以标识持续更改并验证这些更改。此指标指示从源中读取这些更改时的延迟。验证可根据更改数运行比所需数量更多的查询，前提是验证期间出错。
- `ValidationItemQueryTargetLatency` – 在持续复制期间，数据验证可以标识持续更改并逐行验证这些更改。此指标向我们提供从目标读取这些更改时的延迟。验证可根据更改数运行比所需数量更多的查询，前提是验证期间出错。

要从启用 CloudWatch 的统计数据中收集数据验证信息，请在使用控制台创建或修改任务时选择启用 CloudWatch 日志。然后，要查看数据验证信息并确保您的数据已准确地从源迁移到目标，请执行以下操作。

1. 从数据库迁移任务页面选择任务。
2. 选择 CloudWatch 指标选项卡。
3. 从下拉菜单中选择验证。

## 在任务期间重新验证表

在任务运行时，您可以请求 AWS DMS 执行数据验证。

## AWS Management Console

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。

如果以 AWS Identity and Access Management (IAM) 用户身份登录，请确保具有适当的 AWS DMS 访问权限。有关所需权限，请参阅[使用 AWS DMS 所需的 IAM 权限](#)。

2. 从导航窗格中选择任务。
3. 选择具有要重新验证的表的正在运行的任务。
4. 选择表统计数据选项卡。
5. 选择您要重新验证的表（一次最多可选择 10 个表）。如果任务不再运行，则您无法重新验证该表。
6. 选择 Revalidate (重新验证)。

## 使用 JSON 编辑器修改验证规则

要从 AWS DMS 控制台使用 JSON 编辑器向任务添加验证规则，请执行以下操作：

1. 选择数据库迁移任务。
2. 从迁移任务列表中选择您的任务。
3. 如果您的任务正在运行，请从操作下拉菜单中选择停止。
4. 任务停止后，要修改您的任务，请从操作下拉菜单中选择修改。
5. 在表映射部分，选择 JSON 编辑器，然后将您的验证规则添加到表映射中。

例如，您可以添加以下验证规则以在源上运行替换函数。在这种情况下，如果验证规则遇到 null 字节，则会将其验证为空格。

```
{
  "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}"
}
```

## 仅验证任务

您可以创建仅验证的任务来预览和验证数据，而无需执行任何迁移或数据复制。要创建仅验证的任务，请将 `EnableValidation` 和 `ValidationOnly` 设置设为 `true`。启用 `ValidationOnly` 后，还需要满足其他要求。有关更多信息，请参阅 [数据验证任务设置](#)。

在报告许多故障时，对于仅完全加载的迁移类型，仅验证任务的完成速度要比其 CDC 的等同验证任务快得多。但是，对源端点或目标端点的更改会被报告为完全加载模式失败，这可能是一个缺点。

CDC 仅验证任务会根据平均延迟来延迟验证，并在报告失败之前多次重试失败的验证。如果大多数数据比较都导致失败，那么 CDC 模式的仅验证任务会非常缓慢，这可能是一个不利之处。

仅验证的任务必须与复制任务的方向设置相同，尤其是对于 CDC。这是因为 CDC 仅验证任务会根据源上的更改日志，检测哪些行已更改并需要重新验证。如果目标被指定为源，则它只知道 DMS 发送给目标的更改，不能保证捕获复制错误。

## 完全加载仅验证

从 AWS DMS 3.4.6 及更高版本开始，完整加载仅验证任务可一次性快速比较源表和目标表中的所有行，立即报告任何故障，然后关闭。在此模式下针对速度进行了优化，验证永远不会因为失败而暂停。但是，对源端点或目标端点的更改会被报告为失败。

### Note

从 AWS DMS 3.4.6 及更高版本开始，此验证行为也适用于启用验证的完全加载迁移任务。

## CDC 仅验证

CDC 仅验证任务会在全新开始时，在源表和目标表之间验证所有现有行。此外，CDC 仅验证任务会持续运行，重新验证正在进行的复制更改，限制每次验证报告的失败次数，并在失败之前重试不匹配的行。它经过优化，可以防止误报。

如果超过 `FailureMaxCount` 或 `TableFailureMaxCount` 的阈值，则会暂停对表（或整个任务）的验证。这也适用于启用了验证的 CDC 或完全加载+CDC 迁移任务。而且，启用了验证的 CDC 任务会根据平均源和目标延迟，来延迟对每行更改的重新验证。

但是，CDC 仅验证任务不会迁移数据，也没有延迟。默认情况下，它将 `ValidationQueryCdcDelaySeconds` 设置为 180。而且，您可以增加容量以应对高延迟环境，并帮助防止误报。

## 仅验证的使用案例

在迁移或复制任务中，将数据验证部分拆分为单独的仅验证任务的使用案例包括但不限于以下内容：

- 精确控制何时进行验证 – 验证查询会给源端点和目标端点增加额外的负载。因此，先在一项任务中迁移或复制数据，然后在另一项任务中验证结果会有所帮助。
- 减少复制实例的负载 – 将数据验证拆分为在自己的实例上运行可能很有好处。

- 快速获取在给定时刻有多少行不匹配 – 例如在维护时段，进行生产切换到目标端点之前或期间，您可以创建一个完全加载仅验证任务来获取问题的答案。
- 当使用 CDC 组件的迁移任务预计会出现验证失败时 – 例如，如果将 Oracle varchar2 迁移到 PostgreSQL jsonb，CDC 验证会不断重试这些失败的行，并限制每次报告的失败数量。但是，您可以创建完全加载仅验证任务，并更快地获得答案。
- 您已经开发了一个数据修复脚本/实用程序，可以读取验证失败表 – (另请参阅[故障排除](#))。完全加载仅验证任务可以快速报告故障，以便数据修复脚本处理。

有关 JSON 文件中的 ValidationSettings 任务设置的示例，请参阅[任务设置示例](#)。

## 故障排除

在验证期间，AWS DMS 在目标终端节点中创建一个新表：awsdms\_control.awsdms\_validation\_failures\_v1。如果任何记录进入 ValidationSuspended 或 ValidationFailed 状态，则 AWS DMS 将诊断信息写入到 awsdms\_control.awsdms\_validation\_failures\_v1 中。您可以查询该表以帮助纠正验证错误。

有关在目标上更改所创建表的默认架构的信息，请参阅[控制表任务设置](#)。

以下是 awsdms\_control.awsdms\_validation\_failures\_v1 表描述：

列名称	数据类型	描述
TASK_NAME	VARCHAR(128) NOT NULL	AWS DMS 任务标识符。
TABLE_OWNER	VARCHAR(128) NOT NULL	表的架构 (所有者)。
TABLE_NAME	VARCHAR(128) NOT NULL	表名称。
FAILURE_TIME	DATETIME(3) NOT NULL	发生失败的时间。
KEY_TYPE	VARCHAR(128) NOT NULL	保留供将来使用 (值始终为“Row”)

列名称	数据类型	描述
KEY	TEXT NOT NULL	这是行记录类型的主键。
FAILURE_TYPE	VARCHAR(128) NOT NULL	验证错误的严重性。可以是 RECORD_DIFF、MISSING_SOURCE 或 MISSING_TARGET。
DETAILS	VARCHAR(8000) NOT NULL	与给定键不匹配的所有源/目标列值的 JSON 格式的字符串。

通过查询 `awsdms_control.awsdms_validation_failures_v1` 表，以下查询显示任务的所有失败。任务名称应该是任务的外部资源 ID。任务的外部资源 ID 是任务 ARN 中的最后一个值。例如，对于 ARN 值为 `arn:aws:dms:us-west-2:5599:task:VFPFKH4FJR3FTYKK2RYSI` 的任务，任务的外部资源 ID 为 `VFPFKH4FJR3FTYKK2RYSI`。

```
select * from awsdms_validation_failures_v1 where TASK_NAME = 'VFPFKH4FJR3FTYKK2RYSI'
```

TASK_NAME	VFPFKH4FJR3FTYKK2RYSI
TABLE_OWNER	DB2PERF
TABLE_NAME	PERFTEST
FAILURE_TIME	2020-06-11 21:58:44
KEY_TYPE	Row
KEY	{"key": ["3451491"]}
FAILURE_TYPE	RECORD_DIFF
DETAILS	[{"MYREAL": '+1.10106036e-01'}, {"MYREAL": '+1.10106044e-01'}],]

您可以查看 `DETAILS` 字段以确定哪些列不匹配。由于您已知失败记录的主键，您可以查询源和目标终端节点以查看记录的哪个部分不匹配。

## Redshift 验证性能

Amazon Redshift 在多个方面与关系数据库不同，包括列式存储、MPP、数据压缩和其他因素。这些差异使 Redshift 具有与关系数据库不同的性能特征。

在完全加载复制阶段，验证使用范围查询，数据大小由 PartitionSize 设置控制。这些基于范围的查询会选择源表中的所有记录。

对于正在进行的复制，查询会在基于范围的提取和单个记录提取之间切换。查询类型是根据多个因素动态确定的，例如：

- 查询量
- 源表上的 DML 查询类型
- 任务延迟
- 记录总数
- 验证设置，例如 PartitionSize

由于验证查询，您可能在 Amazon Redshift 集群上看到额外的负载。由于上述因素因用例而异，因此您必须检查验证查询性能，并相应地调整集群和表。一些缓解性能问题的方案包括：

- 减小 PartitionSize 和 ThreadCount 设置以帮助减少完全加载验证期间的工作量。请注意，这将减慢数据验证的速度。
- 虽然 Redshift 不强制使用主键，但 AWS DMS 依靠主键来唯一识别目标上的记录以进行数据验证。如果可能，请将主键设置为镜像排序键，以便更快地执行完全加载验证查询。

## 限制

- 数据验证要求表具有主键或唯一索引。
  - 主键列不能是 CLOB、BLOB 或 BYTE 类型。
  - 对于 VARCHAR 或 CHAR 类型的主键列，长度必须小于 1024。您必须在数据类型中指定长度。您不能使用无界数据类型作为数据验证的主键。
  - 使用 NOVALIDATE 子句创建的 Oracle 键不被视为主键或唯一索引。
  - 对于没有主键且只有唯一键的 Oracle 表，具有唯一约束条件的列也必须具有 NOT NULL 约束条件。
- 不支持验证 NULL PK/UK 值。
- 如果目标 PostgreSQL 实例中的主键列的排序规则未设置为“C”，则与 Oracle 中的排序顺序相比，PostgreSQL 的主键的排序顺序将不同。如果 PostgreSQL 和 Oracle 的排序顺序不同，数据验证将无法验证记录。



- 数据验证将针对源和目标数据库生成额外的查询。您必须确保两个数据库具有足够的资源以处理该额外负载。对于 Redshift 目标而言尤其如此。有关更多信息，请参阅下面的[Redshift 验证性能](#)。
- 将几个数据库合并为一个数据库时，不支持数据验证。
- 对于源或目标 Oracle 终端节点，AWS DMS 使用 DBMS\_CRYPTO 来验证 LOB。如果 Oracle 终端节点使用 LOB，则必须为用于访问 Oracle 终端节点的用户账户授予 dbms\_crypto 执行权限。您可以运行以下语句以执行该操作：

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

- 如果验证期间在 AWS DMS 外部修改了目标数据库，则可能不会准确报告差异。如果在 AWS DMS 对目标表执行验证期间，某个应用程序将数据写入到该表中，则可能会发生这种情况。
- 如果在验证期间持续修改一行或多行，则 AWS DMS 无法验证这些行。
- 如果 AWS DMS 检测到超过 10000 个失败或无法验证的记录，则会停止验证。在继续之前，先解决数据的任何根本问题。
- AWS DMS 不支持视图的数据验证。
- AWS DMS 在使用了字符替换任务设置时不支持数据验证。
- AWS DMS 不支持验证 Oracle LONG 类型。
- AWS DMS 不支持在异构迁移期间验证 Oracle 空间数据类型。

有关使用 S3 目标验证的限制，请参阅[使用 S3 目标验证的限制](#)。

## Amazon S3 目标数据验证

AWS DMS 支持验证 Amazon S3 目标中复制的数据。由于 AWS DMS 将复制的数据以平面文件形式存储在 Amazon S3 中，因此我们使用 [Amazon Athena](#) CREATE TABLE AS SELECT (CTAS) 查询来验证数据。

对存储在 Amazon S3 中的数据进行查询的计算量很大。因此，在更改数据捕获 (CDC) 期间，AWS DMS 每天仅在 UTC 午夜 (00:00) 对 Amazon S3 数据运行一次验证。AWS DMS 运行的每个每日验证都称为间隔验证。在间隔验证期间，AWS DMS 验证过去 24 小时内迁移到目标 Amazon S3 存储桶的所有更改记录。有关间隔验证限制的更多信息，请参阅[使用 S3 目标验证的限制](#)。

Amazon S3 目标验证使用 Amazon Athena，因此需要支付额外费用。有关更多信息，请参阅 [Amazon Athena 定价](#)。

**Note**

S3 目标验证需要 AWS DMS 版本 3.5.0 或更高版本。

**主题**

- [S3 目标验证先决条件](#)
- [使用 S3 目标验证的权限](#)
- [使用 S3 目标验证的限制](#)
- [通过 S3 目标验证使用仅验证任务](#)

## S3 目标验证先决条件

在使用 S3 目标验证之前，请检查以下设置和权限：

- 将端点的 [S3Settings](#) 的 DataFormat 值设置为 parquet。有关更多信息，请参见 [S3 的 Parquet 设置](#)。
- 确保向用于创建迁移任务的用户账户分配的角色具有以下一系列权限。参见下文中的[权限](#)。

对于使用持续复制 (CDC) 的任务，请检查以下设置：

- 启用补充日志，这样您就可以获得 CDC 数据的完整记录。有关启用补充日志记录的信息，请参阅本指南[故障排除和诊断支持](#)部分中的[自动将补充日志记录添加到 Oracle 源端点](#)。
- 为目标端点设置 TimestampColumnName 参数。对时间戳列名没有限制。有关更多信息，请参阅 [S3Settings](#)。
- 为目标设置基于日期的文件夹分区。有关更多信息，请参见 [使用基于日期的文件夹分区](#)。

## 使用 S3 目标验证的权限

要设置访问权限以使用 S3 目标验证，请确保向用于创建迁移任务的用户账户分配的角色具有以下一系列权限。请将示例值替换为您的值。

```
{
  "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>/*"
    ]
  }
]

```

```
    }  
  ]  
}
```

## 使用 S3 目标验证的限制

查看使用 S3 目标验证时适用的以下其他限制。有关适用于所有验证的限制，请参阅[限制](#)。

- 您的 DatePartitionSequence 值需要一个 Day 组件。S3 目标验证不支持 YYYYMM 格式。
- 在 CDC 期间运行间隔验证时，您可能在 `awsdms_validation_failures_v1` 表中看到虚假的验证错误。之所以出现这些错误，是因为 AWS DMS 会将间隔验证期间收到的更改迁移到第二天的分区文件夹中。通常，这些更改会写入当天的分区文件夹。这些虚假错误限制了对从动态源数据库复制到静态目标（例如 Amazon S3）的验证。要调查这些虚假错误，请检查验证时段（00:00 UTC）快要结束时的记录，错误通常出现在这些时候。

要尽可能减少虚假错误的数量，请确保任务的 `CDCLatencySource` 较低。有监控延迟的信息，请参阅[复制任务指标](#)。

- 处于 `failed` 或 `stopped` 状态的任务不验证前一天的更改。要最大限度地减少由于意外失败而导致的验证错误，请使用相同的表映射以及源端点和目标端点，创建单独的仅验证任务。有关仅验证任务的更多信息，请参阅[通过 S3 目标验证使用仅验证任务](#)。
- 表统计数据中的验证状态列反映了最近一次间隔验证的状态。因此，存在不匹配的表可能会在第二天的间隔验证后显示为已验证。检查目标 Amazon S3 存储桶中的 `s3_validation_failures` folder 是否存超过一天之前发生的不匹配情况。
- S3 验证使用 Amazon Athena 的存储桶表功能。这允许 S3 验证创建目标表数据的存储桶副本。这意味着表数据的副本被分成与 DMS 验证的内部分区相匹配的子集。Athena 存储桶牌桌的存储桶上限为 100,000 个。S3 验证尝试验证的任何超过此限制的表都将无法通过验证。S3 验证尝试创建的存储桶数量等于以下值：

```
(#records in the table) / (validation partition size setting)
```

要解决此限制，请增加验证分区大小设置，使 S3 验证创建的存储桶数量小于 100,000。有关存储桶的更多信息，请参阅 Amazon Athena [用户指南中的 Athena 中的分区和存储桶](#)。

## 通过 S3 目标验证使用仅验证任务

仅验证任务在不运行迁移的情况下对要迁移的数据进行验证。

即使迁移任务停止，仅验证任务也会继续运行，这样可以确保 AWS DMS 不会错过 00:00 UTC 间隔验证窗口。

在 Amazon S3 目标端点上使用仅验证任务有以下限制：

- 启用了仅验证设置的 Amazon S3 完全加载任务验证是受支持的，但操作方式与其他端点的完全加载和仅验证任务不同。对于作为目标的 S3，此类任务仅针对 S3 目标中的完全加载数据进行验证，不会针对作为 CDC 迁移的一部分迁移的任何数据进行验证。仅使用此功能来验证由仅完全加载任务创建的数据。使用此模式，对于正在运行 CDC 任务的目标，对其中的数据进行验证不会得到有效的验证。
- 仅验证任务仅验证自上次间隔验证窗口 (00:00 UTC) 以来的更改。仅验证任务不会验证前一天的完全加载数据或 CDC 数据。

## 在 AWS Database Migration Service 中标记资源

您可以在 AWS Database Migration Service (AWS DMS) 中使用标签向资源添加元数据。此外，这些标签可与 AWS Identity and Access Management (IAM) 策略结合使用，以管理对 AWS DMS 资源的访问并控制可将什么操作应用于 AWS DMS 资源。最后，您可以将具有类似标签的资源的费用分组在一起，使用这些标签来跟踪成本。

所有 AWS DMS 资源均可加标签：

- 证书
- 数据提供商
- 数据迁移
- 端点
- 事件订阅
- 实例配置文件
- 迁移项目
- 复制实例
- 复制子网组
- 复制任务

AWS DMS 标签是由您定义的名称-值对，与某种 AWS DMS 资源关联。此名称也叫密钥。为键提供值为可选操作。您可使用标签向 AWS DMS 资源分配任意信息。标签密钥可用于定义 (例如) 一个类别，而标签值可以是该类别中的一个项目。例如，可定义“project”标签键和“Salix”标签值，表示向 Salix 项目分配了 AWS DMS 资源。您也可使用标签通过 environment=test 或 environment=production 等键指定 AWS DMS 资源用于测试或生产。我们建议使用一组具有一致性的标签键，以便更轻松地跟踪与 AWS DMS 资源关联的元数据。

使用标签来组织 AWS 账单以反映您自身的成本结构。要执行此操作，请注册以获取包含标签键值的 AWS 账户账单。然后，如需查看组合资源的成本，请按有同样标签键值的资源组织您的账单信息。例如，您可以将特定的应用程序名称用作几个资源的标签，然后组织账单信息，以查看在数个服务中的使用该应用程序的总成本。有关更多信息，请参阅 AWS Billing 用户指南 中的 [使用成本分配标签](#)。

每个 AWS DMS 资源都有一组标签，其中包含分配给该 AWS DMS 资源的所有标签。一个标记集可以包含多达十个标记，也可以为空。如果向 AWS DMS 资源添加一个标签，而该标签的键与资源上某个现有的标签相同，则新值将覆盖旧值。

AWS 不会对您的标记应用任何语义意义；所有标记都会严格地作为字符串进行解析。AWS 根据您在创建资源时使用的设置，DMS 可能会在 AWS DMS 资源上设置标签。

下面列表描述了 AWS DMS 标签的特征。

- 标签密钥是标签的名称，属于必填内容。该字符串值的长度可以是 1 到 128 个 Unicode 字符，并且不能以“aws:”或“dms:”为前缀。字符串只能包含 Unicode 字母、数字、空格、“\_”、“.”、“/”、“=”、“+”、“-”的集合 (Java 正则表达式：“^( [\p{L}\p{Z}\p{N}\_.: / = + \- ]\*)\$”)。
- 标签值则是一个可选的标签字符串值。该字符串值的长度可以是 1 到 256 个 Unicode 字符，并且不能以“aws:”或“dms:”为前缀。字符串只能包含 Unicode 字母、数字、空格、“\_”、“.”、“/”、“=”、“+”、“-”的集合 (Java 正则表达式：“^( [\p{L}\p{Z}\p{N}\_.: / = + \- ]\*)\$”)。

在标签集中，值不必具有唯一性，且可为空值。例如，在“项目/Trinity”和“成本 - 中心/Trinity”的一个标签集中，可以存在键值对。

您可以使用 AWS CLI 或 AWS DMS API 添加、列出和删除 AWS DMS 资源上的标签。在使用 AWS CLI 或 AWS DMS API 时，必须提供要使用的 AWS DMS 资源的 Amazon 资源名称 (ARN)。有关构造 ARN 的详细信息，请参阅[为其构造亚马逊资源名称 \(ARN\) AWS DMS](#)。

注意，对标签进行缓存以用于授权。因此，如果添加和更新 AWS DMS 资源上的标签，可能需要几分钟后标签才可用。

## API

您可使用 AWS DMS API 为 AWS DMS 资源添加、列出或删除标签。

- 要向 AWS DMS 资源添加标签，请使用 [AddTagsToResource](#) 操作。
- 要列出分配给 AWS DMS 资源的标签，请使用 [ListTagsForResource](#) 操作。
- 要从 AWS DMS 资源删除标签，请使用 [RemoveTagsFromResource](#) 操作。

要了解有关如何构建所需 ARN 的更多信息，请参阅[为其构造亚马逊资源名称 \(ARN\) AWS DMS](#)。

在通过 AWS DMS API 使用 XML 时，标签会使用如下架构：

```
<Tagging>
```

```

<TagSet>
  <Tag>
    <Key>Project</Key>
    <Value>Trinity</Value>
  </Tag>
  <Tag>
    <Key>User</Key>
    <Value>Jones</Value>
  </Tag>
</TagSet>
</Tagging>

```

下表提供了允许使用的 XML 标签及其特征的列表。请注意，密钥和值的内容是区分大小写的。例如，project=Trinity 和 PROJECT=Trinity 是两个不同的标签。

标签元素	描述
标签集	标签集是分配给 Amazon RDS 资源的所有标签的容器。每个资源只能有一个标签集。您只可以通过 AWS DMS API 使用标签集。
标签	标签是用户定义的键值对。一个标签集中可以有 1 到 10 个标签。
键	<p>密钥是标签必需的名称。该字符串值的长度可以是 1 到 128 个 Unicode 字符，并且不能以“dms:”或“aws:”为前缀。字符串只能包含 Unicode 字母、数字、空格、“_”、“.”、“/”、“=”、“+”、“-”的集合 (Java 正则表达式：<code>"^([\p{L}\p{Z}\p{N}_.:/=+\-]*)\$"</code>)。</p> <p>密钥在标签集中必须具有唯一性。例如，标签集中不能有键相同但值不同的键-值对，如 project/Trinity 和 project/Xanadu。</p>
值	<p>值是标签的可选内容。该字符串值的长度可以是 1 到 256 个 Unicode 字符，并且不能以“dms:”或“aws:”为前缀。字符串只能包含 Unicode 字母、数字、空格、“_”、“.”、“/”、“=”、“+”、“-”的集合 (Java 正则表达式：<code>"^([\p{L}\p{Z}\p{N}_.:/=+\-]*)\$"</code>)。</p> <p>在标签集中，值不必具有唯一性，且可为空值。例如，在“项目/Trinity”和“成本 - 中心/Trinity”的一个标签集中，可以存在键值对。</p>



# 安全性 AWS Database Migration Service

云安全 AWS 是重中之重。作为 AWS 客户，您可以从专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构中受益。

安全是双方共同承担 AWS 的责任。[责任共担模式](#)将其描述为云的安全性 和云中的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。AWS 还为您提供可以安全使用的服务。作为 [AWS 合规性计划](#) 的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用的合规计划 AWS DMS，请参阅[按合规计划划分的范围内的 AWS 服务](#)。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您组织的要求以及适用的法律法规。

本文档可帮助您了解在使用时如何应用分担责任模型 AWS DMS。以下主题向您介绍如何进行配置 AWS DMS 以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS DMS 资源。

您可以管理对您的 AWS DMS 资源和数据库 (DB) 的访问权限。用于管理访问权限的方法取决于您需要使用以下命令执行的复制任务 AWS DMS：

- 使用 AWS Identity and Access Management (IAM) 策略分配权限，以确定谁有权管理 AWS DMS 资源。AWS DMS 如果您以 IAM 用户身份登录，则要求您拥有相应的权限。例如，您可以使用 IAM 确定哪些人可以创建、描述、修改和删除数据库实例和集群，为资源添加标签或修改安全组。有关 IAM 及其与一起使用的更多信息 AWS DMS，请参阅[的身份和访问管理 AWS Database Migration Service](#)。
- AWS DMS 使用安全套接字层 (SSL) 与传输层安全 (TLS) 进行端点连接。有关将 SSL/TLS 与配合 AWS DMS 使用的更多信息，请参阅[将 SSL 与 AWS Database Migration Service](#)
- AWS DMS 使用 AWS Key Management Service (AWS KMS) 加密密钥来加密您的复制实例使用的存储空间及其终端节点连接信息。AWS DMS 还使用 AWS KMS 加密密钥保护 Amazon S3 和 Amazon Redshift 目标终端节点的静态目标数据。有关更多信息，请参阅[设置加密密钥并指定 AWS KMS 权限](#)。
- AWS DMS 始终在基于 Amazon VPC 服务的虚拟私有云 (VPC) 中创建您的复制实例，以实现尽可能大的网络访问控制。对于数据库实例和实例集群，请使用与复制实例相同的 VPC，或使用其他 VPC 来匹配此级别的访问控制。使用的每个 Amazon VPC 必须与安全组关联，该安全组应具有规则，允许所有端口上的所有流量离开（传出）VPC。使用此方法，只要这些终端节点上启用了正确的传入，就允许从复制实例与您的源和目标数据库终端节点通信。

有关可用网络配置的更多信息 AWS DMS，请参阅[为复制实例设置网络](#)。有关在 VPC 中创建数据库实例或实例集群的更多信息，请参阅 [AWS 文档](#) 中的 Amazon 数据库安全和集群管理文档。有关 AWS DMS 支持的网络配置的更多信息，请参阅 [为复制实例设置网络](#)。

- 要查看数据库迁移日志，您需要对所使用的 IAM 角色拥有相应的 Amazon L CloudWatch logs 权限。有关 AWS DMS 日志记录的更多信息，请参阅 [使用 Amazon CloudWatch 监控复制任务](#)。

## 主题

- [中的数据保护 AWS Database Migration Service](#)
- [的身份和访问管理 AWS Database Migration Service](#)
- [AWS Database Migration Service 的合规性验证](#)
- [AWS Database Migration Service 中的故障恢复能力](#)
- [AWS Database Migration Service 中的基础设施安全性](#)
- [使用资源名称和标签进行精细访问控制](#)
- [设置加密密钥并指定 AWS KMS 权限](#)
- [的网络安全 AWS Database Migration Service](#)
- [将 SSL 与 AWS Database Migration Service](#)
- [更改数据库密码](#)

# 中的数据保护 AWS Database Migration Service

## 数据加密

您可以为支持的 AWS DMS 目标端点的数据资源启用加密。AWS DMS 还会加密与其所有源端点 AWS DMS 和目标端点之间的连接。AWS DMS 此外，您还可以管理用于启用此加密的密钥 AWS DMS 及其支持的目标端点。

### 主题

- [静态加密](#)
- [传输中加密](#)
- [密钥管理](#)

## 静态加密

AWS DMS 支持静态加密，允许您指定在复制的数据复制到支持的 AWS DMS 目标终端节点之前将其推送到 Amazon S3 时要使用的服务器端加密模式。您可以通过设置终端节点的 `encryptionMode` 额外连接属性来指定此加密模式。如果此 `encryptionMode` 设置指定 KMS 密钥加密模式，则还可以创建专门用于加密以下目标终端节点的 AWS DMS 目标数据的自定义 AWS KMS 密钥：

- Amazon Redshift – 有关设置 `encryptionMode` 更多信息，请参阅 [使用 Amazon Redshift 作为 AWS DMS 的目标时的端点设置](#)。有关创建自定义 AWS KMS 加密密钥的更多信息，请参阅 [创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密](#)。
- Amazon S3 – 有关设置 `encryptionMode` 更多信息，请参阅 [使用 Amazon S3 作为 AWS DMS 的目标时的端点设置](#)。有关创建自定义 AWS KMS 加密密钥的更多信息，请参阅 [创建 AWS KMS 密钥以加密 Amazon S3 目标对象](#)。

## 传输中加密

AWS DMS 通过确保其复制的数据从源端点安全地移动到目标端点，支持传输中的加密。这包括对复制实例上的 S3 存储桶进行加密，在复制管道中传输数据时，复制任务将 S3 存储桶用作中间存储。要对源端点和目标端点的任务连接进行加密，请 AWS DMS 使用安全套接字层 (SSL) 或传输层安全 (TLS)。通过对两个端点的连接进行加密，AWS DMS 可以确保您的数据在从源终端节点移动到复制任务以及从任务移动到目标终端节点时是安全的。有关将 SSL/TLS 与配合使用的更多信息，请参阅 [AWS DMS 将 SSL 与 AWS Database Migration Service](#)

AWS DMS 支持默认密钥和自定义密钥，用于加密中间复制存储和连接信息。您可以使用 AWS KMS 来管理这些密钥。有关更多信息，请参阅 [设置加密密钥并指定 AWS KMS 权限](#)。

## 密钥管理

AWS DMS 支持默认密钥或自定义密钥来加密某些目标端点的复制存储、连接信息和目标数据存储。您可以通过使用来管理这些密钥 AWS KMS。有关更多信息，请参阅 [设置加密密钥并指定 AWS KMS 权限](#)。

## 互连网络流量隐私保护

无论是在本地运行还是作为云端 AWS 服务的一部分运行，同一 AWS 区域的源端点和目标端点之间的 AWS DMS 连接都受到保护。（至少一个端点（源端点或目标端点）必须作为云端 AWS 服务的一部分运行。）无论这些组件共享相同的虚拟私有云 (VPC) 还是存在于不同的 VPC 中，如果这些 VPC 都位于同一个 AWS 区域，则此保护都适用。有关支持的网络配置的更多信息 AWS DMS，请参阅[为复制实例设置网络](#)。有关使用这些网络配置时的安全注意事项的更多信息，请参阅 [网络安全 AWS Database Migration Service](#)。

## DMS Fleet Advisor 中的数据保护

DMS Fleet Advisor 收集并分析您的数据库元数据，以确定迁移目标的正确大小。DMS Fleet Advisor 不会访问您的表中的数据，也不会传输这些数据。此外，DMS Fleet Advisor 不会跟踪数据库功能的使用情况，也不会访问您的使用情况统计信息。

在您创建数据库用户以便让 DMS Fleet Advisor 用来处理数据库时，您可以控制对数据库的访问权限。您可以向这些用户授予所需的权限。要使用 DMS Fleet Advisor，您需要向数据库用户授予读取权限。DMS Fleet Advisor 不会修改您的数据库，也不需要写入权限。有关更多信息，请参阅 [为 AWS DMS Fleet Advisor 创建数据库用户](#)。

您可以在数据库中使用数据加密。AWS DMS 还会加密 DMS Fleet Advisor 内部及其数据收集器内部的连接。

DMS 数据收集器使用数据保护应用程序编程接口 ( DPAPI ) 来加密、保护和存储有关客户环境和数据库凭证的信息。在 DMS 数据收集器工作的服务器上，DMS Fleet Advisor 将这些加密数据存储在文件中。DMS Fleet Advisor 不会从这台服务器传输这些数据。有关 DPAPI 的更多信息，请参阅[如何：使用数据保护](#)。

安装 DMS 数据收集器后，您可以查看此应用程序为收集指标而运行的所有查询。您可以在离线模式下运行 DMS 数据收集器，然后在您的服务器上查看收集的数据。此外，您还可以在您的 Amazon S3 存储桶中查看这些收集的数据。有关更多信息，请参阅 [DMS 数据收集器的工作方式](#)。

# 的身份和访问管理 AWS Database Migration Service

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证 ( 登录 ) 和授权 ( 拥有权限 ) 使用 AWS DMS 资源。您可以使用 IAM AWS 服务 , 无需支付额外费用。

## 主题

- [受众](#)
- [使用身份进行身份验证](#)
- [使用策略管理访问](#)
- [如何 AWS Database Migration Service 与 IAM 配合使用](#)
- [AWS Database Migration Service 基于身份的策略示例](#)
- [基于资源的策略示例 AWS KMS](#)
- [使用密钥访问 AWS Database Migration Service 端点](#)
- [将服务相关角色用于 AWS DMS](#)
- [对 AWS Database Migration Service 身份和访问进行故障排除](#)
- [使用 AWS DMS所需的 IAM 权限](#)
- [创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)
- [防止跨服务混淆座席](#)
- [AWS 的托管策略 AWS Database Migration Service](#)

## 受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您所做的工作 AWS DMS。

服务用户-如果您使用 AWS DMS 服务完成工作，则管理员会为您提供所需的凭证和权限。当您使用更多 AWS DMS 功能来完成工作时，您可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS DMS中的特征，请参阅 [对 AWS Database Migration Service 身份和访问进行故障排除](#)。

服务管理员-如果您负责公司的 AWS DMS 资源，则可能拥有完全访问权限 AWS DMS。您的工作是确定您的服务用户应访问哪些 AWS DMS 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务

用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解您的公司如何将 IAM 与配合使用 AWS DMS，请参阅[如何 AWS Database Migration Service 与 IAM 配合使用](#)。

**IAM 管理员：**如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 AWS DMS 的访问权限的详细信息。要查看您可以在 IAM 中使用的 AWS DMS 基于身份的策略示例，请参阅。[AWS Database Migration Service 基于身份的策略示例](#)

## 使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担任 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。AWS IAM Identity Center（IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的[如何登录到您 AWS 账户的](#)。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的[签署 AWS API 请求](#)。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如，AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[多重身份验证](#)和《IAM 用户指南》中的[在 AWS 中使用多重身份验证 \(MFA\)](#)。

## AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关需要您以根用户身份登录的任务的完整列表，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

## IAM 用户和群组

[IAM 用户](#)是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定

的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 [对于需要长期凭证的使用场景定期轮换访问密钥](#)。

[IAM 组](#)是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的[何时创建 IAM 用户（而不是角色）](#)。

## IAM 角色

[IAM 角色](#)是您内部具有特定权限 AWS 账户的身份。它类似于 IAM 用户，但与特定人员不关联。您可以使用 AWS Management Console 通过[切换角色在中临时担任 IAM 角色](#)。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的[使用 IAM 角色](#)。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的[为第三方身份提供商创建角色](#)。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的[权限集](#)。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解角色和基于资源的跨账户访问策略之间的区别，请参阅 [IAM 用户指南中的跨账户资源访问](#)。
- 跨服务访问 — 有些 AWS 服务使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service (Amazon S3) 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
- 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他

AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。

- 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。
- 服务相关角色-服务相关角色是一种链接到的服务角色。AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的[使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

要了解是使用 IAM 角色还是 IAM 用户，请参阅 IAM 用户指南中的[何时创建 IAM 角色 \(而不是用户\)](#)。

## 使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS，当与身份或资源关联时，它会定义其权限。AWS 在委托人 (用户、root 用户或角色会话) 发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的[JSON 策略概览](#)。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM policy，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 iam:GetRole 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。



## 基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[创建 IAM policy](#)。

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅 IAM 用户指南中的[在托管式策略与内联策略之间进行选择](#)。

## 基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

## 访问控制列表 (ACL)

访问控制列表 (ACL) 控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。AWS WAF 要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的[访问控制列表 \(ACL\) 概览](#)。

## 其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- **权限边界**：权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 Principal 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- **服务控制策略 (SCP)**-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管

理的服务。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体（包括每个 AWS 账户根用户实体）的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的 [SCP 的工作原理](#)。

- 会话策略 – 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的 [会话策略](#)。

## 多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的 [策略评估逻辑](#)。

## 如何 AWS Database Migration Service 与 IAM 配合使用

在使用 IAM 管理访问权限之前 AWS DMS，您应该了解哪些可用的 IAM 功能 AWS DMS。要全面了解如何 AWS DMS 和其他 AWS 服务与 IAM 配合使用，请参阅 IAM 用户指南中的与 IAM [配合使用的 AWS 服务](#)。

### 主题

- [AWS DMS 基于身份的策略](#)
- [AWS DMS 基于资源的策略](#)
- [基于 AWS DMS 标签的授权](#)
- [的 IAM 角色适用于 AWS DMS](#)
- [适用于 DMS Fleet Advisor 的身份和访问管理](#)

## AWS DMS 基于身份的策略

使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源，并指定在什么条件下允许或拒绝操作。AWS DMS 支持特定操作、资源和条件键。要了解在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素参考](#)。

### 操作

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

正在执行的策略操作在操作前 AWS DMS 使用以下前缀:dms:. 例如，要授予某人使用 AWS DMS CreateReplicationTask API 操作创建复制任务的权限，您需要将该dms:CreateReplicationTask操作包含在他们的策略中。策略声明必须包含Action或NotAction元素。AWS DMS 定义了自己的一组操作，这些操作描述了您可以使用此服务执行的任务。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示。

```
"Action": [
    "dms:action1",
    "dms:action2"
```

您也可以使用通配符 (\*) 指定多个操作。例如，要指定以单词 Describe 开头的所有操作，请包括以下操作。

```
"Action": "dms:Describe*"
```

要查看 AWS DMS 操作列表，请参阅 IAM 用户指南 AWS Database Migration Service 中的[定义操作](#)。

## 资源

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN\)](#) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (\*) 指示语句应用于所有资源。

```
"Resource": "*" 
```

AWS DMS 适用于以下资源：

- 证书
- 端点
- 事件订阅
- 复制实例
- 复制子网 (安全) 组
- 复制任务

所需的一个或多个 AWS DMS 资源取决于您调用的一个或多个操作。您需要一个允许对资源 ARN 指定的关联资源执行这些操作的策略。

例如，AWS DMS 终端节点资源具有以下 ARN：

```
arn:${Partition}:dms:${Region}:${Account}:endpoint/${InstanceId}
```

有关 ARN 格式的更多信息，请参阅 [Amazon 资源名称 \(ARN\) 和 AWS 服务命名空间](#)。

例如，要在语句中指定 us-east-2 区域的 1A2B3C4D5E6F7G8H9I0J1K2L3M 端点实例，请使用以下 ARN。

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/1A2B3C4D5E6F7G8H9I0J1K2L3M"
```

要指定特定账户包含的所有端点，请使用通配符 ( \* )。

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/*"
```

某些 AWS DMS 操作 (例如创建资源的操作) 无法对特定资源执行。在这些情况下，您必须使用通配符 ( \* )。

```
"Resource": "*"
```

某些 AWS DMS API 操作涉及多个资源。例如，StartReplicationTask 启动复制任务并将其连接到两个数据库端点资源 (一个源和一个目标)，因此 IAM 用户必须具有读取源端点和写入目标端点的权限。要在单个语句中指定多个资源，请使用逗号分隔 ARN。

```
"Resource": [
    "resource1",
    "resource2" ]
```

有关使用策略控制 AWS DMS 资源访问权限的更多信息，请参阅[使用资源名称控制访问](#)。要查看 AWS DMS 资源类型及其 ARN 的列表，请参阅《IAM 用户指南》中的[AWS Database Migration Service 定义的资源](#)。要了解您可以在哪些操作中指定每个资源的 ARN，请参阅[AWS Database Migration Service 定义的操作](#)。

## 条件键

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素 ( 或 Condition 块 ) 中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用[条件运算符](#) ( 例如，等于或小于 ) 的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑 OR 运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM policy 元素：变量和标签](#)。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

AWS DMS 定义自己的条件键集，还支持使用一些全局条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

AWS DMS 定义了一组可以在其条件键中使用的标准标签，还允许您定义自己的自定义标签。有关更多信息，请参阅[使用标签控制访问](#)。

要查看 AWS DMS 条件键列表，请参阅 IAM 用户指南 AWS Database Migration Service 中的[条件密钥](#)。要了解可以对哪些操作和资源使用条件键，请参阅[AWS Database Migration Service 定义的操作](#)和[AWS Database Migration Service 定义的资源](#)。

## 示例

要查看 AWS DMS 基于身份的策略的示例，请参阅 [AWS Database Migration Service 基于身份的策略示例](#)

## AWS DMS 基于资源的策略

基于资源的策略是 JSON 策略文档，用于指定委托人可以对给定 AWS DMS 资源执行哪些操作以及在什么条件下可以执行哪些操作。AWS DMS 支持针对您创建的 AWS KMS 加密密钥的基于资源的权限策略，这些密钥是为了加密迁移到支持的目标端点的数据而创建的。支持的目标端点包括 Amazon Redshift 和 Amazon S3。通过使用基于资源的策略，您可以向每个目标端点的其他账户授予使用这些加密密钥的权限。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为 [基于资源的策略中的委托人](#)。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源位于不同的 AWS 账户中时，您还必须向委托人实体授予访问资源的权限。通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。

该 AWS DMS 服务仅支持一种基于资源的策略，即密钥策略，该策略附加到 AWS KMS 加密密钥。此策略定义哪些委托人实体（账户、用户、角色和联合身份用户）可以对支持的目标端点上的迁移数据进行加密。

要了解如何将基于资源的策略附加到为支持的目标端点创建的加密密钥，请参阅 [创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密](#) 和 [创建 AWS KMS 密钥以加密 Amazon S3 目标对象](#)。

## 示例

有关 AWS DMS 基于资源的策略的示例，请参阅 [基于资源的策略示例 AWS KMS](#)。

## 基于 AWS DMS 标签的授权

您可以为 AWS DMS 资源附加标签或在请求中传递标签 AWS DMS。要根据标签控制访问权限，您可以使用 `dms:ResourceTag/key-name`、`aws:RequestTag/key-name`、或 [条件键在策略的 `aws:TagKeys` 条件元素](#) 中提供标签信息。AWS DMS 定义了一组可以在其条件键中使用的标准标签，还允许您定义自己的自定义标签。有关更多信息，请参阅 [使用标签控制访问](#)。

有关基于标签限制资源访问权的基于身份的策略示例，请参阅 [根据标签访问 AWS DMS 资源](#)。

## 的 IAM 角色适用于 AWS DMS

I [IAM 角色](#) 是您的 AWS 账户中具有特定权限的实体。

### 将临时证书与 AWS DMS

您可以使用临时凭证进行联合身份登录、代入 IAM 角色或代入跨账户角色。您可以通过调用 AWS STS API 操作（例如 [AssumeRole](#) 或 [GetFederation令牌](#)）来获得临时安全证书。

AWS DMS 支持使用临时证书。

### 服务相关角色

[服务相关角色](#) 允许 AWS 服务访问其他服务中的资源以代表您完成操作。服务相关角色显示在 IAM 账户中，并归该服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理 AWS DMS 服务相关角色的详细信息，请参阅 [使用服务相关角色](#)。

### 服务角色

此功能允许服务代表您担任 [服务角色](#)。此角色允许服务访问其他服务中的资源以代表您完成操作。服务角色显示在 IAM 账户中，并归该账户所有。这意味着，IAM 管理员可以更改该角色的权限。但是，这样做可能会中断服务的功能。

AWS DMS 支持两种类型的服务角色，您必须创建这些角色才能使用某些源端点或目标端点：

- 具有允许 AWS DMS 访问以下源端点和目标端点（或其资源）权限的角色：
  - Amazon DynamoDB（作为目标）– 有关更多信息，请参阅 [将 DynamoDB 作为 AWS Database Migration Service 目标时的先决条件](#)。
  - OpenSearch 作为目标 — 有关更多信息，请参阅 [使用 Amazon OpenSearch Service 作为 AWS Database Migration Service 的目标的先决条件](#)。
  - Amazon Kinesis（作为目标）– 有关更多信息，请参阅 [使用 Kinesis 数据流作为目标的先决条件 AWS Database Migration Service](#)。
  - Amazon Redshift（作为目标）– 仅当创建自定义 KMS 加密密钥以便加密目标数据，或者指定自定义 S3 存储桶以便保留中间任务存储时，才需要创建指定角色。有关更多信息，请参阅 [创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密](#) 或 [Amazon S3 存储桶设置](#)。
  - Amazon S3（作为源或目标）– 有关更多信息，请参阅 [使用 Amazon S3 作为来源的先决条件 AWS DMS](#) 或 [将 Amazon S3 作为目标的先决条件](#)。

例如，要从 S3 源端点读取数据或将数据推送到 S3 目标端点，您必须创建服务角色作为针对其中的每个端点操作访问 S3 的先决条件。

- 需要权限才能使用 AWS CLI 和 AWS DMS API 的角色 — 您需要创建的两个 IAM 角色是 `dms-vpc-role` 和 `dms-cloudwatch-logs-role`。如果您使用 Amazon Redshift 作为目标数据库，则还必须创建 IAM 角色并将其 `dms-access-for-endpoint` 添加到您的 AWS 账户。有关更多信息，请参阅 [创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

## 在中选择 IAM 角色 AWS DMS

如果您使用 AWS CLI 或 AWS DMS API 进行数据库迁移，则必须先向 AWS 账户添加某些 IAM 角色，然后才能使用 AWS DMS 的功能。其中两个角色为 `dms-vpc-role` 和 `dms-cloudwatch-logs-role`。如果您使用 Amazon Redshift 作为目标数据库，则还必须将 IAM 角色 `dms-access-for-endpoint` 添加到您的 AWS 账户。有关更多信息，请参阅 [创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

## 适用于 DMS Fleet Advisor 的身份和访问管理

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源，以及允许或拒绝操作的条件。DMS Fleet Advisor 支持特定的操作、资源和条件键。要了解在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素参考](#)。

DMS Fleet Advisor 使用 IAM 角色访问 Amazon Simple Storage Service。IAM 角色是您的 AWS 账户中具有特定权限的实体。有关更多信息，请参阅 [创建 IAM 资源](#)。

## AWS Database Migration Service 基于身份的策略示例

默认情况下，IAM 用户和角色没有创建或修改 AWS DMS 资源的权限。他们也无法使用 AWS Management Console AWS CLI、或 AWS API 执行任务。IAM 管理员必须创建 IAM 策略，以便为用户和角色授予权限以对所需的指定资源执行特定的 API 操作。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户或组。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅《IAM 用户指南》中的 [在 JSON 选项卡上创建策略](#)。

### 主题

- [策略最佳实践](#)
- [使用 AWS DMS 控制台](#)
- [允许用户查看他们自己的权限](#)
- [访问一个 Amazon S3 存储桶](#)
- [根据标签访问 AWS DMS 资源](#)



## 策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS DMS 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#) 或 [工作职能的 AWS 托管策略](#)。
- 应用最低权限 – 在使用 IAM policy 设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的策略和权限](#)。
- 使用 IAM policy 中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM policy 语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM Access Analyzer 策略验证](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 [配置受 MFA 保护的 API 访问](#)。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实操](#)。

## 使用 AWS DMS 控制台

以下策略允许您访问 DMS ( 包括 AWS AWS DMS 控制台 )，还规定了其他亚马逊服务 ( 例如 Amazon EC2 ) 所需的某些操作的权限。

```
{
  "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"
  }
]
}

```

细分这些权限可以帮助您更好地了解使用控制台需要每个权限的原因。

需要使用以下部分以允许用户列出可用的 AWS KMS 密钥和别名，以便在控制台中显示。如果您知道 KMS 密钥的 Amazon 资源名称 ( ARN ) 并且仅使用 AWS Command Line Interface ( AWS CLI ) ，则不需要此条目。

```

{
  "Effect": "Allow",
  "Action": [
    "kms:ListAliases",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

对于需要在端点中传入角色 ARN 的某些端点类型，需要使用以下部分。此外，如果未提前创建所需的 AWS DMS 角色，则 AWS DMS 控制台可以创建角色。如果提前配置了所有角色，这是 `iam:GetRole` 和 `iam:PassRole` 中所需的所有角色。有关角色的更多信息，请参阅[创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

```

{
  "Effect": "Allow",
  "Action": [
    "iam:GetRole",
    "iam:PassRole",
    "iam:CreateRole",
    "iam:AttachRolePolicy"
  ],
  "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

```
}

```

以下部分是必填的，因为 AWS DMS 需要创建 Amazon EC2 实例并为创建的复制实例配置网络。这些资源位于客户的账户中，因此，需要能够代表客户执行这些操作。

```
{
    "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"
}
```

需要使用以下部分，以允许用户查看复制实例指标。

```
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:Get*",
        "cloudwatch:List*"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

需要使用该部分，以允许用户查看复制日志。

```
{
    "Effect": "Allow",
    "Action": [
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:FilterLogEvents",
        "logs:GetLogEvents"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}
```

```
}
```

AWS DMS 控制台会创建多个角色，当您使用 AWS DMS 控制台时，这些角色会自动附加到您的 AWS 账户。如果您使用 AWS Command Line Interface (AWS CLI) 或 AWS DMS API 进行迁移，则需要将这些角色添加到您的账户中。有关添加这些角色的更多信息，请参阅[创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

有关使用此策略访问 AWS DMS 的要求的更多信息，请参阅[使用 AWS DMS 所需的 IAM 权限](#)。

## 允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
  "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": "*"
    }
  ]
}
```

```
    }  
  ]  
}
```

## 访问一个 Amazon S3 存储桶

AWS DMS 使用 Amazon S3 存储桶作为数据库迁移的中间存储。通常，AWS DMS 会为此目的管理默认 S3 存储桶。但是，在某些情况下，尤其是在您使用 AWS CLI 或 AWS DMS API 时，AWS DMS 允许您改为指定自己的 S3 存储桶。例如，您可以指定自己的 S3 存储桶，以将数据迁移到 Amazon Redshift 目标端点。在这种情况下，您需要创建一个具有基于 AWS-managed AmazonDMSRedshiftS3Role 策略的权限的角色。

以下示例显示了 AmazonDMSRedshiftS3Role 策略的一个版本。它允许 AWS DMS 授予您 AWS 账户中的 IAM 用户访问您的 Amazon S3 存储桶的权限。它还允许用户添加、更新和删除对象。

除了授予该用户 `s3:PutObject`、`s3:GetObject` 和 `s3:DeleteObject` 权限外，此策略还授予 `s3:ListAllMyBuckets`、`s3:GetBucketLocation` 和 `s3:ListBucket` 权限。这些是控制台所需的其他权限。其他权限允许 AWS DMS 管理存储桶的生命周期。此外，需要 `s3:GetObjectAcl` 操作才能复制对象。

```
{  
  "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-*"
  }
]
}
```

有关基于此策略创建角色的更多信息，请参阅 [Amazon S3 存储桶设置](#)。

## 根据标签访问 AWS DMS 资源

您可以在基于身份的策略中使用条件，以便基于标签控制对 AWS DMS 资源的访问。此示例说明如何创建允许访问所有 AWS DMS 端点的策略。但是，仅当端点数据库标签 `Owner` 具有该用户的用户名值时，才会授予相应权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "dms:*",
      "Resource": "arn:aws:dms:*:*:endpoint/*",
      "Condition": {
        "StringEquals": {"dms:endpoint-tag/Owner": "${aws:username}"}
      }
    }
  ]
}
```

您可以将该策略附加到您账户中的 IAM 用户。如果名为的用户 `richard-roe` 尝试访问 AWS DMS 终端节点，则必须标记终端节点数据库 `Owner=richard-roe` 或 `owner=richard-roe`。否则，此用户将被拒绝访问。条件标签键 `Owner` 匹配 `Owner` 和 `owner`，因为条件键名称不区分大小写。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。

## 基于资源的策略示例 AWS KMS

AWS DMS 允许您创建自定义 AWS KMS 加密密钥来加密支持的目标端点数据。要了解如何创建密钥策略并将其附加到为支持的目标数据加密而创建的加密密钥，请参阅 [创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密](#) 和 [创建 AWS KMS 密钥以加密 Amazon S3 目标对象](#)。

### 主题

- [用于加密亚马逊 Redshift 目标数据的自定义 AWS KMS 加密密钥的策略](#)

- [用于加密 Amazon S3 目标数据的自定义 AWS KMS 加密密钥的策略](#)

## 用于加密亚马逊 Redshift 目标数据的自定义 AWS KMS 加密密钥的策略

以下示例显示了为 AWS KMS 加密密钥创建的密钥政策的 JSON，该加密密钥是为加密 Amazon Redshift 目标数据而创建的。

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/Admin"
        ]
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms>Delete*",
        "kms:TagResource",
        "kms:UntagResource",

```



```
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
    ],
    "Resource": "*"
},
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::987654321098:role/DMS-Redshift-endpoint-access-role"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
{
    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::987654321098:role/DMS-Redshift-endpoint-access-role"
        ]
    },
    "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": true
        }
    }
}
]
```

```
}
```

在此示例中，您可以看到密钥政策引用了角色，以访问在创建密钥之前创建的 Amazon Redshift 目标端点数据。在该示例中，该角色为 `DMS-Redshift-endpoint-access-role`。您还可以查看不同委托人（用户和角色）所允许的不同密钥操作。例如，任何具有 `DMS-Redshift-endpoint-access-role` 的用户都可以对目标数据进行加密、解密和重新加密。这样的用户还可以生成数据密钥以供导出，以加密外部的数据 AWS KMS。它们还可以返回有关 AWS KMS 密钥的详细信息，例如您刚刚创建的密钥。此外，此类用户还可以管理 AWS 资源的附件，如目标端点。

## 用于加密 Amazon S3 目标数据的自定义 AWS KMS 加密密钥的策略

以下示例显示了为 AWS KMS 加密密钥创建的密钥政策的 JSON，该加密密钥是为加密 Amazon S3 目标数据而创建的。

```
{
  "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
  }
}
]
```

在此示例中，您可以看到密钥政策引用了角色，以访问在创建密钥之前创建的 Amazon S3 目标端点数据。在该示例中，该角色为 `DMS-S3-endpoint-access-role`。您还可以查看不同委托人（用户和角色）所允许的不同密钥操作。例如，任何具有 `DMS-S3-endpoint-access-role` 的用户都可以对目标数据进行加密、解密和重新加密。这样的用户还可以生成数据密钥以供导出，以加密外部的数据 AWS KMS。它们还可以返回有关 AWS KMS 密钥的详细信息，例如您刚刚创建的密钥。此外，此类用户还可以管理 AWS 资源的附件，如目标端点。

## 使用密钥访问 AWS Database Migration Service 端点

对于 AWS DMS，密钥是一种加密密钥，您可以使用它来表示一组用户凭据，通过秘密身份验证对支持的 AWS DMS 源端点或目标端点的数据库连接进行身份验证。对于同时使用 Oracle 自动存储管理 (ASM) 的 Oracle 终端节点，AWS DMS 需要一个代表访问 Oracle ASM 的用户凭据的额外密钥。

您可以使用一项服务来创建 AWS DMS 需要进行秘密身份验证的一个或多个机密 AWS Secrets Manager，该服务用于安全地创建、存储和检索凭据，以访问云端和内部部署中的应用程序、服务和 IT 资源。包括支持在无需干预的情况下自动定期轮换加密密钥值，从而为凭证提供额外的安全级别。在中启用机密值轮换 AWS Secrets Manager 还可以确保这种密钥值轮换不会对依赖该密钥的任何数据库迁移产生任何影响。要对端点数据库连接进行秘密身份验证，请创建一个密钥，将其身份或 ARN 分配给 `SecretsManagerSecretId`，并将其包含在端点设置中。要对作为 Oracle 端点一部分的 Oracle ASM 进行秘密身份验证，请创建一个密钥，将其身份或 ARN 分配给 `SecretsManagerOracleAsmSecretId`，并将其包含在端点设置中。

### Note

不能使用由 Amazon RDS Aurora 管理的主凭证。这些凭证不包括建立连接 AWS DMS 所需的主机或端口信息。而是应创建新用户和密钥。有关创建用户和密钥的信息，请参阅后面的[使用创建 AWS Management Console 机密和秘密访问角色](#)。

有关的更多信息 AWS Secrets Manager，请参阅[什么是 S AWS secrets Manager？](#) 在《AWS Secrets Manager 用户指南》中。

AWS DMS 支持在支持的源端点和目标端点上对以下本地或 AWS 托管数据库进行秘密身份验证：

- Amazon DocumentDB
- IBM Db2 LUW
- Microsoft SQL Server
- MongoDB
- MySQL
- Oracle
- PostgreSQL
- Amazon Redshift
- SAP ASE

要连接到这些数据库中的任何一个，您可以选择在端点设置中输入以下值中的一组，但不能同时输入两组值：

- 明文值，以使用 `UserName`、`Password`、`ServerName` 和 `Port` 设置对数据库连接进行身份验证。对于同时使用 Oracle ASM 的 Oracle 端点，还包括其他明文值，以使用 `AsmUserName`、`AsmPassword` 和 `AsmServerName` 设置对 ASM 进行身份验证。
- 使用 `SecretsManagerSecretId` 和 `SecretsManagerAccessRoleArn` 设置的值进行秘密身份验证。对于使用 Oracle ASM 的 Oracle 端点，还包括 `SecretsManagerOracleAsmSecretId` 和 `SecretsManagerOracleAsmAccessRoleArn` 设置的值。这些设置的密钥值可能包括以下参数：
  - `SecretsManagerSecretId` – 您在 AWS Secrets Manager 中为访问端点数据库而创建的密钥的完整 Amazon 资源名称 (ARN)、部分 ARN 或友好名称。
  - `SecretsManagerAccessRoleArn`— 您在 IAM 中创建的用于代表您访问此密钥的 `SecretsManagerSecretId` 秘密 AWS DMS 访问角色的 ARN。
  - `SecretsManagerOracleAsmSecretId` – 您在 AWS Secrets Manager 中为访问 Oracle ASM 而创建的密钥的完整 Amazon 资源名称 (ARN)、部分 ARN 或友好名称。
  - `SecretsManagerOracleAsmAccessRoleArn` – 您在 IAM 中为方便 AWS DMS 代表您访问 `SecretsManagerOracleAsmSecretId` 密钥而创建的 secret 访问角色的 ARN。

#### Note

您还可以使用单个机密访问角色来提供对 `SecretsManagerSecretId` 密钥和密钥的 AWS DMS `SecretsManagerOracleAsmSecretId` 访问权限。如果您为这两个密钥创建单个秘

密访问角色，请确保将此访问角色的相同 ARN 分配给 `SecretsManagerAccessRoleArn` 和 `SecretsManagerOracleAsmAccessRoleArn`。例如，如果两个密钥的秘密访问角色的 ARN 分配给了变量 `ARN2xsecrets`，则可按如下方式设置 ARN：

```
SecretsManagerAccessRoleArn = ARN2xsecrets;
SecretsManagerOracleAsmAccessRoleArn = ARN2xsecrets;
```

有关创建这些值的更多信息，请参阅[使用创建 AWS Management Console 机密和秘密访问角色](#)。

为端点创建并指定所需的密钥及秘密访问角色端点设置后，使用这些密钥详细信息更新将运行 `CreateEndpoint` 或 `ModifyEndpoint` API 请求的用户账户的权限。确保这些账户权限包括对 `IAM:GetRole` 秘密访问角色的 `SecretsManager:DescribeSecret` 权限和对机密的权限。AWS DMS 需要这些权限才能验证访问角色及其密钥。

提供和验证所需的用户权限

1. 登录 AWS Management Console 并打开 AWS Identity and Access Management 控制台，网址为 <https://console.aws.amazon.com/iam/>。
2. 选择用户，然后选择用于发出 `CreateEndpoint` 和 `ModifyEndpoint` API 调用的用户 ID。
3. 从权限选项卡中选择 {} JSON。
4. 请确保用户具有以下所示的权限。

```
{
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "iam:GetRole",
      "iam:PassRole"
    ],
    "Resource": "SECRET_ACCESS_ROLE_ARN"
  },
  {
    "Effect": "Allow",
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "SECRET_ARN"
  }
]
```

5. 如果用户没有这些权限，添加权限。
6. 如果使用 IAM 角色发出 DMS API 调用，请对相应的角色重复上述步骤。
7. 打开终端，使用假设上面使用的角色或用户来验证权限是否正确授予。AWS CLI
  - a. SecretAccessRole 使用 IAM `get-role` 命令验证用户的权限。

```
aws iam get-role --role-name ROLE_NAME
```

将 *ROLE\_NAME* 替换为 `SecretsManagerAccessRole` 的名称。

如果命令返回错误消息，请确保已正确授予权限。

- b. 使用 Secrets Manager `describe-secret` 命令验证用户对密钥的权限。

```
aws secretsmanager describe-secret --secret-id SECRET_NAME OR SECRET_ARN --  
region=REGION_NAME
```

用户可以是友好名称、部分 ARN 或完整 ARN。有关更多信息，请参阅 [describe-secret](#)。

如果命令返回错误消息，请确保已正确授予权限。

## 使用创建 AWS Management Console 机密和秘密访问角色

您可以使用创建 AWS Management Console 用于终端节点身份验证的密钥，并创建允许您代表您 AWS DMS 访问该密钥的策略和角色。

使用 AWS DMS 可用于对数据库进行源端点和目标端点连接进行身份验证的密钥来创建密钥 AWS Management Console

1. 登录 AWS Management Console 并打开 AWS Secrets Manager 控制台，网址为 <https://console.aws.amazon.com/secretsmanager/>。
2. 选择存储新密钥。
3. 在存储新密钥页面上，在选择密钥类型下选择其他类型的密钥，然后选择明文。

**Note**

从此刻起，这是唯一需要输入明文凭证才能连接到端点数据库的地方。

## 4. 在明文字段中：

- 对于身份分配给 `SecretsManagerSecretId` 的密钥，输入以下 JSON 结构。

```
{
  "username": db_username,
  "password": db_user_password,
  "port": db_port_number,
  "host": db_server_name
}
```

**Note**

这是对端点数据库进行身份验证所需的最少 JSON 成员列表。您可以根据需要将任何其他 JSON 端点设置添加为 JSON 成员，全部采用小写。但是，对于端点身份验证，AWS DMS 会忽略任何其他 JSON 成员。

此处，*db\_username* 是访问数据库的用户的名称，*db\_user\_password* 是数据库用户的密码，*db\_port\_number* 是访问数据库的端口号，*db\_server\_name* 是 Web 上的数据库服务器名称（地址），如下例所示。

```
{
  "username": "admin",
  "password": "some_password",
  "port": "8190",
  "host": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com"
}
```

- 对于身份分配给 `SecretsManagerOracleAsmSecretId` 的密钥，输入以下 JSON 结构。

```
{
  "asm_user": asm_username,
  "asm_password": asm_user_password,
  "asm_server": asm_server_name
}
```



```
}
```

**Note**

这是对 Oracle 端点的 Oracle ASM 进行身份验证所需的最少 JSON 成员列表。也是您可以根据可用的 Oracle ASM 端点设置指定的完整列表。

此处，*asm\_username* 是访问 Oracle ASM 的用户的名称，*asm\_user\_password* 是 Oracle ASM 用户的密码，*asm\_server\_name* 是 Web 上的 Oracle ASM 服务器名称（地址），包括端口，如下例所示。

```
{
  "asm_user": "oracle_asm_user",
  "asm_password": "oracle_asm_password",
  "asm_server": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com:8190/+ASM"
}
```

5. 选择 AWS KMS 加密密钥来加密密钥。您可以接受由为您的服务创建的默认加密密钥，AWS Secrets Manager 也可以选择自己创建的 AWS KMS 密钥。
6. 指定引用此密钥的名称和可选的描述。这是您用作 `SecretsManagerSecretId` 或 `SecretsManagerOracleAsmSecretId` 值的友好名称。
7. 如果要在密钥中启用自动轮换，则需要选择或创建一个 AWS Lambda 函数，该函数有权轮换密钥的证书，如上所述。但是，在设置自动轮换以使用 Lambda 函数之前，请确保该函数的配置设置将以下四个字符添加到 `EXCLUDE_CHARACTERS` 环境变量的值中。

```
;.:+{}
```

AWS DMS 不允许在用于端点凭证的密码中使用这些字符。配置 Lambda 函数以排除这些字符，可以防止 AWS Secrets Manager 生成这些字符作为其轮换密码值的一部分。将自动轮换设置为使用 Lambda 函数后，AWS Secrets Manager 立即轮换密钥以验证您的密钥配置。

**Note**

根据数据库引擎配置的不同，数据库可能无法获取轮换后的凭证。在这种情况下，您需要手动重新启动任务以刷新凭证。

- 查看您的密钥并将其存储在中 AWS Secrets Manager。然后，您可以在中按其友好名称查找每个密钥 AWS Secrets Manager，然后检索密钥 ARN 作为值，`SecretsManagerSecretId`或`SecretsManagerOracleAsmSecretId`根据需要检索该密钥 ARN，以验证对终端节点数据库连接和 Oracle ASM ( 如果使用 ) 的访问权限。

创建秘密访问策略和角色来设置您

的`SecretsManagerAccessRoleArn`或`SecretsManagerOracleAsmAccessRoleArn`，这 AWS DMSAWS Secrets Manager 允许访问您的相应密钥

- 登录 AWS Management Console 并打开 AWS Identity and Access Management (IAM) 控制台，[网址为 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/)。
- 选择策略，然后选择创建策略。
- 选择 JSON 并输入以下策略，以启用对密钥的访问和解密。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": secret_arn,
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
      ],
      "Resource": kms_key_arn,
    }
  ]
}
```

此处，*secret\_arn* 是密钥的 ARN，您可以根据需要从 `SecretsManagerSecretId` 或 `SecretsManagerOracleAsmSecretId` 中获取，*kms\_key\_arn* 是您用于加密密钥的 AWS KMS 密钥的 ARN，如以下示例所示。

```
{
  "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

如果您使用由创建的默认加密密钥 AWS Secrets Manager，则无需为指定 AWS KMS 权限 `kms_key_arn`。

如果您希望策略提供对两个密钥的访问权限，只需为另一个 `secret_arn` 指定一个额外的 JSON 资源对象。

如果您的密钥位于不同账户中，那么 `SecretsManagerAccessRoleArn` 角色需要额外的策略来验证跨账户密钥。对于此类使用案例，请将操作 `secretsmanager:DescribeSecret` 添加到策略中。有关设置跨账户密钥的更多详细信息，请参阅 [其他账户中的用户对 S AWS secrets Manager 密钥的权限](#)。

4. 查看并创建具友好名称和描述（可选）的策略。
5. 选择角色，然后选择创建角色。
6. 选择 AWS 服务作为可信实体的类型。
7. 从服务列表中选择 DMS 作为可信服务，然后选择下一步：权限。
8. 查找并附加您在步骤 4 中创建的策略，然后继续添加所有标签并查看角色。此时，编辑该角色的信任关系，以使用您的 AWS DMS 区域服务委托人作为可信实体。此主体采用以下格式。

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

此处，*region-name* 是区域的名称，如 `us-east-1`。因此，该 AWS DMS 区域的区域服务负责人随之而来。

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

9. 编辑完角色的可信实体后，创建具友好名称和描述（可选）的角色。现在，您可以在 IAM 中按友好名称查找新角色，然后检索角色 ARN 作为 `SecretsManagerAccessRoleArn` 或 `SecretsManagerOracleAsmAccessRoleArn` 值，以验证端点数据库连接。

在专用子网中将密钥管理器与复制实例一起使用

1. 创建密钥管理器 VPC 端点并记下该端点的 DNS。有关创建密钥管理器 VPC 端点的更多信息，请参阅《AWS Secrets Manager 用户指南》中的[通过 VPC 端点连接到 Secrets Manager](#)。
2. 将复制实例安全组附加到密钥管理器 VPC 端点。
3. 对于复制实例安全组出站规则，允许目的地为 `0.0.0.0/0` 的所有流量。
4. 设置端点额外连接属性，`secretsManagerEndpointOverride=secretsManager endpoint DNS`，以提供密钥管理器 VPC 端点 DNS，如以下示例所示。

```
secretsManagerEndpointOverride=vpce-1234a5678b9012c-12345678.secretsmanager.eu-west-1.vpce.amazonaws.com
```

## 将服务相关角色用于 AWS DMS

AWS Database Migration Service 使用 AWS Identity and Access Management (IAM) [服务相关角色](#)。服务相关角色是一种独特类型的 IAM 角色，它与 AWS DMS 直接相关。服务相关角色由 AWS DMS 预定义，并包含该服务代表您调用其他 AWS 服务所需的一切权限。

服务相关角色使 AWS DMS 的设置更轻松，因为您不必手动添加必要的权限。AWS DMS 定义其服务相关角色的权限，除非另行定义，否则仅 AWS DMS 可以代入其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其他 IAM 实体的权限策略。

只有在首先删除相关资源后，才能删除服务相关角色。这将保护您的 AWS DMS 资源，因为您不会无意中删除对资源的访问权限。

有关支持服务相关角色的其他服务的信息，请参阅[与 IAM 配合使用的 AWS 服务](#)，并查找 Service-linked roles（服务相关角色）列中显示为 Yes（是）的服务。请选择 Yes 与查看该服务的服务相关角色文档的链接。

## AWS DMS 功能的服务相关角色

### 主题

- [AWS DMS Fleet Advisor 的服务相关角色](#)
- [AWS DMS Serverless 的服务相关角色](#)

## AWS DMS Fleet Advisor 的服务相关角色

AWS DMS Fleet Advisor 使用名为 `AWSServiceRoleForDMSFleetAdvisor` 的服务相关角色 – DMS Fleet Advisor 使用此服务相关角色管理 Amazon CloudWatch 指标。此服务相关角色附加到以下托管策略：`AWSDMSFleetAdvisorServiceRolePolicy`。有关此策略的更新，请参阅[AWS 的托管策略](#) [AWS Database Migration Service](#)。

`AWSServiceRoleForDMSFleetAdvisor` 服务相关角色信任以下服务以担任该角色：

- `dms-fleet-advisor.amazonaws.com`

名为 `AWSDMSFleetAdvisorServiceRolePolicy` 的角色权限策略允许 AWS DMS Fleet Advisor 对指定资源完成以下操作：

- 操作：`all AWS resources` 上的 `cloudwatch:PutMetricData`

此权限允许主体将指标数据点发布到 Amazon CloudWatch。AWS DMS Fleet Advisor 需要此权限才能从 CloudWatch 显示数据库指标图表。

以下代码示例显示了用于创建 `AWSDMSFleetAdvisorServiceRolePolicy` 角色的 `AWSDMSFleetAdvisorServiceRolePolicy` 策略。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Resource": "*",
    "Action": "cloudwatch:PutMetricData",
    "Condition": {
      "StringEquals": {
        "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
      }
    }
  }
}
```

```
}  
}  
}
```

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[服务相关角色权限](#)。

#### 为 AWS DMS Fleet Advisor 创建服务相关角色

您可以通过 IAM 控制台，使用 DMS – Fleet Advisor 使用案例创建服务相关角色。在 AWS CLI 或 AWS API 中，使用 `dms-fleet-advisor.amazonaws.com` 服务名称创建服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[创建服务相关角色](#)。如果您删除了此服务相关角色，可以使用同样的过程再次创建角色。

请务必在创建数据收集器之前创建此角色。DMS Fleet Advisor 在 AWS Management Console 中使用此角色显示数据库指标图表。有关更多信息，请参阅[创建数据收集器](#)。

#### 为 AWS DMS Fleet Advisor 编辑服务相关角色

AWS DMS 不允许编辑 `AWSServiceRoleForDMSFleetAdvisor` 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务相关角色](#)。

#### 为 AWS DMS Fleet Advisor 删除服务相关角色

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样，您便没有未被主动监控或维护的未使用实体。但是，您必须先清除服务相关角色的资源，然后才能手动删除它。

#### Note

如果在您试图删除资源时 AWS DMS 服务正在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟后重试。

#### 删除 `AWSServiceRoleForDMSFleetAdvisor` 使用的 AWS DMS 资源

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 在导航窗格中，在发现下选择数据收集器。数据收集器页面将打开。
3. 选择您的数据收集器，然后选择删除。
4. 要确认删除，在文本输入字段中输入连接器名称。接下来，选择删除。

**⚠ Important**

如果您删除 DMS 数据收集器，DMS Fleet Advisor 会从清单中删除您使用此收集器发现的所有数据库。

删除所有数据收集器后，您可以删除服务相关角色。

### 使用 IAM 手动删除服务相关角色

可使用 IAM 控制台、AWS CLI 或 AWS API 删除 `AWSServiceRoleForDMSFleetAdvisor` 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[删除服务相关角色](#)。

### AWS DMS Fleet Advisor 服务相关角色的受支持区域

AWS DMS Fleet Advisor 支持在服务可用的所有区域中使用服务相关角色。有关更多信息，请参阅[支持 AWS 区域](#)。

### AWS DMS Serverless 的服务相关角色

AWS DMS Serverless 使用名为的服务相关角色。`AWSServiceRoleForDMSServerless` AWS DMS 使用此服务相关角色代表您创建和管理 AWS DMS 资源，例如 Amazon CloudWatch 指标。AWS DMS 使用此角色，因此您只需要关注复制即可。此服务相关角色附加到以下托管策略：`AWSDMSServerlessServiceRolePolicy`。有关此策略的更新，请参阅[AWS 的托管策略 AWS Database Migration Service](#)。

`AWSServiceRoleForDMSServerless` 服务相关角色信任以下服务来代入该角色：

- `dms.amazonaws.com`

以下代码示例显示了您用于创建 `AWSServiceRoleForDMSServerless` 角色的 `AWSDMSServerlessServiceRolePolicy` 策略。

```
{
  "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": "*"
}
```



```
    "Resource": [  
      "arn:aws:dms:*:*:rep:*",  
      "arn:aws:dms:*:*:endpoint:*"  
    ]  
  }  
]
```

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[服务相关角色权限](#)。

### 为 AWS DMS Serverless 创建服务相关角色

创建复制时，AWS DMS 无服务器会以编程方式创建 AWS DMS 无服务器服务关联角色。您可以在 IAM 控制台中查看此角色。也可以选择手动创建此角色。要手动创建角色，请使用 IAM 控制台创建包含 DMS 用例的服务相关角色。在 AWS CLI 或 AWS API 中，使用 `dms.amazonaws.com` 作为服务名称创建服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[创建服务相关角色](#)。如果您删除了此服务相关角色，可以使用同样的过程再次创建角色。

#### Note

如果在账户中存在复制时删除角色，则复制会导致失败。

### 为 AWS DMS Serverless 编辑服务相关角色

AWS DMS 不允许您编辑 `AWSServiceRoleForDMSServerless` 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务相关角色](#)。

### 为 AWS DMS Serverless 删除服务相关角色

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样，您便没有未被主动监控或维护的未使用实体。但是，您必须先清除服务相关角色的资源，然后才能手动删除它。

#### Note

如果您尝试删除资源时 AWS DMS 服务正在使用该角色，则删除可能会失败。如果发生这种情况，请等待几分钟后重试。

## 要删除使用的 AWS DMS 资源 AWSServiceRoleForDMSServerless

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。
2. 在导航窗格中，在发现下选择无服务器。将打开无服务器页面。
3. 选择您的无服务器复制，然后选择删除。
4. 要确认删除，在文本输入字段中输入无服务器复制名称。接下来，选择删除。

删除所有无服务器复制后，可以删除服务相关角色。

### 使用 IAM 手动删除服务相关角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除 AWSServiceRoleForDMSServerless 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[删除服务相关角色](#)。

### AWS DMS Serverless 服务相关角色的受支持区域

AWS DMS Serverless 支持在提供服务的所有地区使用服务相关角色。

## 对 AWS Database Migration Service 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS DMS 和 IAM 时可能遇到的常见问题。

### 主题

- [我无权在以下位置执行操作 AWS DMS](#)
- [我无权执行 iam : PassRole](#)
- [我是一名管理员，想允许其他人访问 AWS DMS](#)
- [我想允许 AWS 账户之外的人访问我的 AWS DMS 资源](#)

### 我无权在以下位置执行操作 AWS DMS

如果 AWS Management Console 告诉您您无权执行某项操作，则必须联系管理员寻求帮助。管理员是指提供用户名和密码的人员。

当 mateojackson IAM 用户尝试使用控制台查看有关 AWS DMS 终端节点的详细信息但没有 dms: DescribeEndpoint 权限时，就会出现以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
dms:DescribeEndpoint on resource: my-postgresql-target
```

在这种情况下，Mateo 请求管理员更新其策略，以允许他使用 `dms:DescribeEndpoint` 操作访问 `my-postgresql-target` 端点资源。

## 我无权执行 `iam:PassRole`

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 AWS DMS。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 AWS DMS 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

## 我是一名管理员，想允许其他人访问 AWS DMS

要允许其他人访问 AWS DMS，您必须为需要访问的人员或应用程序创建 IAM 实体（用户或角色）。它们将使用该实体的凭证访问 AWS。然后，您必须将策略附加到实体，以便在 AWS DMS 中向其授予正确的权限。

要立即开始使用，请参阅《IAM 用户指南》中的 [创建您的第一个 IAM 委派用户和组](#)。

## 我想允许 AWS 账户之外的人访问我的 AWS DMS 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解是否 AWS DMS 支持这些功能，请参阅[如何 AWS Database Migration Service 与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅[IAM 用户指南中的向您拥有 AWS 账户的另一个 IAM 用户提供访问权限](#)。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅[IAM 用户指南中的向第三方提供访问权限](#)。AWS 账户
- 要了解如何通过联合身份验证提供访问权限，请参阅《IAM 用户指南》中的[为经过外部身份验证的用户（联合身份验证）提供访问权限](#)。
- 要了解使用角色和基于资源的策略进行跨账户访问的区别，请参阅[IAM 用户指南中的跨账户资源访问](#)。

## 使用 AWS DMS所需的 IAM 权限

您使用特定 IAM 权限和 IAM 角色来使用 AWS DMS。如果您以 IAM 用户身份登录并想使用 AWS DMS，您的账户管理员必须将本节中讨论的策略附加到您用来运行的 IAM 用户、群组或角色 AWS DMS。有关 IAM 权限的更多信息，请参阅 [IAM 用户指南](#)。

以下策略为您提供访问其他亚马逊服务（例如 IAM AWS DMS AWS KMS、Amazon EC2 和亚马逊）所需的某些操作的权限 CloudWatch。CloudWatch实时监控您的 AWS DMS 迁移，并收集和跟踪表明迁移进度的指标。您可以使用 CloudWatch 日志来调试任务的问题。

### Note

您可以使用标记进一步限制对 AWS DMS 资源的访问权限。有关使用标记限制 AWS DMS 资源访问权限的更多信息，请参阅[使用资源名称和标签进行精细访问控制](#)。

```
{
  "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"
  }
]
}

```

细分以下这些权限可以助力您更好地了解需要每个权限的原因。

以下部分是允许用户调用 AWS DMS API 操作所必需的。

```
{
```

```

    "Effect": "Allow",
    "Action": "dms:*",
    "Resource": "arn:aws:dms:region:account:resourcetype/id"
}

```

要允许用户列出要在控制台中显示的可用 AWS KMS 密钥和别名，则需要以下部分。如果您知道 KMS 密钥的 Amazon 资源名称 (ARN)，并且只使用 AWS Command Line Interface (AWS CLI)，则无需输入此项。

```

{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

对于需要在端点中传入 IAM 角色 ARN 的某些端点类型，需要使用以下部分。此外，如果未提前创建所需的 AWS DMS 角色，则 AWS DMS 控制台可以创建角色。如果提前配置了所有角色，则只需要 `iam:GetRole` 和 `iam:PassRole`。有关角色的更多信息，请参阅[创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

```

{
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

以下部分是必填的，因为 AWS DMS 需要创建 Amazon EC2 实例并为创建的复制实例配置网络。这些资源位于客户的账户中，因此，需要能够代表客户执行这些操作。

```

{
    "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"
}

```

需要使用以下部分，以允许用户查看复制实例指标。

```

{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:Get*",
        "cloudwatch:List*"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

需要使用该部分，以允许用户查看复制日志。

```

{
    "Effect": "Allow",
    "Action": [
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:FilterLogEvents",
        "logs:GetLogEvents"
    ],
    "Resource": "arn:aws:service:region:account:resourcetype/id"
}

```

AWS DMS 控制台会创建多个角色，当您使用控制 AWS DMS 台时，这些角色会自动附加到您的 AWS 账户。如果您使用 AWS Command Line Interface (AWS CLI) 或 AWS DMS API 进行迁移，则需要将这些角色添加到您的账户。有关添加这些角色的更多信息，请参阅[创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。



## 创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色

如果您使用 AWS CLI 或 AWS DMS API 进行数据库迁移，则必须先向 AWS 账户添加三个 IAM 角色，然后才能使用的功能 AWS DMS。其中两个角色为 `dms-vpc-role` 和 `dms-cloudwatch-logs-role`。如果您使用 Amazon Redshift 作为目标数据库，则还必须将 IAM 角色 `dms-access-for-endpoint` 添加到您的 AWS 账户。

对托管策略的更新是自动的。如果您为 IAM 角色使用自定义策略，请确保定期在此文档中检查对托管策略的更新。您可以使用 `get-policy` 和 `get-policy-version` 命令的组合查看托管策略的详细信息。

例如，以下 `get-policy` 命令检索有关指定 IAM 角色的信息。

```
aws iam get-policy --policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole
```

命令返回的信息如下所示。

```
{
  "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"
  }
}
```

以下 `get-policy-version` 命令检索 IAM policy 信息。

```
aws iam get-policy-version --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole --version-id v3
```

命令返回的信息如下所示。

```
{
  "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
  }
}
```

您可以使用相同命令获取有关 AmazonDMSCloudWatchLogsRole 和 AmazonDMSRedshiftS3Role 托管策略的信息。

#### Note

如果您使用 AWS DMS 控制台进行数据库迁移，则这些角色会自动添加到您的 AWS 账户中。

以下过程创建 `dms-vpc-role`、`dms-cloudwatch-logs-role` 和 `dms-access-for-endpoint` IAM 角色。

创建用于 AWS CLI 或 AWS DMS API 的 `dms-vpc-role` IAM 角色

1. 使用以下 IAM policy 创建 JSON 文件。将 JSON 文件命名为 `dmsAssumeRolePolicyDocument.json`。

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

使用以下命令 AWS CLI 使用创建角色。

```
aws iam create-role --role-name dms-vpc-role --assume-role-policy-document file://
dmsAssumeRolePolicyDocument.json
```

2. 使用以下命令将 `AmazonDMSVPCManagementRole` 策略附加到 `dms-vpc-role`。

```
aws iam attach-role-policy --role-name dms-vpc-role --policy-arn
arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole
```

## 创建用于 AWS CLI 或 AWS DMS API 的 dms-cloudwatch-logs-role IAM 角色

1. 使用以下 IAM policy 创建 JSON 文件。将 JSON 文件命名为 `dmsAssumeRolePolicyDocument2.json`。

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

使用以下命令 AWS CLI 使用创建角色。

```
aws iam create-role --role-name dms-cloudwatch-logs-role --assume-role-policy-document file://dmsAssumeRolePolicyDocument2.json
```

2. 使用以下命令将 `AmazonDMSCloudWatchLogsRole` 策略附加到 `dms-cloudwatch-logs-role`。

```
aws iam attach-role-policy --role-name dms-cloudwatch-logs-role --policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSCloudWatchLogsRole
```

如果使用 Amazon Redshift 作为目标数据库，则必须创建 IAM 角色 `dms-access-for-endpoint` 以提供对 Amazon S3 的访问。

## 创建 dms-access-for-endpoint IAM 角色以用于 Amazon Redshift 作为目标数据库

1. 使用以下 IAM policy 创建 JSON 文件。将 JSON 文件命名为 `dmsAssumeRolePolicyDocument3.json`。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "2",
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. 使用以下命令 AWS CLI 使用创建角色。

```
aws iam create-role --role-name dms-access-for-endpoint --assume-role-policy-
document file://dmsAssumeRolePolicyDocument3.json
```

3. 使用以下命令将 AmazonDMSRedshiftS3Role 策略附加到 `dms-access-for-endpoint` 角色。

```
aws iam attach-role-policy --role-name dms-access-for-endpoint \
```

```
--policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSRedshiftS3Role
```

现在，您应该已经准备好了 IAM 策略才能使用 AWS CLI 或 AWS DMS API。

## 防止跨服务混淆座席

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的“服务”）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况，AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

我们建议在资源策略中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文密钥来限制为资源 AWS Database Migration Service 提供其他服务的权限。如果 `aws:SourceArn` 值不包含账户 ID，例如 AWS DMS 复制实例名称（ARN），则您必须使用两个全局条件上下文键来限制权限。如果同时使用全局条件上下文密钥和包含账户 ID 的 `aws:SourceArn` 值，则 `aws:SourceAccount` 值和 `aws:SourceArn` 值中的账户在同一策略语句中使用时，必须使用相同的账户 ID。如果您只希望将一个资源与跨服务访问相关联，请使用 `aws:SourceArn`。如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 `aws:SourceAccount`。

AWS DMS 从 3.4.7 及更高版本开始，支持混乱的副手选项。有关更多信息，请参阅 [AWS Database Migration Service 3.4.7 版本说明](#)。如果您的复制实例使用 AWS DMS 版本 3.4.6 或更低版本，请确保在设置混淆代理选项之前升级到最新版本。

防范混淆代理问题最有效的方法是使用 `aws:SourceArn` 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符字符 (\*) 的 `aws:SourceArn` 全局上下文条件键。例如，`arn:aws:dms:*:123456789012:rep:*`。

### 主题

- [IAM 角色可与 AWS DMS API 配合使用以防止跨服务混淆副手](#)
- [IAM policy 将预检评测存储在 Amazon S3 中，以防跨服务混淆代理](#)
- [使用 Amazon DynamoDB 作为目标终端节点用于跨服务混淆副 AWS DMS 手预防](#)

## IAM 角色可与 AWS DMS API 配合使用以防止跨服务混淆副手

要使用 AWS CLI 或 AWS DMS API 进行数据库迁移，必须先将 `dms-vpc-role` 和 `dms-cloudwatch-logs-role` IAM 角色添加到您的 AWS 账户，然后才能使用的功能 AWS DMS。有关更多信息，请参阅 [创建要与 AWS CLI 和 AWS DMS API 一起使用的 IAM 角色](#)。

以下示例显示了将 `dms-vpc-role` 角色用于 `my-replication-instance` 复制实例的策略。使用这些策略可防止混淆代理问题。

```
{
  "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 policy 将预检评测存储在 Amazon S3 中，以防跨服务混淆代理

要将预评测结果存储在 S3 存储桶中，您需要创建一个允许 AWS DMS 管理 Amazon S3 中的对象的 IAM policy。有关更多信息，请参阅 [创建 IAM 资源](#)。

以下示例显示了在 IAM 角色上设置的带有混淆代理条件的信任策略，该策略 AWS DMS 允许访问指定用户账户下的所有任务和评估。

```
{
  "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:*"
            ]
        }
    }
}
]
}

```

## 使用 Amazon DynamoDB 作为目标终端节点用于跨服务混淆副 AWS DMS 手预防

要使用 Amazon DynamoDB 作为数据库迁移的目标终端节点，您必须创建 AWS DMS 允许代入和授予对 DynamoDB 表的访问权限的 IAM 角色。然后，当您在 AWS DMS 中创建目标 DynamoDB 端点时使用此角色。有关更多信息，请参阅 [将 Amazon DynamoDB 作为目标](#)。

以下示例显示了在允许所有 AWS DMS 终端节点访问 DynamoDB 表的 IAM 角色上设置的具有混淆代理条件的信任策略。

```

{
  "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 的托管策略 AWS Database Migration Service

### 主题

- [AWS 托管策略：AmazonDMSVPC ManagementRole](#)
- [AWS 托管策略：AWSDMSServerlessServiceRolePolicy](#)
- [AWS 托管策略：AmazonDMS CloudWatch LogsRole](#)
- [AWS 托管策略：AWSDMSFleetAdvisorServiceRolePolicy](#)
- [AWS DMSAWS 托管策略的更新](#)

### AWS 托管策略：AmazonDMSVPC ManagementRole

该策略附属于该dms-vpc-role角色，AWS DMS 允许您代表您执行操作。

此策略向贡献者授予 AWS DMS 允许管理网络资源的权限。

### 权限详细信息

本策略包括以下操作：

- `ec2:CreateNetworkInterface`— AWS DMS 需要此权限才能创建网络接口。这些接口对于 AWS DMS 复制实例连接到源数据库和目标数据库至关重要。
- `ec2:DescribeAvailabilityZones`— 此权限 AWS DMS 允许检索有关区域中可用区域的信息。AWS DMS 使用此信息来确保在正确的区域中配置资源以实现冗余和可用性。
- `ec2:DescribeInternetGateways`— AWS DMS 可能需要此权限才能理解 VPC 中配置的互联网网关。如果复制实例或数据库需要访问互联网，则此信息至关重要。
- `ec2:DescribeSecurityGroups`— 安全组控制实例和资源的入站和出站流量。AWS DMS 需要描述安全组，以正确配置网络接口并确保复制实例和数据库之间的正常通信。
- `ec2:DescribeSubnets`— 此权限 AWS DMS 允许列出 VPC 中的子网。AWS DMS 使用此信息启动相应子网中的复制实例，确保它们具有必要的网络连接。
- `ec2:DescribeVpcs`— 描述 VPC 对于 AWS DMS 了解复制实例和数据库所在的网络环境至关重要。这包括了解 CIDR 块和其他特定于 VPC 的配置。
- `ec2>DeleteNetworkInterface`— AWS DMS 需要此权限才能在不再需要时清理它创建的网络接口。这有助于资源管理并避免不必要的成本。

- `ec2:ModifyNetworkInterfaceAttribute`— 修改其管理 AWS DMS 的网络接口的属性需要此权限。这可能包括调整设置以确保连接性和安全性。
- `ec2:DescribeDhcpOptions`— AWS DMS 检索指定 VPC 的 DHCP 选项集详细信息。这些信息是为复制实例正确配置网络所必需的。
- `ec2:DescribeNetworkInterfaces`— AWS DMS 检索 VPC 内现有网络接口的相关信息。这些信息是正确配置网络接口和确保迁移过程的正确网络连接所必需的。AWS DMS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2>DeleteNetworkInterface",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    }
  ]
}
```

## AWS 托管策略：AWSDMSServerlessServiceRolePolicy

该策略附属于该 `AWSServiceRoleForDMSServerless` 角色，AWS DMS 允许您代表您执行操作。有关更多信息，请参阅 [AWS DMS Serverless 的服务相关角色](#)。

此策略向贡献者授予 AWS DMS 允许管理复制资源的权限。

### 权限详细信息

该策略包含以下权限。

- `dms`— 允许委托人与 AWS DMS 资源交互。

```
{
  "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:*"
    ]
  }
]
}

```

## AWS 托管策略 : AmazonDMS CloudWatch LogsRole

该策略附属于该 `dms-cloudwatch-logs-role` 角色，AWS DMS 允许您代表您执行操作。有关更多信息，请参阅 [将服务相关角色用于 AWS DMS](#)。

此策略向贡献者授予允许 AWS DMS 向日志发布复制 CloudWatch 日志的权限。

### 权限详细信息

该策略包含以下权限。

- `logs`— 允许委托人将日志发布到日 CloudWatch 志。此权限是必需的，这样 AWS DMS 才能使用它 CloudWatch 来显示复制日志。

```

{
  "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-*"
    ]
}
]
}

```

## AWS 托管策略：AWSDMSFleetAdvisorServiceRolePolicy

您无法附加 AWSDMSFleetAdvisorServiceRolePolicy 到您的 IAM 实体。此策略附加到服务相关角色，该角色允许 AWS DMS Fleet Advisor 代表您执行操作。有关更多信息，请参阅 [将服务相关角色用于 AWS DMS](#)。

该策略向贡献者授予权限，允许 AWS DMS Fleet Advisor 发布亚马逊 CloudWatch 指标。

### 权限详细信息

该策略包含以下权限。

- **cloudwatch**— 允许委托人向 Amazon CloudWatch 发布指标数据点。此权限是必需的，这样 AWS DMS Fleet Advisor CloudWatch 才能使用它来显示包含数据库指标的图表。

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Resource": "*",
    "Action": "cloudwatch:PutMetricData",
    "Condition": {
      "StringEquals": {
        "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
      }
    }
  }
}

```

## AWS DMS AWS 托管策略的更新

查看 AWS DMS 自该服务开始跟踪这些更改以来 AWS 托管策略更新的详细信息。要获得有关此页面变更的自动提醒，请订阅“AWS DMS 文档历史记录”页面上的 RSS feed。

更改	描述	日期
<a href="#">AmazonDMSV ManagementRole PC — 更改</a>	AWS DMS 添加 <code>ec2:DescribeDhcpOptions</code> 了 AWS DMS 允许代表您管理网络设置的 <code>ec2:DescribeNetworkInterfaces</code> 操作。	2024年6月17日
<a href="#">AWSDMSServerlessServiceRolePolicy : 新策略</a>	AWS DMS 添加了 AWS DMS 允许代表您创建和管理服务的 <code>AWSDMSServerlessServiceRolePolicy</code> 角色，例如发布 Amazon CloudWatch 指标。	2023 年 5 月 22 日
<a href="#">AmazonDMS — 更改 CloudWatch LogsRole</a>	AWS DMS 在授予的每个权限中添加了无服务器资源的 ARN，以允许将无服务器复制配置中的 AWS DMS 复制日志上传到日志。CloudWatch	2023 年 5 月 22 日
<a href="#">AWSDMSFleetAdvisor ServiceRolePolicy : 新策略</a>	AWS DMS Fleet Advisor 添加了一项新政策，允许向亚马逊发布指标数据点 CloudWatch。	2023 年 3 月 6 日
AWS DMS 开始跟踪更改	AWS DMS 开始跟踪其 AWS 托管策略的更改。	2023 年 3 月 6 日



## AWS Database Migration Service 的合规性验证

作为多个 AWS Database Migration Service 合规性计划的一部分，第三方审核员将评估 AWS 的安全性和合规性。这包括以下程序：

- SOC
- PCI
- ISO
- FedRAMP
- DoD CC SRG
- HIPAA BAA
- MTCS
- CS
- K-ISMS
- ENS High
- OSPAR
- HITRUST CSF

有关特定合规性计划范围内的 AWS 服务列表，请参阅[合规性计划范围内的 AWS 服务](#)。有关常规信息，请参阅[AWS 合规性计划](#)。

您可以使用 AWS Artifact 下载第三方审计报告。有关更多信息，请参阅[下载 AWS 构件中的报告](#)。

您使用 AWS DMS 的合规性责任取决于您数据的敏感度、贵公司的合规性目标以及适用的法律法规。AWS 提供以下资源来帮助满足合规性：

- [安全性与合规性 Quick Start 指南](#) – 这些部署指南讨论了架构注意事项，并提供了在 AWS 上部署关注安全性和合规性的基准环境的步骤。
- [Amazon Web Services 上的 HIPAA 安全性和合规性架构设计白皮书](#) – 此白皮书介绍了公司如何使用 AWS 创建符合 HIPAA 标准的应用程序。
- [AWS 合规性资源](#) – 此业务手册和指南集合可能适用于您的行业和位置。
- [AWS Config](#) – 此 AWS 服务评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- [AWS Security Hub](#) – 此 AWS 服务提供了 AWS 中安全状态的全面视图，可帮助您检查是否符合安全行业标准和最佳实践。



## AWS Database Migration Service 中的故障恢复能力

AWS全球基础设施围绕AWS区域和可用区构建。AWS区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅 [AWS 全球基础设施](#)。

除 AWS 全球基础设施之外，在选择多可用区选项时，AWS DMS 会为使用多可用区部署的复制实例提供高可用性和故障转移支持。

在多可用区部署中，AWS 自动在不同可用区中预置和维护复制实例的备用副本。主复制实例将同步复制到备用副本。如果主复制实例发生故障或没有响应，备用副本将以最少中断恢复任何正在运行的任务。由于主复制实例会不断将其状态复制到备用副本中，因此多可用区部署会产生一些性能开销。

有关使用多可用区部署的更多信息，请参阅 [使用 AWS DMS 复制实例](#)。

# AWS Database Migration Service 中的基础设施安全性

作为一项托管式服务，AWS Database Migration Service 受 AWS 全球网络安全保护。有关 AWS 安全服务以及 AWS 如何保护基础架构的信息，请参阅 [AWS 云安全](#)。要按照基础设施安全最佳实践设计您的 AWS 环境，请参阅《安全性支柱 AWS Well-Architected Framework》中的 [基础设施保护](#)。

您可以使用 AWS 发布的 API 调用通过网络访问 AWS DMS。客户端必须支持以下内容：

- 传输层安全性协议 ( TLS ) 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 ( PFS ) 的密码套件，例如 DHE ( Ephemeral Diffie-Hellman ) 或 ECDHE ( Elliptic Curve Ephemeral Diffie-Hellman )。大多数现代系统 ( 如 Java 7 及更高版本 ) 都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 生成临时安全凭证来对请求进行签名。

您可以从任何网络位置调用这些 API 操作。AWS DMS 还支持基于资源的访问策略，可以在其中指定对操作和资源的限制，例如，基于源 IP 地址的限制。此外，可以使用 AWS DMS 策略控制来自特定 Amazon VPC 端点或特定虚拟私有云 (VPC) 的访问。事实上，这隔离了在 AWS 网络中仅从特定 VPC 到给定 AWS DMS 资源的网络访问。有关将基于资源的访问策略与 AWS DMS 结合使用的更多信息 ( 包括示例 )，请参阅 [使用资源名称和标签进行精细访问控制](#)。

要将与 AWS DMS 的通信限制在单个 VPC 内，您可以创建一个允许通过 AWS PrivateLink 连接到 AWS DMS 的 VPC 接口端点。AWS PrivateLink 可协助确保对 AWS DMS 及其关联结果的任何调用，限制在为其创建接口端点的特定 VPC 内。然后，您可以将此接口端点的 URL 指定为使用 AWS CLI 或 SDK 运行的每一个 AWS DMS 命令的选项。这样做有助于确保您与 AWS DMS 的整个通信始终限制在 VPC 内，并且对公共互联网是不可见的。

创建接口端点以在单个 VPC 中访问 DMS

1. 登录到 AWS Management Console，然后通过以下网址打开 Amazon VPC 控制台：<https://console.aws.amazon.com/vpc/>。
2. 在导航窗格中，选择端点。将打开创建端点页面，您可以在其中创建从 VPC 到 AWS DMS 的接口端点。
3. 选择 AWS 服务，然后搜索并为服务名称选择一个值，在本例中 AWS DMS 为以下形式。

```
com.amazonaws.region.dms
```

此处，*region* 指的是 AWS DMS 运行的 AWS 区域，例如 `com.amazonaws.us-west-2.dms`。

4. 对于 VPC，选择要从其中创建接口端点的 VPC，例如 `vpc-12abcd34`。
5. 为可用区和子网 ID 选择值。这些值应该表示所选 AWS DMS 端点可以运行的位置，例如 `us-west-2a` (`usw2-az1`) 和 `subnet-ab123cd4`。
6. 选择启用 DNS 名称，以创建带有 DNS 名称的端点。此 DNS 名称由端点 ID (`vpce-12abcd34efg567hij`) 和随机字符串 (`ab12dc34`) 组成，中间用连字符连接。它们与服务名称之间用点分隔开，服务名称采用反向点分隔，并增加了 `vpce` (`dms.us-west-2.vpce.amazonaws.com`)。

例如，`vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com`。

7. 对于安全组，选择要用于端点的组。

设置安全组时，请确保允许来自安全组内部的出站 HTTPS 调用。有关更多信息，请参阅《Amazon VPC 用户指南》中的[创建安全组](#)。

8. 对于策略，选择完全访问权限或自定义值。例如，您可以选择类似以下代码的自定义策略，限制端点对某些操作和资源的访问权限。

```
{
  "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": "*"
    }
  ]
}
```

此处的示例策略允许任何 AWS DMS API 调用，但删除或修改特定复制实例除外。

现在，您可以将使用在步骤 6 中创建的 DNS 名称形成的 URL，指定为选项。您可以使用创建的接口端点，为每个 AWS DMS CLI 命令或 API 操作指定此选项，以访问服务实例。例如，可以在此 VPC 中运行 DMS CLI 命令 `DescribeEndpoints`，如下所示。

```
$ aws dms describe-endpoints --endpoint-url https://vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com
```

如果启用私有 DNS 选项，则不必在请求中指定端点 URL。

有关创建和使用 VPC 接口端点（包括启用私有 DNS 选项）的更多信息，请参阅《Amazon VPC 用户指南》中的[接口 VPC 端点 \(AWS PrivateLink\)](#)。

## 使用资源名称和标签进行精细访问控制

您可以使用基于 Amazon 资源名称 (ARN) 的资源名称和资源标签来管理对 AWS DMS 资源的访问权限。为此，请在 IAM 策略中定义允许的操作或包括条件语句。

### 使用资源名称控制访问

您可以创建一个 IAM 用户账户，并根据 AWS DMS 资源的 ARN 分配策略。

以下策略拒绝访问带有 ARN `arn:aws:dms:us-east-1:152683116:rep:doh67ztoxglixmihkitv` 的 AWS DMS 复制实例：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-east-1:152683116:rep:D0H67ZTOXGLIXMIHKITV"
    }
  ]
}
```

例如，在该策略生效时，以下命令将失败：

```
$ aws dms delete-replication-instance
  --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:D0H67ZTOXGLIXMIHKITV"

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:D0H67ZTOXGLIXMIHKITV

$ aws dms modify-replication-instance
```

```
--replication-instance-arn "arn:aws:dms:us-east-1:152683116:rep:D0H67ZTOXGLIXMIHKITV"
```

```
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:D0H67ZTOXGLIXMIHKITV
```

您还可以指定 IAM 策略来限制对 AWS DMS 终端节点和复制任务的访问。

以下策略使用 AWS DMS 终端节点的 ARN 限制对终端节点的访问。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNH0A6XRQSZCUGX"
    }
  ]
}
```

例如，在使用终端节点的 ARN 的策略生效时，以下命令将失败。

```
$ 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
```

以下策略使用 AWS DMS 任务的 ARN 限制对任务的访问权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:dms:us-east-1:152683116:task:U03YR4N47DXH3ATT4YMW0IT"
    }
  ]
}
```

例如，在使用任务的 ARN 的策略生效时，以下命令将失败。

```
$ 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
```

## 使用标签控制访问

AWS DMS 定义了一组常用的键值对，可在客户定义的策略中使用，无需任何额外的标记要求。有关为 AWS DMS 资源添加标签的更多信息，请参阅[在 AWS Database Migration Service 中标记资源](#)。

以下列出了可与之配合使用的标准标签 AWS DMS：

- `aw CurrentTime s`：— 表示请求日期和时间，允许根据时间标准限制访问权限。
- `aws: EpochTime` — 此标签与前面的 `aws: CurrentTime` 标签类似，不同之处在于当前时间表示为自 Unix 时代以来经过的秒数。
- `aws: MultiFactorAuthPresent` — 这是一个布尔标记，用于指示请求是否通过多重身份验证签名。
- `aws: MultiFactorAuthAge` — 提供对多因素身份验证令牌使用年限（以秒为单位）的访问权限。
- `aws: principaltype` — 提供对当前请求的主体类型（用户、账户、联合用户等）的访问。
- `aw SourceIp s`：— 表示发出请求的用户的源 IP 地址。
- `aw UserAgent s`：— 提供有关请求资源的客户端应用程序的信息。
- `aws:userid` — 提供对发出请求的用户的 ID 的访问。
- `aws:username` — 提供对发出请求的用户的名称的访问。
- `dms: InstanceClass` — 提供对复制实例主机的计算大小的访问权限。
- `dms: StorageSize` — 提供对存储卷大小（以 GB 为单位）的访问。

您还可以定义自己的标签。客户定义的标签是保存在标签服务中的简单键值对。AWS 您可以将它们添加到 AWS DMS 资源，包括复制实例、终端节点和任务。这些标签是通过使用策略中的 IAM“条件”语句匹配的，并使用特定的条件标签引用这些标签。标签键前面带有“dms”、资源类型和“tag”前缀。下面显示了标签格式。

```
dms:{resource type}-tag/{tag key}={tag value}
```

例如，假设您要定义一个策略，以仅允许包含“stage=production”标签的复制实例成功完成 API 调用。以下条件语句会匹配具有给定标签的资源。

```
"Condition":
{
  "streq":
  {
    "dms:rep-tag/stage": "production"
  }
}
```

```
}
```

您将以下标签添加到与该策略条件匹配的复制实例中。

```
stage production
```

除了已分配给 AWS DMS 资源的标签外，还可以编写策略来限制可以应用于给定资源的标签键和值。在此情况下，标签前缀为“req”。

例如，以下策略语句将用户可为给定资源分配的标签限制为特定的允许值列表。

```
"Condition":
{
  "streq":
  {
    "dms:rep-tag/stage": [ "production", "development", "testing" ]
  }
}
```

以下策略示例根据 AWS DMS 资源标签限制对资源的访问权限。

以下策略限制对标签值为“Desktop”且标签键为“Env”的复制实例的访问：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "dms:rep-tag/Env": [
            "Desktop"
          ]
        }
      }
    }
  ]
}
```

```
]
}
```

根据在标签值为“Desktop”且标签键为“Env”时限制访问的 IAM 策略，以下命令成功或失败。

```
$ aws dms list-tags-for-resource
  --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7J0JY0JXWDOZNFEN
  --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:46DH0U7J0JY0JXWDOZNFEN
```

```
$ aws dms remove-tags-from-resource
  --resource-name arn:aws:dms:us-east-1:152683116:rep:46DH0U7J0JY0JXWDOZNFEN
  --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:46DH0U7J0JY0JXWDOZNFEN
```

以下策略限制对标签值为“桌面”且标签键为“Env”的 AWS DMS 端点的访问权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "dms:endpoint-tag/Env": [
            "Desktop"
          ]
        }
      }
    }
  ]
}
```

根据在标签值为“Desktop”且标签键为“Env”时限制访问的 IAM 策略，以下命令成功或失败。

```
$ aws dms list-tags-for-resource
```

```
--resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLF52344IZWA6I
{
  "TagList": [
    {
      "Value": "Desktop",
      "Key": "Env"
    }
  ]
}
```

```
$ aws dms delete-endpoint
```

```
--endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLF52344IZWA6I"
```

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:J2YCZPNGOLFY52344IZWA6I
```

以下策略限制对标签值为“Desktop”且标签键为“Env”的复制任务的访问。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dms:*"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "dms:task-tag/Env": [
            "Desktop"
          ]
        }
      }
    }
  ]
}
```

根据在标签值为“Desktop”且标签键为“Env”时限制访问的 IAM 策略，以下命令成功或失败。

```
$ 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



## 设置加密密钥并指定 AWS KMS 权限

AWS DMS 对复制实例使用的存储空间和端点连接信息进行加密。要加密复制实例使用的存储，请 AWS DMS 使用您的 AWS 账户独有的 AWS Key Management Service (AWS KMS) 密钥。您可以使用查看和管理此密钥 AWS KMS。您可以使用您账户 (aws/dms) 中的默认 KMS 密钥，也可以创建自定义 KMS 密钥。如果您有现有的 KMS 密钥，也可以使用该密钥加密。

### Note

用作加密 AWS KMS 密钥的任何自定义密钥或现有密钥都必须是对称密钥。AWS DMS 不支持使用非对称加密密钥。有关对称和非对称加密密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 <https://docs.aws.amazon.com/kms/latest/developerguide/symmetric-asymmetric.html>。

在首次启动复制实例时，如果未从创建复制实例页面的高级部分中选择自定义 KMS 密钥，将创建默认 KMS 密钥 (aws/dms)。如果您使用默认的 KMS 密钥，您只需要将 kms:ListAliases 和 kms:DescribeKey 权限授予用于迁移的 IAM 用户账户。有关使用默认 KMS 密钥的更多信息，请参阅 [使用 AWS DMS 所需的 IAM 权限](#)。

要使用自定义 KMS 密钥，请使用以下选项之一为自定义 KMS 密钥分配权限：

- 将用于迁移的 IAM 用户账户添加为 AWS KMS 自定义密钥的密钥管理员或密钥用户。这样做可确保将必需的 AWS KMS 权限授予 IAM 用户账户。除了向 IAM 用户账户授予使用 AWS DMS 的 IAM 权限之外，还需执行此操作。有关向密钥用户授予权限的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 [允许密钥用户使用 KMS 密钥](#)。
- 如果您不希望添加 IAM 用户账户作为自定义 KMS 密钥的密钥管理员或密钥用户，则除了必须向 IAM 用户账户授予使用 AWS DMS 的 IAM 权限以外，还需授予以下附加权限。

```
{
    "Effect": "Allow",
    "Action": [
        "kms:ListAliases",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:Encrypt",
        "kms:ReEncrypt*"
    ],
```

```
"Resource": "*"
},
```

AWS DMS 也适用于 KMS 密钥别名。有关创建您自己的 AWS KMS 密钥以及授予用户对 KMS 密钥的访问权限的更多信息，请参阅《[AWS KMS 开发人员指南](#)》。

如果您未指定 KMS 密钥标识符，则 AWS DMS 使用您的默认加密密钥。AWS KMS 为您的 AWS 账户创建默认加密密钥。您的 AWS 账户在每个 AWS 区域都有不同的默认加密密钥。

要管理用于加密 AWS DMS 资源的 AWS KMS 密钥，请使用 AWS Key Management Service。AWS KMS 将安全、高度可用的硬件和软件相结合，提供可扩展到云端的密钥管理系统。使用 AWS KMS，您可以创建加密密钥并定义控制如何使用这些密钥的策略。

您可以在 AWS KMS 里面找到 AWS Management Console

1. 登录 AWS Management Console 并打开 AWS Key Management Service (AWS KMS) 控制台，[网址为 https://console.aws.amazon.com/kms](https://console.aws.amazon.com/kms)。
2. 要更改 AWS 区域，请使用页面右上角的区域选择器。
3. 选择以下选项之一来使用 AWS KMS 按键：
  - 要查看您的账户中为您 AWS 创建和管理的密钥，请在导航窗格中选择 AWS 托管密钥。
  - 要查看您账户中自己所创建和管理的密钥，请在导航窗格中选择 Customer managed keys (客户托管密钥)。

AWS KMS 支持 AWS CloudTrail，因此您可以审核密钥使用情况，以验证密钥的使用是否正确。您的 AWS KMS 密钥可以与亚马逊 RDS、Amazon S3、Amazon Redshift AWS DMS 和亚马逊 EBS 等支持的 AWS 服务结合使用。

您还可以创建专门用于加密以下 AWS DMS 终端节点的目标数据的自定义 AWS KMS 密钥：

- Amazon Redshift – 有关更多信息，请参阅 [创建 AWS KMS 密钥并使用该密钥对 Amazon Redshift 目标数据进行加密](#)。
- Amazon S3 – 有关更多信息，请参阅 [创建 AWS KMS 密钥以加密 Amazon S3 目标对象](#)。

使用 KMS 密钥创建 AWS DMS 资源后，您无法更改这些资源的加密密钥。在创建 AWS DMS 资源之前，请务必确定您的加密密钥要求。

## 的网络安全 AWS Database Migration Service

您在使用时创建的网络的安全要求 AWS Database Migration Service 取决于您的网络配置方式。网络安全的一般规则 AWS DMS 如下：

- 复制实例必须具有对源和目标终端节点的访问权限。复制实例的安全组必须具有网络 ACL 或规则来允许从实例数据库端口上传出到数据库终端节点。
- 数据库终端节点必须包括网络 ACL 和安全组规则，允许来自复制实例的传入访问。根据配置，您可以使用复制实例的安全组、私有 IP 地址、公有 IP 地址或 NAT 网关的公有地址来实现这一点。
- 如果您的网络使用 VPN 隧道，则充当 NAT 网关的 Amazon EC2 实例必须使用具有规则的安全组，该规则允许复制实例通过它发送流量。

默认情况下，AWS DMS 复制实例使用的 VPC 安全组的规则允许所有端口的出口 0.0.0.0/0。如果您修改此安全组或者使用自己的安全组，则必须至少允许传出到相应数据库端口上的源和目标终端节点。

您可用于数据库迁移的网络配置均要求有特定安全注意事项：

- [一个 VPC 中具有所有数据库迁移组件的配置](#) – 端点使用的安全组必须允许从复制实例向数据库端口上的传入。确保由复制实例使用的安全组已传入到终端节点，或者您可以在由终端节点使用的安全组中创建规则，允许访问复制实例的私有 IP 地址。
- [多 VPC 配置](#) – 复制实例使用的安全组规则必须具有针对数据库上的 VPC 范围和数据库端口的规则。
- [使用 AWS Direct Connect 或 VPN 配置到 VPC 的网络](#) – VPN 隧道，允许流量从 VPC 通过隧道流向本地 VPN。在此配置中，VPC 包含路由规则，会将以特定 IP 地址或范围为目标流量发送到主机，该主机可以桥接从 VPC 到本地 VPN 的流量。如果是这种情况，NAT 主机包括自己的安全组设置，必须允许从复制实例私有 IP 地址或安全组到 NAT 实例的流量。
- [使用 Internet 连接 VPC 的网络配置](#) – VPC 安全组必须包括路由规则，将并非以 VPC 为目标的流量发送到互联网网关。在此配置中，与终端节点连接显示为来自复制实例上的公有 IP 地址。
- [使用不在 VPC 中的 RDS 数据库实例与 VPC 中的数据库实例进行配置 ClassicLink](#) – 当源或目标 Amazon RDS 数据库实例不在 VPC 中，并且不与复制实例所在的 VPC 共享安全组时，您可以设置代理服务器并使用代理服务器 ClassicLink 来连接源数据库和目标数据库。
- 源端点位于复制实例所使用的 VPC 的外部并使用 NAT 网关 – 您可以使用绑定到单个弹性网络接口的单个弹性 IP 地址来配置网络地址转换 (NAT) 网关。之后，该弹性网络接口会收到一个 NAT 标识符 (nat-#####)。如果 VPC 包含到 NAT 网关而不是 Internet 网关的默认路由，复制实例将改为显示使用 Internet 网关的公有 IP 地址连接数据库终端节点。在这种情况下，对 VPC 外部的数据库终端节点的传入需要允许从 NAT 地址的传入，而不是复制实例的公有 IP 地址。

- 适用于非 RDBMS 引擎的 VPC 端点 – AWS DMS 不支持适用于非 RDBMS 引擎的 VPC 端点。

# 将 SSL 与 AWS Database Migration Service

您可以使用安全套接字层 (SSL) 为源和目标终端节点的连接加密。为此，您可以使用 AWS DMS 管理控制台或 AWS DMS API 为端点分配证书。您也可以使用 AWS DMS 控制台来管理您的证书。

并非所有数据库都以相同的方式使用 SSL。Amazon Aurora MySQL 兼容版使用服务器名称 ( 集群中主实例的端点 ) 作为 SSL 的端点。Amazon Redshift 终端节点已使用 SSL 连接，并且不需要由 AWS DMS 设置 SSL 连接。Oracle 终端节点需要其他步骤；有关更多信息，请参阅 [Oracle 终端节点的 SSL 支持](#)。

## 主题

- [将 SSL 与 AWS DMS 配合使用的限制](#)
- [管理证书](#)
- [为 MySQL 兼容、PostgreSQL 或 SQL Server 终端节点启用 SSL](#)

要将证书分配到某个终端节点，您需要提供对部署到终端节点的服务器 SSL 证书进行签名时使用的根证书或者直至根的中间 CA 证书链 (作为证书捆绑包)。只接受 PEM 格式的 X509 文件的证书。在导入证书时，您收到可用于为终端节点指定证书的 Amazon 资源名称 (ARN)。如果使用 Amazon RDS，您可以下载由 Amazon RDS 托管的 `rds-combined-ca-bundle.pem` 文件中提供的根 CA 和证书捆绑包。有关下载此文件的更多信息，请参阅《Amazon RDS 用户指南》中的 [使用 SSL/TLS 来加密与数据库实例的连接](#)。

您可从多个 SSL 模式中选择用于 SSL 证书验证的模式。

- none – 不加密连接。该选项不安全，但需要的开销较少。
- require – 使用 SSL ( TLS ) 加密连接，但不进行任何 CA 验证。该选项更为安全，但需要的开销较多。
- verify-ca – 加密连接。该选项更为安全，但需要的开销较多。该选项将验证服务器证书。
- verify-full – 加密连接。该选项更为安全，但需要的开销较多。该选项验证服务器证书并验证服务器主机名与证书的主机名属性匹配。

并非所有 SSL 模式均适用于所有数据库终端节点。下表显示了各个数据库引擎支持哪些 SSL 模式。

数据库引擎	none	require	verify-ca	verify-full
MySQL/MariaDB/ Amazon Aurora MySQL	默认	不支持	支持	支持
Microsoft SQL Server	默认	支持	不支持	支持
PostgreSQL	默认	支持	支持	支持
Amazon Redshift	默认	SSL 未启用	SSL 未启用	SSL 未启用
Oracle	默认	不支持	支持	不支持
SAP ASE	默认	SSL 未启用	SSL 未启用	支持
MongoDB	默认	支持	不支持	支持
Db2 LUW	默认	不支持	支持	不支持
Db2 for z/OS	默认	不支持	支持	不支持

### Note

DMS 控制台或 API 上的 SSL 模式选项不适用于某些数据流和 NoSQL 服务，如 Kinesis 和 DynamoDB。默认情况下，它们是安全的，因此 DMS 显示的 SSL 模式设置为“无”（SSL 模式=无）。您无需为端点提供任何配置即可使用 SSL。例如，使用 Kinesis 作为目标端点时，默认情况下它是安全的。所有对 Kinesis 的 API 调用都使用 SSL，因此无需在 DMS 端点中添加额外的 SSL 选项。您可以使用 HTTPS 协议，通过 SSL 端点安全地存放数据和检索数据，DMS 在连接 Kinesis Data Stream 时默认使用该协议。

## 将 SSL 与 AWS DMS 配合使用的限制

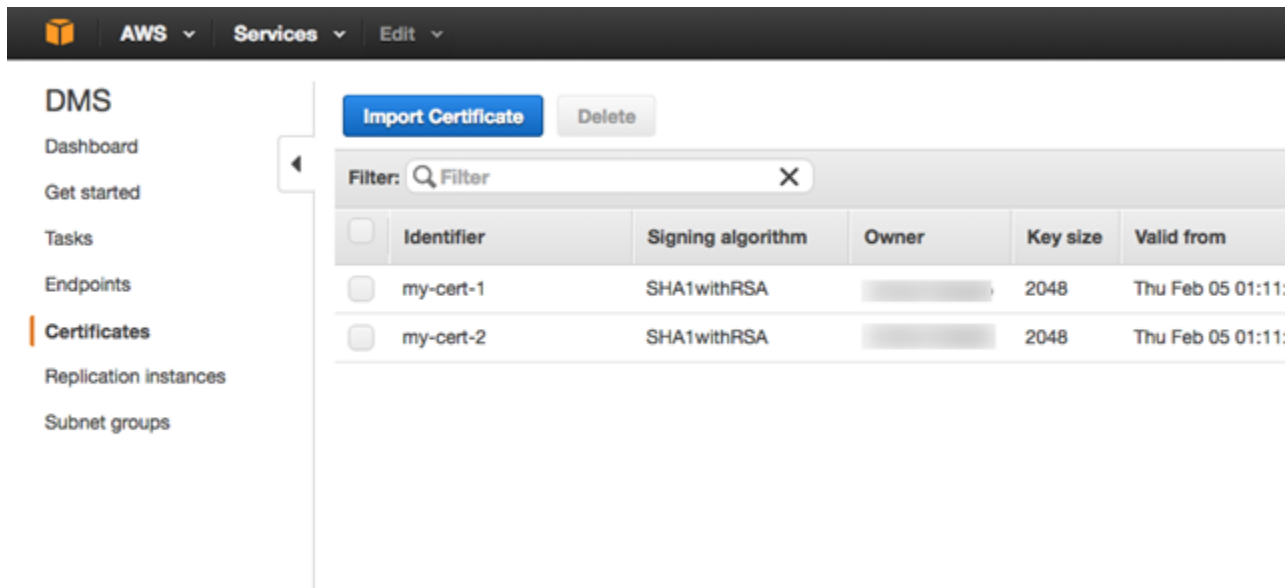
以下是使用 SSL 的限制 AWS DMS：

- 不支持与亚马逊 Redshift 目标终端节点的 SSL 连接。AWS DMS 使用亚马逊 S3 存储桶将数据传输到亚马逊 Redshift 数据库。此传输默认情况下由 Amazon Redshift 加密。

- 使用启用了 SSL 的 Oracle 终端节点执行更改数据捕获 (CDC) 任务时，可能会出现 SQL 超时。如果您遇到 CDC 计数器未体现预期数字的问题，请在任务设置的 ChangeProcessingTuning 部分将 MinimumTransactionSize 参数设置为较低的值。您可以从最低值 100 开始。有关 MinimumTransactionSize 参数的更多信息，请参阅 [更改处理优化设置](#)。
- 您只能以 .pem 和 .sso (Oracle wallet) 格式导入证书。
- 在某些情况下，您的服务器 SSL 证书可能由中间证书颁发机构 (CA) 签名。此种情况下，请确保从中间 CA 到根 CA 的整个证书链作为单个 .pem 文件导入。
- 如果在服务器上使用自签名证书，请选择 require 以作为 SSL 模式。require SSL 模式隐式信任服务器的 SSL 证书，而不会尝试验证证书是否由 CA 签名。

## 管理证书

您可以使用 DMS 控制台来查看和管理 SSL 证书。您也可以使用 DMS 控制台导入证书。



## 为 MySQL 兼容、PostgreSQL 或 SQL Server 终端节点启用 SSL


您可以添加指向新创建终端节点或现有终端节点的 SSL 连接。

使用 SSL 创建 AWS DMS 终端节点

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。

如果您以 AWS Identity and Access Management (IAM) 用户身份登录，请确保您拥有相应的访问权限 AWS DMS。有关数据库迁移所需权限的更多信息，请参阅 [使用 AWS DMS 所需的 IAM 权限](#)。

2. 在导航窗格中，选择证书。
3. 选择导入证书。
4. 将要用于加密连接的证书上传到终端节点。

 Note

您还可以在创建或修改端点时使用 AWS DMS 控制台上传证书，方法是在创建数据库终端节点页面上选择添加新 CA 证书。

要将 Aurora Serverless 作为目标，请获取在[将 TLS/SSL 与 Aurora Serverless 结合使用](#)中提到的证书。


5. 如[步骤 2：指定源和目标端点](#)中所述创建终端节点

### 修改现有 AWS DMS 终端节点以使用 SSL

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。

如果以 IAM 用户身份登录，请确保具有适当的 AWS DMS 访问权限。有关数据库迁移所需权限的更多信息，请参阅 [使用 AWS DMS 所需的 IAM 权限](#)。

2. 在导航窗格中，选择证书。
3. 选择导入证书。
4. 将要用于加密连接的证书上传到终端节点。

 Note

您还可以在创建或修改端点时使用 AWS DMS 控制台上传证书，方法是在创建数据库终端节点页面上选择添加新 CA 证书。

5. 在导航窗格中选择终端节点，选择要修改的终端节点，然后选择修改。
6. 为 SSL mode (SSL 模式) 选择一个值。



如果选择 `verify-ca` 或 `verify-full` 模式，则需指定要用于 CA certificate (CA 证书) 的证书，如下所示。

## 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. 选择修改。
8. 在修改终端节点后，选择终端节点，然后选择 Test connection (测试连接) 以确定 SSL 连接是否正常工作。

创建源和目标终端节点之后，创建使用这些终端节点的任务。有关创建任务的更多信息，请参阅[步骤 3：创建任务并迁移数据](#)。

## 更改数据库密码

在大多数情况下，更改源或目标终端节点的数据库密码非常直接。如果您需要更改当前在迁移或复制任务中使用的终端节点的数据库密码，则该过程需要额外的几个步骤。以下过程演示了如何完成此操作。

### 更改迁移或复制任务中的终端节点的数据库密码

1. 登录 AWS Management Console 并打开 AWS DMS 控制台，[网址为 https://console.aws.amazon.com/dms/v2/](https://console.aws.amazon.com/dms/v2/)。

如果以 IAM 用户身份登录，请确保具有适当的 AWS DMS 访问权限。有关所需权限的更多信息，请参阅[使用 AWS DMS 所需的 IAM 权限](#)。

2. 在导航窗格中，选择数据库迁移任务。
3. 选择使用要更改数据库密码的终端节点的任务，然后选择停止。
4. 任务停止时，您可以使用数据库中的本机工具来更改终端节点的数据库密码。
5. 返回到 DMS 管理控制台，然后从导航窗格中选择终端节点。
6. 选择已更改密码的数据库的终端节点，然后选择修改。
7. 在密码框中键入新密码，然后选择保存。
8. 从导航窗格中选择数据库迁移任务。
9. 选择之前停止的任务，然后选择重新启动/恢复。
10. 根据您的希望如何继续执行该任务，选择重新启动或恢复，然后选择启动任务。

## AWS Database Migration Service 的限额

接下来，您可以找到 AWS Database Migration Service (AWS DMS) 的资源配额和命名约束。

AWS DMS 可以迁移的数据库的最大大小取决于多种因素。这些因素包括源环境、源数据库中数据的分布以及源系统的繁忙程度。

确定特定系统是否适合作为 AWS DMS 的候选的最佳方式就是进行测试。一开始不要着急，这样您就可以让配置正常工作，然后添加一些复杂的对象。最后，尝试进行完全加载来测试。

## AWS Database Migration Service 资源配额

每个 AWS 区域的每个 AWS 账户都有关于可以创建的 AWS DMS 资源数量的配额。达到某一资源的配额时，再进行创建该资源的调用就会失败并引发异常。

下表列出了每个 AWS 区域的 AWS DMS 资源及其配额。

资源	默认限额
API 请求节流	每秒最多 200 个请求
API 请求刷新速率	每秒 8 个请求
每个用户账户的复制实例数	60
复制实例的存储总量	30000 GB
每个用户账户的事件订阅数	60
每个用户账户的复制子网组数	60
每个复制子网组的子网	60
每个用户账户的端点数	1000
每个复制实例的端点数	100
每个用户账户的任务数	600
每个复制实例的任务数	200

资源	默认限额
每个用户账户的证书数	100
每个用户账户的数据提供程序数	1000
每个用户账户的实例配置文件数	60
每个用户账户的迁移项目数	10
每个用户账户的 DMS 数据收集器数	10
一次性生成的目标建议	100
DMS 数据收集器每小时可以上传的文件数	500
每个用户账户的同构数据迁移数	600
一次性运行的同构数据迁移数	100
每个迁移项目的同构数据迁移数	10
无服务器复制	100

有关 API 请求节流配额和刷新速率的更多信息，请参阅[了解 API 请求节流](#)。

30000 GB 的存储配额应用到您在给定 AWS 区域中的所有 AWS DMS 复制实例。此存储用于在目标无法与源保持同步时缓存更改，以及存储日志信息。

## 了解 API 请求节流

AWS DMS 支持的请求速率可变，但最大 API 请求配额为每秒 200 个 API 调用。换言之，当您的 API 请求超过此速率时就会受到限制。此外，您可以将每秒 API 调用次数限制为较小的值，具体取决于 AWS DMS 在您发出另一个 API 请求之前刷新您的配额所需的时间。此配额应用到您直接发出 API 调用，以及在使用 AWS DMS 管理控制台时代表您进行的调用。

要了解 API 请求节流的工作原理，可以想象一下，AWS DMS 维护一个用于跟踪您的 API 请求的令牌存储桶。在这种情况下，存储桶中的每个令牌都允许您发出一个 API 调用。在任何时候，您的存储桶中最多只能有 200 个令牌。当您发出 API 调用时，AWS DMS 会从存储桶中删除一个令牌。如果您在不到一秒钟的时间内发出了 200 个 API 调用，则您的存储桶被清空，任何发出另一次 API 调用的尝

试都将失败。在您没有进行 API 调用的每一秒中，AWS DMS 向存储桶中添加 8 个令牌，最多不超过 200 个令牌。这是 AWS DMS API 请求刷新速率。在节流后的任何时候，当您的存储桶中添加了令牌时，您就可以进行 API 调用，其数量与可用令牌数相同，直到您的调用再次受到限制。

如果您使用 AWS CLI 来运行 API 调用时受限，则 AWS DMS 会返回如下错误：

```
An error occurred (ThrottlingException) when calling the AwsDmsApiCall operation (reached max retries: 2): Rate exceeded
```

这里 *AwsDmsApiCall* 是被限制的 AWS DMS API 操作的名称，例如 `DescribeTableStatistics`。然后，您可以在延迟足够长的时间后，重试调用或发出另一次调用以避免节流。

#### Note

与某些其他服务（例如 Amazon EC2）管理的 API 请求节流不同，您不能命令增加由 AWS DMS 管理的 API 请求节流配额。

# 排除 AWS Database Migration Service 中的迁移任务问题

在下文中，您可以找到有关对使用 AWS Database Migration Service (AWS DMS) 时的问题进行故障排除的主题。这些主题可以帮助您解决使用 AWS DMS 和所选端点数据库的常见问题。

如果您已提交 AWS Support 案例，则您的支持工程师可能会确定您的某个端点数据库配置存在潜在问题。工程师可能还会要求您运行支持脚本以返回有关数据库的诊断信息。有关从此类支持脚本下载、运行和上传诊断信息的详细信息，请参阅[使用 AWS DMS 中的诊断支持脚本](#)。

出于故障排除目的，在复制实例中 AWS DMS 收集跟踪文件并转储文件。如果出现需要故障排除的问题，AWS 您可以将这些文件提供给 Support。默认情况下，DMS 会清除超过三十天的跟踪和转储文件。要选择退出跟踪和转储文件收集，请向 Support AWS 提交案例。

## 主题

- [迁移任务运行缓慢](#)
- [任务状态栏没有进展](#)
- [任务已完成但未迁移任何内容](#)
- [缺少外键和辅助索引](#)
- [AWS DMS 不创建 CloudWatch 日志](#)
- [连接 Amazon RDS 时出现问题](#)
- [出现网络问题](#)
- [CDC 在完全加载后粘滞](#)
- [重新启动任务时出现主键冲突错误](#)
- [架构初始加载失败](#)
- [任务失败，出现未知错误](#)
- [任务重新启动操作从头开始加载表](#)
- [每个任务的表数量导致问题](#)
- [在 LOB 列上创建主键时，任务失败](#)
- [不带主键的目标表上出现重复的记录](#)
- [预留 IP 范围内的源端点失败](#)
- [Amazon Athena 查询中的时间戳是乱码](#)
- [排除 Oracle 中的问题](#)

- [排除 MySQL 中的问题](#)
- [排除 PostgreSQL 中的问题](#)
- [排除 Microsoft SQL Server 中的问题](#)
- [排除 Amazon Redshift 中的问题](#)
- [排除 Amazon Aurora MySQL 中的问题](#)
- [排除 SAP ASE 的问题](#)
- [排除 IBM Db2 中的问题](#)
- [排除 AWS Database Migration Service 中的延迟问题](#)
- [使用 AWS DMS 中的诊断支持脚本](#)
- [使用 AWS DMS 诊断支持 AMI](#)

## 迁移任务运行缓慢

多个问题可能导致迁移任务运行缓慢，或导致后续任务的运行速度慢于初始任务的运行速度。

导致迁移任务运行缓慢的最常见原因是，分配给 AWS DMS 复制实例的资源不足。为确保您的实例拥有足够的资源来运行任务，请检查复制实例的 CPU、内存、交换文件和 IOPS 的使用情况。例如，多个将 Amazon Redshift 作为端点的任务是 I/O 密集型的。您可以增加复制实例的 IOPS 或跨多个复制实例拆分任务以实现更高效的迁移。

有关如何确定复制实例的大小的更多信息，请参阅[选择复制实例的最佳大小](#)。

您可以执行以下操作来提高初始迁移加载的速度：

- 如果您的目标是 Amazon RDS 数据库实例，请确保没有为目标数据库实例启用多可用区。
- 在加载期间，请关闭目标数据库上的任何自动备份或日志记录功能，并在迁移完成后重新启用这些功能。
- 如果该功能在目标上可用，请使用预调配 IOPS。
- 如果迁移数据包含 LOB，请确保已针对 LOB 迁移优化任务。有关针对 LOB 进行优化的更多信息，请参阅[目标元数据任务设置](#)。

## 任务状态栏没有进展

任务状态栏提供了任务进度的估计值。此估计值的质量取决于源数据库的表统计数据的质量；表统计数据的质量越好，估计值就越准确。

对于只有一个表而且该表没有估算的行统计数据的任务，AWS DMS 无法提供任何类型的完成百分比估计值。在此情况下，可使用任务状态和行加载指示来确认任务是否正在运行且取得进展。

## 任务已完成但未迁移任何内容

如果任务完成后未迁移任何内容，请执行以下操作。

- 检查创建端点的用户对要迁移的表是否具有读取权限。
- 检查要迁移的对象是否为表。如果是视图，则更新表映射并将 object-locator 指定为“view”或“all”。有关更多信息，请参见 [通过控制台指定表选择和转换规则](#)。

## 缺少外键和辅助索引

AWS DMS 会创建表、主键和（在某些情况下）唯一索引，但不会创建高效迁移源中数据时不需要的任何其他对象。例如，它不会创建二级索引、非主键约束或数据默认值。

要迁移数据库中的二级对象，请使用数据库的本机工具（如果您将迁移到与源数据库相同的数据库引擎）。如果您要迁移到的数据库引擎不同于源数据库用来迁移二级对象的数据库引擎，请使用 AWS Schema Conversion Tool (AWS SCT)。

## AWS DMS不创建 CloudWatch 日志

如果您的复制任务未创建 CloudWatch 日志，请确保您的账户具有该 dms-cloudwatch-logs-role 角色。如果此角色不存在，请执行以下操作以创建：

1. 登录到 AWS Management Console，然后通过以下网址打开 IAM 控制台：<https://console.aws.amazon.com/iam/>。
2. 选择角色选项卡。选择创建角色 )。
3. 在选择受信任实体的类型部分中，选择 AWS 服务。
4. 在选择使用案例部分中，选择 DMS。
5. 选择下一步: 权限。
6. 在搜索栏 AmazonDMS CloudWatchLogsRole 中输入，然后选中 AmazonDMS CloudWatchLogsRole MS 旁边的复选框。这会授 AWS DMS 予访问权限 CloudWatch。
7. 选择下一步：标签。
8. 选择下一步：审核。



9. 对于角色名称，输入 **dms-cloudwatch-logs-role**。此名称区分大小写。
10. 选择创建角色。

## 连接 Amazon RDS 时出现问题

有多个原因会导致您无法连接到设置为源或目标的 Amazon RDS 数据库实例。以下是一些需要检查的项目：

- 检查用户名和密码组合是否正确。
- 检查 Amazon RDS 控制台中为实例显示的端点值是否与用于创建 AWS DMS 端点的端点标识符相同。
- 检查 Amazon RDS 控制台中为实例显示的端口值是否与分配给 AWS DMS 端点的端口相同。
- 检查分配给 Amazon RDS 数据库实例的安全组是否允许来自 AWS DMS 复制实例的连接。
- 如果 AWS DMS 复制实例和 Amazon RDS 数据库实例不在同一个虚拟私有云 (VPC) 中，请检查数据库实例是否可公开访问。

### 错误消息：线程连接字符串不正确：线程值 0 不正确

在测试与终端节点的连接时通常会出现此错误。此错误表示连接字符串出错。例如，主机 IP 地址后面有一个空格。另一种可能是在连接字符串中复制了错误字符。

## 出现网络问题

最常见的联网问题涉及 AWS DMS 复制实例所使用的 VPC 安全组。默认情况下，此安全组中的规则允许在所有端口上进入 0.0.0.0/0。在许多情况下，您可以修改此安全组或使用自己的安全组。如果这样做，请确保在源端点和目标端点各自的数据库端口上向其提供传出。

其他与配置相关的问题可能包括：

- 复制实例以及源和目标端点位于同一 VPC 中 – 端点所使用的安全组必须允许复制实例通过数据库端口传入。确保复制实例使用的安全组可以传入端点。或者您可以在由端点使用的安全组中创建规则，允许复制实例的私有 IP 地址访问。
- 源端点位于复制实例所使用的 VPC 的外部（使用互联网网关）– VPC 安全组必须包含将未指定发往 VPC 的流量发送到互联网网关的路由规则。在此配置中，与终端节点的连接显示为来自复制实例上的公有 IP 地址。

- 源端点位于复制实例所使用的 VPC 的外部 ( 使用 NAT 网关 ) – 您可以使用绑定到单个弹性网络接口的单个弹性 IP 地址来配置网络地址转换 (NAT) 网关。此 NAT 网关接收 NAT 标识符 (nat-#####)。

在某些情况下，VPC 包含指向该 NAT 网关的默认路由，而不是互联网网关。在这种情况下，复制实例改为使用 NAT 网关的公有 IP 地址联系数据库端点。此时，对 VPC 外部的数据库端点的传入需要允许从 NAT 地址的传入，而不是复制实例的公有 IP 地址。

有关如何使用您自己的本地名称服务器的信息，请参阅 [使用您自己的本地名称服务器](#)。

## CDC 在完全加载后粘滞

在多个 AWS DMS 设置相互冲突的情况下，复制更改在完全加载迁移后将减慢或粘滞。

例如，假设目标表准备模式参数设置为不执行任何操作或截断。在这种情况下，您指示 AWS DMS 不要对目标表进行任何设置，包括创建主索引和唯一索引。如果您尚未在目标表上创建主键或唯一键，则 AWS DMS 必须对每个更新执行全表扫描。此方法将大大影响性能。

## 重新启动任务时出现主键冲突错误

当数据保留在上一个迁移任务中的目标数据库中时，会出现此错误。如果目标表准备模式选项设置为不执行任何操作，则 AWS DMS 不会在目标表上进行任何准备，包括清理从上一任务中插入的数据。

要重新启动任务并避免出现这些错误，请删除上次运行任务时插入目标表中的行。

## 架构初始加载失败

在某些情况下，架构的初始加载可能会失败，出现错误

```
Operation:getSchemaListDetails:errType=, status=0, errMessage=, errDetails=.
```

在这种情况下，AWS DMS 用来连接到源端点的用户账户没有必要的权限。

## 任务失败，出现未知错误

可能会因为各种各样的原因导致出现未知类型错误。但是，我们经常发现问题涉及分配给 AWS DMS 复制实例的资源不足。

为确保您的复制实例拥有足够的资源来执行迁移，请检查实例的 CPU、内存、交换文件和 IOPS 的使用情况。有关监控的更多信息，请参阅[AWS Database Migration Service 指标](#)。

## 任务重新启动操作从头开始加载表

AWS DMS 在未完成表的初始加载时从头开始重新加载表。当任务重新启动时，AWS DMS 在初始加载未完成时从头开始重新加载表。

## 每个任务的表数量导致问题

每个复制任务上的表数量没有设置限制。但是，根据经验，我们建议将一个任务中的表数量限制在 60000 以下。当单个任务使用 60000 个以上的表时，资源使用通常是一个瓶颈。

## 在 LOB 列上创建主键时，任务失败

在 FULL LOB 或 LIMITED LOB 模式下，AWS DMS 不支持 LOB 数据类型的主键。

DMS 最初将具有 LOB 列的行迁移为 null，然后更新 LOB 列。因此，当在 LOB 列上创建主键时，初始插入失败，因为主键不能为 null。解决方法是添加另一列作为主键，然后从 LOB 列中删除主键。

## 不带主键的目标表上出现重复的记录

运行完全加载和 CDC 任务会在没有主键或唯一索引的目标表上创建重复的记录。为避免在完全加载和 CDC 任务期间在目标表上出现重复的记录，请确保目标表具有主键或唯一索引。

## 预留 IP 范围内的源端点失败

如果 AWS DMS 源数据库使用的 IP 地址位于 192.168.0.0/24 的预留 IP 范围内，则源端点连接测试将失败。以下步骤提供了一个可能的解决方法：

1. 查找一个不在预留范围内且可以与 192.168.0.0/24 中的源数据库通信的 Amazon EC2 实例。
2. 安装 socat 代理并运行它。下面是一个示例。

```
yum install socat

socat -d -d -lmlocal2 tcp4-listen:database_port,bind=0.0.0.0,reuseaddr,fork
tcp4:source_database_ip_address:database_port
&
```

将 Amazon EC2 实例 IP 地址和上面给出的数据库端口用于 AWS DMS 端点。确保端点具有允许 AWS DMS 访问数据库端口的安全组。请注意，代理需要在 DMS 任务执行期间运行。根据使用案例，您可能需要自动设置代理。

## Amazon Athena 查询中的时间戳是乱码

如果 Athena 查询中的时间戳出现乱码，请使用 AWS Management Console 或 [ModifyEndpoint](#) 操作将 Amazon S3 终端节点的值 `parquetTimestampInMillisecond` 设置为 `true`。有关更多信息，请参阅 [S3Settings](#)。

## 排除 Oracle 中的问题

接下来，您可以了解如何排除将 AWS DMS 与 Oracle 数据库一起使用时的特定问题。

### 主题

- [从视图中提取数据](#)
- [从 Oracle 12c 迁移 LOB](#)
- [在甲骨文 LogMiner 和二进制阅读器之间切换](#)
- [错误：Oracle CDC 已停止 122301 Oracle CDC 已超出最大重试计数器。](#)
- [自动将补充日志记录添加到 Oracle 源端点](#)
- [未捕获 LOB 更改](#)
- [错误：ORA-12899: 值对于列 column-name 过大](#)
- [未正确解释 NUMBER 数据类型](#)
- [完全加载期间缺少记录](#)
- [表错误](#)
- [错误：无法检索 Oracle 存档的重做日志目标 ID](#)
- [评估 Oracle 重做或存档日志的读取性能](#)

## 从视图中提取数据

您可以从视图中提取一次数据；您无法将其用于持续复制。要能够从视图中提取数据，您必须将以下代码添加到 Oracle 源端点页面的端点设置部分中。从视图中提取数据时，此视图将显示为目标架构上的表。

```
"ExposeViews": true
```

## 从 Oracle 12c 迁移 LOB

AWS DMS 可以使用两种方法来捕获 Oracle 数据库的更改，即二进制读取器和 Oracle LogMiner。默认情况下，AWS DMS 使用 Oracle LogMiner 来捕获更改。但是，在 Oracle 12c 上，甲骨文 LogMiner 不支持 LOB 列。要捕获对 Oracle 12c 上的 LOB 列进行的更改，请使用 Binary Reader。

### 在甲骨文 LogMiner 和二进制阅读器之间切换

AWS DMS 可以使用两种方法来捕获源 Oracle 数据库的更改，即 Binary Reader 和 Oracle LogMiner。甲骨文 LogMiner 是默认的。要切换到使用 Binary Reader 捕获更改，请执行以下操作：

使用 Binary Reader 捕获更改

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 选择端点。
3. 选择要使用 Binary Reader 的 Oracle 源端点。
4. 选择修改。
5. 选择高级，然后为额外的连接属性添加以下代码：

```
useLogminerReader=N
```

6. 使用 Oracle 开发人员工具（例如 SQL-Plus）将以下附加权限授予用于连接到 Oracle 端点的 AWS DMS 用户账户：

```
SELECT ON V_$TRANSPORTABLE_PLATFORM
```

## 错误：Oracle CDC 已停止 122301 Oracle CDC 已超出最大重试计数器。

如果在 AWS DMS 能够使用所需的 Oracle 存档日志捕获更改之前从服务器中删除该日志，则会出现此错误。增加数据库服务器上的日志保留策略。对于 Amazon RDS 数据库，请运行以下过程来增加日志保留期。例如，以下代码将 Amazon RDS 数据库实例上的日志保留期增至 24 小时。

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
```

## 自动将补充日志记录添加到 Oracle 源端点

默认情况下，AWS DMS 已禁用补充日志记录。要自动为源 Oracle 终端节点启用补充日志记录，请执行以下操作：

将补充日志记录添加到源 Oracle 终端节点

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 选择端点。
3. 选择要将补充日志记录添加到的 Oracle 源端点。
4. 选择 Modify(修改)。
5. 选择高级，然后将以下代码添加到额外的连接属性文本框中：

```
addSupplementalLogging=Y
```

6. 选择修改。

## 未捕获 LOB 更改

当前，表必须具有 AWS DMS 的主键才能捕获 LOB 更改。如果包含 LOB 的表不具有主键，您可以执行多个操作来捕获 LOB 更改：

- 向表添加主键。此操作与添加 ID 列并使用触发器向该列填充序列一样简单。
- 创建将系统生成的 ID 包含为主键的表的实体化视图，并迁移该实体化视图而非表。

- 创建逻辑备用，向表添加主键并从逻辑备用迁移。

## 错误：ORA-12899: 值对于列 *column-name* 过大

错误“ORA-12899：值对于列 *column-name* 太大”通常是由几个问题引起的。

其中一个问题是源数据库和目标数据库使用的字符集不匹配。

另一个问题是两个数据库之间的国家语言支持 (NLS) 设置不同。通常，当源数据库 NLS\_LENGTH\_SEMANTICS 参数设置为 CHAR 而目标数据库 NLS\_LENGTH\_SEMANTICS 参数设置为 BYTE 时，会出现此错误。

## 未正确解释 NUMBER 数据类型

Oracle NUMBER 数据类型可转换为各种 AWS DMS 数据类型，具体取决于 NUMBER 的精度和小数位数。此处记录了这些转换：[Oracle 的源数据类型](#)。使用源 Oracle 端点上的端点设置，也会影响 NUMBER 类型的转换方式。这些端点设置记录在[使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)中。

## 完全加载期间缺少记录

执行完全加载时，AWS DMS 会在数据库级别查找未完成的事务，然后等待事务提交。例如，根据任务设置 TransactionConsistencyTimeout=600，AWS DMS 会等待 10 分钟，即使未完成的事务位于未包含在表映射中的表上。但是，如果未完成的事务位于表映射中包含的表上，并且该事务未及时提交，则会导致目标表中缺少记录。

如果您知道未完成的事务需要较长的时间才能提交，则可以修改 TransactionConsistencyTimeout 任务设置并延长等待时间。

另请注意，FailOnTransactionConsistencyBreached 任务设置的默认值为 false。这意味着 AWS DMS 继续应用其他事务，但会错过未完成的事务。如果您希望在未完成的事务没有及时关闭时任务失败，则可以 FailOnTransactionConsistencyBreached 将设置为 true。

## 表错误

如果 WHERE 子句未引用主键列，并且未对所有列使用补充日志，则 Table Error 会在复制期间出现在表统计信息中。

要修复此问题，请为所引用表的所有列启用补充日志记录。有关更多信息，请参见[设置补充日志记录](#)。

## 错误：无法检索 Oracle 存档的重做日志目标 ID

当您的 Oracle 源未生成任何存档日志或 V\$ARCHIVED\_LOG 为空时，就会发生此错误。您可以通过手动切换日志来解决错误。

对于 Amazon RDS 数据库，请运行以下过程来切换日志文件。switch\_logfile 过程没有任何参数。

```
exec rdsadmin.rdsadmin_util.switch_logfile;
```

对于自管理 Oracle 源数据库，请使用以下命令强制切换日志。

```
ALTER SYSTEM SWITCH LOGFILE ;
```

## 评估 Oracle 重做或存档日志的读取性能

如果您的 Oracle 源遇到性能问题，则可以评估 Oracle 重做或存档日志的读取性能，以寻找提高性能的方法。要测试重做或存档日志的读取性能，请使用 [AWS DMS 诊断亚马逊机器映像 \(AMI\)](#)。

您可以使用 AWS DMS 诊断 AMI 执行以下操作：

- 使用 bFile 方法评估重做日志文件的性能。
- 使用该 LogMiner 方法评估重做日志文件的性能。
- 使用 PL/SQL (dbms\_lob.read) 方法评估重做日志文件的性能。
- 使用单线程来评估 ASMFile 上的读取性能。
- 使用多线程来评估 ASMFile 上的读取性能。
- 使用 Direct OS Readfile() Windows 或 Pread64 Linux 函数来评估重做日志文件。

然后，您可以根据结果采取补救措施。

### 测试 Oracle 重做或归档日志文件的读取性能

1. 创建 AWS DMS 诊断 AMI Amazon EC2 实例并连接到该实例。

有关更多信息，请参阅 [使用 AWS DMS 诊断 AMI](#)。

2. 运行 awsreplperf 命令。



```
$ awsreplperf
```

该命令显示 AWS DMS Oracle 读取性能实用程序选项。

```
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. 从列表选择一个选项。
4. 输入以下数据库连接和存档日志信息。

```
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. 检查显示的输出以获取相关的读取性能信息。例如，以下显示了选择选项编号 2 “读取使用” 可能产生的输出 LogMiner。

```
Enter your choice>>2
Oracle user name: [system] >> * * *
Oracle password :
Oracle connection name : [orcllx] >> * * * 1521/porcl
Oracle event trace ? : [N] >>n
Full path to redo or archive log file: [] >>+EBSFRA/PORCL/ONLINELOG/group_11.1380.1101828345
Elapsed Time : 7044.83973 sec
Read speed in : 0.088575 MB/sec
LogMinerRead: counted 1198389 redo log rows, total undo / redo size : 655073562
```

6. 要退出该实用程序，请输入 0 (零)。

## 后续步骤

- 当结果显示读取速度低于可接受的阈值时，请在端点上运行 [Oracle 诊断支持脚本](#)，查看“等待时间”、“加载配置文件”和“IO 配置文件”部分。然后调整任何可能提高读取性能的异常配置。例如，如果您的重做日志文件最大为 2 GB，请尝试将 LOG\_BUFFER 增加到 200 MB 以帮助提高性能。
- 查看 [AWS DMS 最佳实践](#)，确保以最佳方式配置了您的 DMS 复制实例、任务和端点。

## 排除 MySQL 中的问题

接下来，您可以了解如何排除将 AWS DMS 与 MySQL 数据库一起使用时的特定问题。

### 主题

- [由于已禁用二进制日志记录，无法为 Amazon RDS 数据库实例终端节点完成 CDC 任务](#)
- [与目标 MySQL 实例的连接在任务期间中断](#)
- [将自动提交添加到与 MySQL 兼容的终端节点](#)
- [在与 MySQL 兼容的目标终端节点上禁用外键](#)
- [字符被替换为问号](#)
- [“错误事件”日志条目](#)
- [使用 MySQL 5.5 的更改数据捕获](#)
- [增加 Amazon RDS 数据库实例的二进制日志保留期](#)
- [日志消息：源数据库中的一些更改在应用于目标数据库时不起作用。](#)
- [错误：标识符太长](#)
- [错误：不受支持的字符集导致字段数据转换失败](#)
- [错误：代码页 1252 到 UTF8 \[120112\] 字段数据转换失败](#)
- [索引、外键或级联更新内容或删除内容未迁移](#)

## 由于已禁用二进制日志记录，无法为 Amazon RDS 数据库实例终端节点完成 CDC 任务

Amazon RDS 数据库实例会因禁用自动化备份而出现该问题。可通过将备份保留期设置为非零值来启用自动备份。

## 与目标 MySQL 实例的连接在任务期间中断

如果涉及 LOB 的任务与 MySQL 目标的连接断开，则您可能需要在任务日志中查看以下类型的错误。

```
[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.
```

在这种情况下，您可能需要调整某些任务设置。

要解决任务与 MySQL 目标的连接断开的问题，请执行以下操作：

- 检查数据库变量 `max_allowed_packet` 集是否足以保留您的最大 LOB。
- 检查您是否让以下变量集具有较大的超时值。建议您对其中每个变量使用的值至少为 5 分钟。
  - `net_read_timeout`
  - `net_write_timeout`
  - `wait_timeout`

有关设置 MySQL 系统变量的信息，请参阅 [MySQL 文档](#) 中的 [服务器系统变量](#)。

## 将自动提交添加到与 MySQL 兼容的终端节点

将自动提交添加到与 MySQL 兼容的目标终端节点

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 选择端点。
3. 选择要将自动提交添加到的与 MySQL 兼容的目标端点。
4. 选择修改。
5. 选择高级，然后将以下代码添加到额外的连接属性文本框中：

```
Initstmt= SET AUTOCOMMIT=1
```

6. 选择修改。

## 在与 MySQL 兼容的目标终端节点上禁用外键

在目标 MySQL、Amazon Aurora MySQL 兼容版或 MariaDB 端点的高级部分的额外的连接属性中，您可以添加以下内容以在 MySQL 上禁用外键检查。

在与 MySQL 兼容的目标终端节点上禁用外键

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS DMS 控制台：<https://console.aws.amazon.com/dms/v2/>。
2. 选择端点。
3. 选择要禁用外键的 MySQL、Aurora MySQL 或 MariaDB 目标端点。
4. 选择修改。
5. 选择高级，然后将以下代码添加到额外的连接属性文本框中：

```
Initstmt=SET FOREIGN_KEY_CHECKS=0
```

6. 选择修改。

## 字符被替换为问号

导致出现该问题的最常见情况是，当源终端节点字符使用了 AWS DMS 不支持的字符集进行编码时。

## “错误事件”日志条目

迁移日志中的“错误事件”条目通常指示对源数据库端点尝试执行了不受支持的数据定义语言 (DDL) 操作。不支持的 DDL 操作会导致复制实例无法跳过的事件，从而导致记录错误事件。

要修复此问题，请从头开始重新启动任务。这样做会重新加载表，并在发出不支持的 DDL 操作后的时间点开始捕获更改。

## 使用 MySQL 5.5 的更改数据捕获

Amazon RDS MySQL 兼容数据库的 AWS DMS 更改数据捕获 (CDC, Change Data Capture) 需要完整映像的基于行的二进制日志记录，此功能在 MySQL 版本 5.5 或早期版本中不受支持。要使用 AWS DMS CDC，必须将 Amazon RDS 数据库实例升级到 MySQL 版本 5.6。

### 增加 Amazon RDS 数据库实例的二进制日志保留期

AWS DMS 需要为更改数据捕获保留二进制日志文件。要增加 Amazon RDS 数据库实例上的日志保留期，请使用以下过程。以下示例将二进制日志保留期增至 24 小时。

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

**日志消息：**源数据库中的一些更改在应用于目标数据库时不起作用。

当 AWS DMS 将 MySQL 数据库列的值更新为其现有值时，将从 MySQL 返回消息 `zero rows affected`。这种行为与其他数据库引擎（例如 Oracle 和 SQL Server）不同。即使替换值与当前值相同，这些引擎也会更新一行。

### 错误：标识符太长

当标识符太长时，会出现以下错误：

```
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)
```

在某些情况下，您可以将 AWS DMS 设置为在目标数据库中创建表和主键。在这些情况下，DMS 当前为主键使用的名称，与源数据库中使用的主键名不同。相反，DMS 创建基于表名的主键名。当表名较长时，所创建的自动生成标识符的长度会超出 MySQL 允许的限制。

为了解决这个问题，当前的方法是先在目标数据库中预先创建表和主键。然后使用任务，将任务设置目标表准备模式设置为不执行任何操作或截断来填充目标表。

### 错误：不受支持的字符集导致字段数据转换失败

当不受支持的字符集导致字段数据转换失败时，会出现以下错误：

```
"[SOURCE_CAPTURE ]E: Column 'column-name' uses an unsupported character set [120112]
A field data conversion failed. (mysql_endpoint_capture.c:2154)
```

请检查与连接相关的数据库参数。以下命令可用于设置这些参数。

```
SHOW VARIABLES LIKE '%char%';
```

## 错误：代码页 1252 到 UTF8 [120112] 字段数据转换失败

如果源 MySQL 数据库中具有非代码页 1252 字符，则在迁移过程中可能会发生以下错误。

```
[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)
```

解决办法是：您可以对您的源 MySQL 终端节点使用 CharsetMapping 额外连接属性来指定字符集映射。如果您添加此端点设置，则可能需要从头开始重新启动 AWS DMS 迁移任务。

例如，以下端点设置可用于源字符集为 Utf8 或 latin1 的 MySQL 源端点。65001 是 UTF8 代码页标识符。

```
CharsetMapping=utf8,65001
CharsetMapping=latin1,65001
```

## 索引、外键或级联更新内容或删除内容未迁移

AWS DMS 不支持迁移次要对象，例如索引和外键。要复制通过级联更新或删除操作对子表所做的更改，您需要在目标表上激活触发外键约束。要解决此限制，请在目标表上手动创建外键。然后，为完全加载和 CDC 创建单个任务，或者为完全加载和 CDC 创建两个单独的任务，如下所述：

## 创建支持完全加载和 CDC 的单个任务

此过程介绍如何为完全加载和 CDC 使用单个任务来迁移外键和索引。

### 创建完全加载和 CDC 任务

1. 在目标上手动创建带有外键和索引的表，以匹配源表。
2. 将以下 ECA 添加到目标 AWS DMS 端点：

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. 创建 DO\_NOTHING 任务，将 TargetTablePrepMode 设置为 AWS DMS。
4. 将 Stop task after full load completes 设置为 StopTaskCachedChangesApplied。
5. 启动任务。AWS DMS 在完成完全加载后自动停止任务，并应用所有缓存的更改。
6. 移除您之前添加的 SET FOREIGN\_KEY\_CHECKS ECA。
7. 继续执行任务。任务进入 CDC 阶段，并将源数据库的持续更改应用到目标。

## 分别创建完全加载任务和 CDC 任务

这些过程介绍如何分别为完全加载和 CDC 使用单独的任务来迁移外键和索引。

### 创建完全加载任务

1. 在目标上手动创建带有外键和索引的表，以匹配源表。
2. 将以下 ECA 添加到目标 AWS DMS 端点：

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. 创建 AWS DMS 任务，将 TargetTablePrepMode 参数设置为 DO\_NOTHING，并将 EnableValidation 设置为 FALSE。
4. 启动任务。AWS DMS 在完成完全加载后自动停止任务，并应用所有缓存的更改。
5. 任务完成后，记下 UTC 格式的完全加载任务开始时间，或者二进制日志文件名和位置，以启动仅限 CDC 的任务。请参阅日志，获取从初始完全加载开始时间的戳，采用 UTC 格式。

## 创建仅限 CDC 的任务

1. 移除您之前设置的 SET FOREIGN\_KEY\_CHECKS ECA。
2. 创建仅 CDC 的任务，并将起始位置设置为上一步中记录的完全加载开始时间。或者，您可以使用上一步中记录的二进制日志位置。将 TargetTablePrepMode 设置为 DO\_NOTHING。如果需要，请将 EnableValidation 设置设为 TRUE 来启用数据验证。
3. 启动仅限 CDC 的任务，并监控日志中是否存在错误。

### Note

此解决方法仅适用于 MySQL 到 MySQL 的迁移。此方法不能与批量应用功能一起使用，因为批量应用要求目标表没有活动外键。

## 排除 PostgreSQL 中的问题

接下来，您可以了解如何排除将 AWS DMS 与 PostgreSQL 数据库一起使用时的特定问题。

### 主题

- [JSON 数据类型被截断](#)
- [用户定义的数据类型的列未正确迁移](#)
- [错误：未选择要在其中执行创建操作的架构](#)
- [未使用 CDC 复制针对表的删除内容和更新内容](#)
- [未传播截断语句](#)
- [阻止 PostgreSQL 捕获 DDL](#)
- [选择在其中创建用于捕获 DDL 的数据库对象的架构](#)
- [迁移到 PostgreSQL 后缺少 Oracle 表](#)
- [ReplicationSlotDiskUsage 在长时间交易（例如 ETL 工作负载）期间，增加并且 restart\\_lsn 停止向前移动](#)
- [使用视图作为源的任务未复制行](#)



## JSON 数据类型被截断

AWS DMS 将 PostgreSQL 中的 JSON 数据类型视为 LOB 数据类型列。这意味着，使用受限 LOB 模式时的 LOB 大小限制适用于 JSON 数据。

例如，假设有限的 LOB 模式设置为 4096 KB。在这种情况下，任何大于 4096 KB 的 JSON 数据都会在 4096 KB 的限制下被截断，无法通过 PostgreSQL 中的验证测试。

以下日志信息显示 JSON 由于受限 LOB 模式设置而截断，并且验证失败。

```
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)
```

## 用户定义的数据类型的列未正确迁移

在从 PostgreSQL 源复制时，AWS DMS 将为所有列（带用户定义的数据类型的列除外）创建带相同数据类型的目标表。在此类情况下，在目标中创建数据类型作为“可变字符”。

### 错误：未选择要在其中执行创建操作的架构

在某些情况下，您可能会看到错误“SQL\_ERROR SqlState : 3F000 : 7 消息 NativeError : 错误：未选择要在中创建的架构”。

当您的 JSON 表映射包含架构的通配符值但源数据库不支持该值时，可能会发生此错误。

## 未使用 CDC 复制针对表的删除内容和更新内容

如果源表没有主键，则更改数据捕获 (CDC) 期间的删除和更新操作将被忽略。AWS DMS 支持带有主键的 PostgreSQL 表的更改数据捕获 (CDC)。

如果某个表没有主键，提前写入日志 (WAL) 将不包含数据库行的前映像。在这种情况下，AWS DMS 无法更新表。对于要复制的删除操作，请在源表上创建主键。

## 未传播截断语句

在使用更改数据捕获 (CDC) 时，AWS DMS 不支持 TRUNCATE 操作。

## 阻止 PostgreSQL 捕获 DDL

您可以通过添加以下端点设置语句来阻止 PostgreSQL 目标端点捕获 DDL 语句。

```
"CaptureDDLs": "N"
```

## 选择在其中创建用于捕获 DDL 的数据库对象的架构

您可以控制在哪些架构中创建用于捕获 DDL 的数据库对象。添加以下端点设置语句。端点设置参数在源端点的选项卡中可用。

```
"DdlArtifactsSchema": "xyzddlschema"
```

## 迁移到 PostgreSQL 后缺少 Oracle 表

在这种情况下，您的表和数据通常仍然可以访问。

Oracle 默认为大写表名，而 PostgreSQL 默认为小写表名。当您执行从 Oracle 到 PostgreSQL 的迁移时，我们建议您在任务的表映射部分下提供特定转换规则。这些是转换表名大小写的转换规则。

如果您在未使用转换规则来转换表名的大小写的情况下迁移表，请在引用表名时为其添加引号。

## ReplicationSlotDiskUsage 在长时间交易（例如 ETL 工作负载）期间，增加并且 restart\_lsn 停止向前移动

启用逻辑复制后，每个事务在内存中保留的最大更改数为 4MB。之后，更改会溢出到磁盘。因此，ReplicationSlotDiskUsage 会增加，并且在事务完成/中止并且回滚完成之前，restart\_lsn 不会有进展。由于这是一个长时间运行的事务，可能需要花费较长时间回滚。

因此，在启用逻辑复制时，请避免长时间运行的事务。请改为尝试将事务分成几个较小的事务。

## 使用视图作为源的任务未复制行

要迁移视图，请将 table-type 设置为 all 或 view。有关更多信息，请参见 [通过控制台指定表选择 and 转换规则](#)。

以下来源支持视图。

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2 LUW
- SAP Adaptive Server Enterprise (ASE)

## 排除 Microsoft SQL Server 中的问题

接下来，您可以了解如何排除将 AWS DMS 与 Microsoft SQL Server 数据库一起使用时的特定问题。

主题

- [捕获 SQL Server 数据库的更改时出错](#)
- [缺少身份列](#)
- [错误：SQL Server 不支持发布](#)
- [更改未显示在目标中](#)
- [跨分区映射的不一致表](#)

## 捕获 SQL Server 数据库的更改时出错

更改数据捕获 (CDC) 期间出错通常指示未满足某个先决条件。例如，经常被忽视的先决条件是完整数据库备份。任务日志中会指出此忽略以及以下错误：

```
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)
```

在[使用微软 SQL Server 数据库作为来源 AWS DMS](#)中查看有关将 SQL Server 作为源的先决条件。

## 缺少身份列

在创建目标架构时，AWS DMS 不支持身份列。在初始加载完成后，必须添加这些列。

## 错误：SQL Server 不支持发布

在将 SQL Server Express 作为源终端节点时，会生成以下错误：

```
RetCode: SQL_ERROR SqlState: HY000 NativeError: 21106  
Message: This edition of SQL Server does not support publications.
```

AWS DMS 当前不支持将 SQL Server Express 作为源或目标。

## 更改未显示在目标中

AWS DMS 要求源 SQL Server 数据库处于“FULL”或“BULK LOGGED”数据恢复模式中以便一致地捕获更改。不支持“SIMPLE”模式。

SIMPLE 恢复模式记录允许用户恢复其数据库所需的最少信息。在发生检查点操作时，将自动截断所有非活动日志条目。

所有操作仍会被记录下来。但是，一旦出现检查点，日志就会自动截断。这种截断意味着日志可供重复使用，并且可以覆盖较旧的日志条目。当日志条目被覆盖时，无法捕获更改。这个问题就是 AWS DMS 不支持 SIMPLE 数据恢复模式的原因。有关将 SQL Server 作为源所需的其他先决条件的信息，请参阅[使用微软 SQL Server 数据库作为来源 AWS DMS](#)。

## 跨分区映射的不一致表

在更改数据捕获 (CDC) 期间，当 AWS DMS 无法对表正确执行 CDC 时，对特殊结构的表的迁移将暂停。并提示类似以下消息：

```
[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)
```

对 SQL Server 表运行 CDC 时，AWS DMS 会解析 SQL Server tlog。在每条 tlog 记录上，AWS DMS 会解析包含在更改过程中插入、更新或删除的列数据的十六进制值。

为了解析十六进制记录，AWS DMS 会从 SQL Server 系统表中读取表元数据。这些系统表用于标识特殊结构的表列是什么，并显示它们的一些内部属性，例如“xoffset”和“空位位置”。

AWS DMS 要求表的所有原始分区的元数据都是相同的。但是在某些情况下，特殊结构的表在所有分区上没有相同的元数据。在这些情况下，AWS DMS 可以暂停该表上的 CDC，以避免错误地解析更改并向目标提供错误的信息。解决方法如下所示：

- 如果表具有集群索引，请执行索引重建。
- 如果表没有集群索引，请向表中添加集群索引（稍后可以根据需要删除）。

## 排除 Amazon Redshift 中的问题

接下来，您可以了解如何排除将 AWS DMS 与 Amazon Redshift 数据库一起使用时的特定问题。

### 主题

- [加载到不同 AWS 区域内的 Amazon Redshift 集群中](#)
- [错误：关系“attrep\\_apply\\_exceptions”已存在](#)
- [其名称以“awsdms\\_changes”开头的表出错](#)
- [查看集群中名称类似于 dms.awsdms\\_changes000000000XXXX 的表](#)
- [使用 Amazon Redshift 所需的权限](#)

## 加载到不同 AWS 区域内的 Amazon Redshift 集群中

您不能加载到与您的 AWS DMS 复制实例位于不同 AWS 区域的 Amazon Redshift 集群中。DMS 要求您的复制实例和 Amazon Redshift 集群处于相同区域中。

### 错误：关系“attrep\_apply\_exceptions”已存在

在将 Redshift 终端节点指定为 PostgreSQL 终端节点时，通常会出现“关系‘awsdms\_apply\_exceptions’已存在”错误。要纠正该问题，请修改终端节点并将目标引擎更改为“redshift”。

### 其名称以“awsdms\_changes”开头的表出错

同时运行两个尝试将数据加载到同一个 Amazon Redshift 集群中的任务时，通常会出现名称以“awsdms\_changes”开头的表错误消息。由于临时表的命名方式，当更新同一表时，并发任务会发生冲突。

### 查看集群中名称类似于 dms.awsdms\_changes000000000XXXX 的表

从存储在 Amazon S3 中的文件加载数据时，AWS DMS 创建临时表。这些临时表的名称的前缀均为 dms.awsdms\_changes。这些表是必需的，以便 AWS DMS 能够在首次加载数据时以及在将数据放置到最终目标表之前存储数据。

## 使用 Amazon Redshift 所需的权限

要将 AWS DMS 与 Amazon Redshift 一起使用，您用来访问 Amazon Redshift 的用户账户必须具有以下权限：

- CRUD ( 选择、插入、更新、删除 )
- 批量加载
- 创建、更改、删除 ( 如果任务的定义要求 )

要查看有关将 Amazon Redshift 作为目标的所有先决条件，请参阅[将 Amazon Redshift 数据库作为 AWS Database Migration Service 目标](#)。

## 排除 Amazon Aurora MySQL 中的问题

接下来，您可以了解如何排除将 AWS DMS 与 Amazon Aurora MySQL 数据库一起使用时的特定问题。

## 主题

- [错误：CHARACTER SET UTF8 字段由“,”终止，包含在“”中，行由“\n”终止](#)

## 错误：CHARACTER SET UTF8 字段由“,”终止，包含在“”中，行由“\n”终止

如果您使用 Amazon Aurora MySQL 作为目标，则可能会在日志中看到类似以下内容的错误。此类错误通常表示您将 ANSI\_QUOTES 作为 SQL\_MODE 参数的一部分。将 ANSI\_QUOTES 作为 SQL\_MODE 参数的一部分会导致像处理引号一样处理双引号，在运行任务时导致出现问题。

要纠正此错误，请从 SQL\_MODE 参数中删除 ANSI\_QUOTES。

```
2016-11-02T14:23:48 [TARGET_LOAD ]E: Load data sql statement. load data local infile
"/rdsdbdata/data/tasks/7X04FJHCV0N7TYTLQ6RX3CQH DU/data_files/4/LOAD000001DF.csv" into
table
`VOSPUSER`.`SANDBOX_SRC_FILE` CHARACTER SET UTF8 fields terminated by ','
enclosed by '"' lines terminated by '\n'(`SANDBOX_SRC_FILE_ID`,`SANDBOX_ID`,
`FILENAME`,`LOCAL_PATH`,`LINES_OF_CODE`,`INSERT_TS`,`MODIFIED_TS`,`MODIFIED_BY`,
`RECORD_VER`,`REF_GUID`,`PLATFORM_GENERATED`,`ANALYSIS_TYPE`,`SANITIZED`,`DYN_TYPE`,
`CRAWL_STATUS`,`ORIG_EXEC_UNIT_VER_ID`); (provider_syntax_manager.c:2561)
```

## 排除 SAP ASE 的问题

接下来，您可以了解如何排除将 AWS DMS 与 SAP ASE 数据库一起使用时的特定问题。

## 错误：当源具有包含 NULL 值的复合唯一索引时，LOB 列的值为 NULL

使用 SAP ASE 作为源时，如果表配置了允许 NULL 值的复合唯一索引，则在持续复制过程中，LOB 值可能不会迁移。这种行为通常是 DMS 复制实例客户端上默认将 ANSI\_NULL 设置为 1 的结果。

为确保 LOB 字段正确迁移，请为任务的 AWS DMS 源端点提供端点设置 'AnsiNull=0'。

## 排除 IBM Db2 中的问题

接下来，您可以了解如何排除将 AWS DMS 与 IBM Db2 数据库一起使用时的特定问题。

## 错误：不支持从时间戳恢复任务

对于持续复制 (CDC)，如果您计划从特定时间戳开始复制，请将连接属性 `StartFromContext` 设置为所需的时间戳。有关更多信息，请参阅[使用 Db2 LUW 时的端点设置](#)。将 `StartFromContext` 设置为所需的时间戳可以防止出现以下问题：

```
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.
```

## 排除 AWS Database Migration Service 中的延迟问题

本节概述了在持续复制阶段 (CDC)，造成 AWS DMS 任务延迟的常见原因。AWS DMS 异步复制数据。延迟是在源上提交了更改与将更改复制到目标之间的延误。延迟可能是由于复制组件的配置错误所致，如下所示：

- 源端点或数据来源
- 目标端点或数据来源
- 复制实例
- 这些组件之间的网络

我们建议您使用测试迁移作为概念验证，用于收集有关复制的信息。然后，您可以使用此信息来调整复制配置以尽可能减少延迟。有关运行概念验证迁移的信息，请参阅[运行概念验证](#)。

### 主题

- [CDC 延迟的类型](#)
- [CDC 延迟的常见原因](#)
- [排除延迟问题](#)

## CDC 延迟的类型

本节包含 CDC 期间可能发生的复制延迟的类型。



## 源延迟

从源端点中捕获的最后一个事件与复制实例的当前系统时间戳之间的间隔（秒）。您可以使用该 CDCLatencySource CloudWatch 指标监控数据源和复制实例之间的延迟。较高的 CDCLatencySource 指标表示从源捕获更改的过程存在延迟。例如，如果您的应用程序在 10:00 向源提交插入内容，并 AWS DMS 在 10:02 使用更改，则 CDCLatencySource 指标为 120 秒。

有关 CloudWatch 指标的信息 AWS DMS，请参阅 [复制任务指标](#)。

## 目标延迟

源上第一个等待提交到目标上时间的时间戳，与 DMS 复制实例的当前系统时间戳之间的间隔（秒）。您可以使用该 CDCLatencyTarget CloudWatch 指标监控数据源和数据目标提交之间的延迟。这意味着 CDCLatencyTarget 包括从来源进行读取的任何延迟。因此，CDCLatencyTarget 始终大于或等于 CDCLatencySource。

例如，如果您的应用程序在 10:00 向源提交插入内容，并 AWS DMS 在 10:02 提交插入并在 10:05 将其写入目标，则 CDCLatencyTarget 指标为 300 秒。

## CDC 延迟的常见原因

本节包含 CDC 期间复制可能遇到延迟的原因。

### 主题

- [端点资源](#)
- [复制实例资源](#)
- [网络速度和带宽](#)
- [DMS 配置](#)
- [复制场景](#)

## 端点资源

以下因素会严重影响复制性能和延迟：

- 源数据库和目标数据库配置
- 实例大小
- 源或目标数据存储的预置不足或配置错误

要确定AWS托管源和目标端点问题导致延迟的原因，请监控以下 CloudWatch 指标：

- FreeMemory
- CPUUtilization
- 吞吐量和 I/O 指标，例如 WriteIOPS、WriteThroughput 或 ReadLatency
- 事务量指标，例如 CDCIncomingChanges。

有关监控 CloudWatch 指标的信息，请参阅[AWS Database Migration Service 指标](#)。

## 复制实例资源

复制实例资源对于复制至关重要，您应确保没有资源瓶颈，因为它们会导致源和目标延迟。

要确定复制实例中存在的资源瓶颈，请验证以下几点：

- CPU、内存、每秒 I/O 和存储等关键 CloudWatch 指标没有出现峰值或持续较高的值。
- 您的复制实例的大小适合工作负载。有关如何确定复制实例的合适大小的信息，请参阅[选择复制实例的最佳大小](#)。

## 网络速度和带宽

网络带宽是影响数据传输的一个因素。要分析复制的网络性能，请执行以下操作之一：

- 检查实例级别的 ReadThroughput 和 WriteThroughput 指标。有关监控 CloudWatch 指标的信息，请参阅[AWS Database Migration Service 指标](#)。
- 使用 AWS DMS 诊断支持 AMI。如果您所在的地区没有诊断支持 AMI 可用，则可以从任何支持的区域下载该 AMI，然后将其复制到您所在的区域以执行网络分析。有关诊断支持 AMI 的信息，请参阅[使用 AWS DMS 诊断支持 AMI](#)。

AWS DMS 中的 CDC 是单线程的，以确保数据一致性。因此，您可以通过计算单线程数据传输速率来确定您的网络可以支持的数据量。例如，如果您的任务使用 100 Mbps（每秒兆比特）网络连接到的源，则您的复制的理论最大带宽分配为 12.5 MBps（每秒兆字节）。这相当于每小时 45 Gb。如果源服务器上的事务日志生成的速率大于每小时 45 GB，则表示该任务有 CDC 延迟。对于 100 Mbps 的网络，这些速率是理论上的最大值；网络流量以及源和目标上的资源开销等其他因素会减少实际可用带宽。

## DMS 配置

本节包含有助于减少延迟的推荐复制配置。

- **端点设置**：您的源端点和目标端点设置可能会导致复制实例的性能不佳。启用资源密集型功能的端点设置会影响性能。例如，对于 Oracle 端点，禁用 LogMiner 和使用 Binary Reader 可以提高性能，因为 LogMiner 这会占用大量资源。以下端点设置可提高 Oracle 端点的性能：

```
useLogminerReader=N;useBfile=Y
```

有关端点设置的更多信息，请参阅[使用 AWS DMS 端点](#)主题中关于源和目标端点引擎的文档。

- **任务设置**：针对您的特定复制场景的某些任务设置可能会导致您的复制实例性能不佳。例如，默认情况下，对除 Amazon Redshift 之外的所有端点，AWS DMS 默认情况下为 CDC 使用事务应用模式 (BatchApplyEnabled=false)。但是，对于有大量更改的源，将 BatchApplyEnabled 设置为 true 可能会提高性能。

有关任务设置的更多信息，请参阅[为 AWS Database Migration Service 任务指定任务设置](#)。

- **仅 CDC 任务的启动位置**：从位于过去的位置或时间戳启动仅 CDC 任务，将会启动具有更高 CDC 源延迟的任务。根据源端的更改量，任务延迟需要一段时间才能消失。
- **LOB 设置**：由于 AWS DMS 复制大型二进制数据的方式，大型对象数据类型可能会影响复制性能。有关更多信息，请参阅以下主题：
  - [在任务中为源数据库设置 LOB 支持 AWS DMS](#)
  - [迁移大型二进制对象 \(LOB\)](#)

## 复制场景

本节介绍特定的复制场景及其如何影响延迟。

### 主题

- [在较长的一段时间内停止任务](#)
- [缓存更改](#)
- [跨区域复制](#)

## 在较长的一段时间内停止任务

停止任务时，AWS DMS 保存从源读取的最后一个事务日志的位置。当您恢复任务执行时，DMS 会尝试从相同的事务日志位置继续读取。在几小时或几天后恢复任务会导致 CDC 源延迟增加，直到 DMS 处理完积压的事务为止。

## 缓存更改

缓存更改是在 AWS DMS 运行完全加载阶段时，应用程序写入数据来源的更改。DMS 直到完全加载阶段完成并且 CDC 阶段开始时才应用这些更改。对于具有大量事务的来源，应用缓存更改需要更长的时间，因此 CDC 阶段开始时，源延迟会增加。我们建议您在事务量较低时运行完全加载阶段，以尽可能减少缓存的更改数量。

## 跨区域复制

将 DMS 端点或复制实例放在不同的 AWS 区域会增加网络延迟。这会增加复制延迟。为了获得最佳性能，请将源端点、目标端点和复制实例放在同一 AWS 区域。

## 排除延迟问题

本节包含复制延迟的故障排除步骤。

要排除延迟问题，请执行以下操作：

- 首先，确定任务的延迟类型和时间长度。从 DMS 控制台或 CLI 查看任务的“表统计数据”部分。如果计数器发生变化，则数据传输正在进行中。检查 `CDCLatencySource` 和 `CDCLatencyTarget` 指标，以确定 CDC 期间是否存在瓶颈。
- 如果高 `CDCLatencySource` 或 `CDCLatencyTarget` 指标表明您的复制存在瓶颈，请检查以下内容：
  - 如果 `CDCLatencySource` 较高且 `CDCLatencyTarget` 等于 `CDCLatencySource`，则表明您的源端点存在瓶颈，并且 AWS DMS 正在顺利地向目标端点写入数据。参见下文中的 [排除源延迟问题](#)。
  - 如果 `CDCLatencySource` 较低且 `CDCLatencyTarget` 较高，则表示您的目标端点存在瓶颈，并且 AWS DMS 正在顺畅地从源读取数据。参见下文中的 [排除目标延迟问题](#)。
  - 如果 `CDCLatencySource` 较高且 `CDCLatencyTarget` 明显高于 `CDCLatencySource`，则表示源读取和目标写入都存在瓶颈。首先调查源延迟，然后再调查目标延迟。

有关监控 DMS 指标的信息，请参阅 [监控 AWS DMS 任务](#)。

## 排除源延迟问题

以下主题描述了特定于源端点类型的复制场景。

### 主题

- [Oracle 端点故障排除](#)
- [MySQL 端点故障排除](#)
- [PostgreSQL 端点故障排除](#)
- [SQL Server 端点故障排除](#)

### Oracle 端点故障排除

本节包含特定于 Oracle 的复制场景。

#### 源读取暂停

在以下情况下，AWS DMS 会暂停从 Oracle 源读取。此行为是设计使然。您可以使用任务日志调查这种情况的原因。在任务日志中查找类似于以下内容的消息。有关使用任务日志的更多信息，请参阅[查看和管理 AWS DMS 任务日志](#)。

- SORTER 消息：这表明 DMS 正在缓存复制实例上的事务。有关更多信息，请参阅下面的[任务日志中的 SORTER 消息](#)。
- 调试任务日志：如果 DMS 中断了读取进程，则您的任务会将以下消息重复写入调试任务日志，而不会更改上下文或时间戳：
  - 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 为每个新的重做或存档日志操作记录以下消息。

```
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)
```

如果源有新的重做或归档日志操作，并且 AWS DMS 没有将这些消息写入日志，则这意味着任务没有处理事件。

## 重做生成速率高

如果您的任务正在处理重做或存档日志，但源延迟仍然很高，请尝试确定重做日志的生成速率和生成模式。如果您有很高的重做日志生成水平，这会增加源延迟，因为任务会读取所有重做和存档日志，以便提取与复制的表相关的更改。

要确定重做生成速率，请使用以下查询。

- 每日重做生成速率：

```
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;
```

- 每小时重做生成速率：

```
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 ;
```

在此场景中要排除延迟问题，请检查以下内容：

- 检查网络带宽和复制的单线程性能，确保底层网络可以支持源重做生成速率。有关网络带宽如何影响复制性能的信息，请参见前文中的[网络速度和带宽](#)。

- 检查您是否正确设置了补充日志记录。请避免在源上进行额外的日志记录，例如在表的所有列上启用日志记录。有关设置补充日志记录的信息，请参阅[设置补充日志记录](#)。
- 确认您使用了正确的 API 来读取重做或归档日志。您可以使用 Oracle LogMiner 或 AWS DMS 二进制阅读器。LogMiner 读取在线重做日志和存档的重做日志文件时，Binary Reader 会直接读取和解析原始重做日志文件。因此，Binary Reader 的性能更高。如果您的重做日志生成速率超过 10 GB/小时，建议您使用 Binary Reader。有关更多信息，请参见 [在 CDC 中使用 Oracle LogMiner 或 AWS DMS 二进制阅读器](#)。
- 检查 ArchivedLogsOnly 是否设置为 Y。如果设置了此端点设置，则 AWS DMS 会从存档重做日志进行读取。这会增加源延迟，因为 AWS DMS 要等待在线重做日志存档后再读取。有关更多信息，请参阅[ArchivedLogsOnly](#)。
- 如果您的 Oracle 源使用自动存储管理 (ASM)，有关如何正确配置数据存储的信息，请参阅 [使用 Oracle 作为源时在 Oracle ASM 上存储 REDO AWS DMS](#)。您还可以使用 asmUsePLSQLArray 额外连接属性 (ECA) 进一步优化读取性能。有关使用 asmUsePLSQLArray 的信息，请参阅 [使用 Oracle 作为来源时的终端节点设置 AWS DMS](#)。

## MySQL 端点故障排除

本节包含特定于 MySQL 的复制场景。AWS DMS 定期扫描 MySQL 二进制日志以复制更改。以下情况下，此过程可能会增加延迟：

### 主题

- [源上长时间运行的事务](#)
- [源端工作负载过高](#)
- [二进制日志争用](#)

### 源上长时间运行的事务

由于 MySQL 仅将已提交的事务写入二进制日志，因此长时间运行的事务会导致与查询运行时间成比例的延迟峰值。

要识别长时间运行的事务，请使用以下查询或使用慢速查询日志：

```
SHOW FULL PROCESSLIST;
```

有关使用慢速查询日志的信息，请参阅 [MySQL 文档](#) 中的 [慢速查询日志](#)。

为避免长时间运行的事务导致延迟峰值，请重构源事务，以缩短查询运行时间或增加提交频率。

## 源端工作负载过高

由于 DMS CDC 是单线程的，因此大量事务会增加源延迟。要确定源延迟是否由于工作负载繁重而造成，请将出现延迟期间生成的二进制日志的数量和大小，与出现延迟之前生成的日志的数量和大小进行比较。要检查二进制日志和 DMS CDC 线程状态，请使用以下查询：

```
SHOW BINARY LOGS;  
SHOW PROCESSLIST;
```

有关 CDC 二进制日志转储线程状态的更多信息，请参阅[复制源线程状态](#)。

您可以将源上生成的最新二进制日志位置与 DMS 当前正在处理的事件进行比较来确定延迟。要确定源上最新的二进制日志，请执行以下操作：

- 在 SOURCE\_CAPTURE 组件上启用调试日志。
- 从任务调试日志中，检索 DMS 处理二进制日志和位置详细信息。
- 使用以下查询来确定源上最新的二进制日志：

```
SHOW MASTER STATUS;
```

要进一步优化性能，请调整 EventsPollInterval。默认情况下，DMS 每 5 秒轮询一次二进制日志，但您可以通过降低此频率来提高性能。有关 EventsPollInterval 设置的更多信息，请参阅[使用 MySQL 作为来源时的端点设置 AWS DMS](#)。

## 二进制日志争用

迁移包含大量数据的多个表时，对于 MySQL 5.7.2 或更高版本，建议将表拆分为单独的任务。在 MySQL 5.7.2 及更高版本中，主转储线程创建的锁争用较少并提高了吞吐量。因此，转储线程在读取事件时不再锁定二进制日志。这意味着多个转储线程可以同时读取二进制日志文件。这还意味着转储线程可以在客户端写入二进制日志时读取二进制日志。有关转储线程的更多信息，请参阅[复制线程](#)和[MySQL 5.7.2 发行说明](#)。

对于 5.7.2 之前版本的 MySQL 源，要提高复制性能，请尝试使用 CDC 组件整合任务。

## PostgreSQL 端点故障排除

本节包含特定于 PostgreSQL 的复制场景。

### 主题

- [源上长时间运行的事务](#)



- [源端工作负载过高](#)
- [高网络吞吐量](#)
- [在 Aurora PostgreSQL 中泄露文件](#)

## 源上长时间运行的事务

当源数据库中有长时间运行的事务（例如单个事务中有几千次插入）时，在事务完成之前，DMS CDC 事件和事务计数器不会增加。这种延迟可能会导致延迟问题，您可以使用 `CDCLatencyTarget` 指标来衡量。

要查看长期运行的事务，请执行以下操作之一：

- 使用 `pg_replication_slots` 视图。如果 `restart_lsn` 值未更新，PostgreSQL 很可能因为长时间运行的活动事务而无法发布预写日志 (WAL)。有关 `pg_replication_slots` 视图的信息，请参阅 [PostgreSQL 15.4 文档](#) 中的 [pg\\_replication\\_slots](#)。
- 使用以下查询返回数据库中所有活动查询的列表以及相关信息：

```
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;
```

在查询结果中，`age` 字段显示每个查询的活动时长，您可以使用该时长来确定长时间运行的查询。

## 源端工作负载过高

如果您的源 PostgreSQL 的工作负载很高，请检查以下事项以减少延迟：

- 使用 `test_decoding` 插件，从具有高每秒事务数 (TPS) 值的源数据库迁移表的子集时，您可能会遇到高延迟。这是因为 `test_decoding` 插件会将所有数据库更改发送到复制实例，然后 DMS 会根据任务的表映射对这些更改进行筛选。不属于任务表映射的表中的事件可能会增加源延迟。
- 使用以下方法之一检查 TPS 吞吐量。
  - 对于 Aurora PostgreSQL 来源，请使用该指标。CommitThroughput CloudWatch
  - 对于在 Amazon RDS 或本地运行的 PostgreSQL，请通过 PSQL 客户端版本 11 或更高版本使用以下查询（在查询期间按 **enter** 可显示结果进度）：

```
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";
```

- 为了减少使用 `test_decoding` 插件时的延迟，请考虑改为使用 `pglogical` 插件。与 `test_decoding` 插件不同，`pglogical` 插件会筛选源端的预写日志 (WAL) 更改，并且仅将相关更改发送到复制实例。有关将 `pglogical` 插件与 AWS DMS 结合使用的信息，请参阅[配置 `pglogical` 插件](#)。

## 高网络吞吐量

使用 `test_decoding` 插件时，您的复制可能会占用大量的网络带宽，尤其是在有大量事务期间。这是因为 `test_decoding` 插件会处理更改，并将它们转换为人类可读格式，这比原始二进制格式更大。

为了提高性能，请考虑改用 `pglogical` 插件，它是一个二进制插件。与 `test_decoding` 插件不同，`pglogical` 插件生成二进制格式的输出，从而导致压缩的预写日志 (WAL) 流更改。

## 在 Aurora PostgreSQL 中泄露文件

在 PostgreSQL 版本 13 及更高版本中，`logical_decoding_work_mem` 该参数决定了用于解码和流式传输的内存分配。[有关该 `logical\_decoding\_work\_mem` 参数的更多信息，请参阅 PostgreSQL 13.13 文档中的 PostgreSQL 中的资源消耗。](#)

逻辑复制会累积内存中所有事务的更改，直到这些事务提交为止。如果所有事务中存储的数据量超过数据库参数指定的量 `logical_decoding_work_mem`，则 DMS 会将事务数据溢出到磁盘，以便为新的解码数据释放内存。

长时间运行的事务或许多子事务可能会导致 DMS 消耗更多的逻辑解码内存。内存使用量的增加会导致 DMS 在磁盘上创建溢出文件，从而导致复制过程中的源延迟很高。

要减少源工作负载增加的影响，请执行以下操作：

- 减少长时间运行的交易。
- 减少子交易的数量。
- 避免执行会生成大量日志记录的操作，例如在单个事务中删除或更新整个表。改为以较小的批量执行操作。

您可以使用以下 CloudWatch 指标来监控源上的工作负载：

- `TransactionLogsDiskUsage` : 逻辑 WAL 当前占用的字节数。如果逻辑复制槽无法跟上新写入的步伐，或者如果任何长时间运行的事务无法对旧文件进行垃圾回收，则此值会单调增加。
- `ReplicationSlotDiskUsage` : 逻辑复制插槽当前使用的磁盘空间量。

您可以通过调整 `logical_decoding_work_mem` 参数来减少源延迟。此参数的默认值为 64 MB。此参数限制每个逻辑流复制连接使用的内存量。我们建议将该 `logical_decoding_work_mem` 值设置得比该 `work_mem` 值高得多，以减少 DMS 写入磁盘的已解码更改量。

我们建议您定期检查泄漏文件，尤其是在迁移活动繁忙或延迟期间。如果 DMS 正在创建大量泄漏文件，则意味着逻辑解码的运行效率不高，这可能会增加延迟。要缓解这种情况，请增加 `logical_decoding_work_mem` 参数值。

您可以使用该 `aurora_stat_file` 函数检查当前事务溢出情况。有关更多信息，请参阅 Amazon Relational Database Service 开发人员指南中的 [调整工作内存以进行逻辑解码](#)。

## SQL Server 端点故障排除

本节包含特定于 SQL Server 的复制场景。为确定要从 SQL Server 复制哪些更改，AWS DMS 读取事务日志，并对源数据库运行定期扫描。复制延迟通常是因为资源有限，SQL Server 限制这些扫描所导致。这也可能是由于短时间内写入事务日志的事件数量显著增加所导致。

### 主题

- [索引重建](#)
- [大型事务](#)
- [Amazon RDS SQL Server 的 MS-CDC 轮询间隔配置错误](#)
- [从同一个源数据库复制多个 CDC 任务](#)

### 索引重建

SQL Server 在重建大型索引时使用单个事务。这会生成大量事件，如果 SQL Server 同时重建多个索引，就可能会占用大量日志空间。发生这种情况时，可以预计会出现短暂的复制高峰。如果您的 SQL Server 源持续出现日志峰值，请检查以下内容：

- 首先，使用 `CDCLatencySource` 和 `CDCLatencySource` CloudWatch 指标或通过查看任务日志中的吞吐量监控消息来检查延迟峰值的时间段。有关 CloudWatch 指标的信息 AWS DMS，请参阅 [复制任务指标](#)。

- 检查在延迟高峰期期间，活动事务日志或日志备份的大小是否增加。还要检查在这段时间内是否运行了维护作业或重建。有关检查事务日志大小的信息，请参阅 [SQL Server 技术文档](#) 中的 [监控日志空间使用情况](#)。
- 验证您的维护计划是否符合 SQL Server 最佳实践。有关 SQL Server 维护最佳实践的信息，请参阅 [SQL Server 技术文档](#) 中的 [索引维护策略](#)。

要修复索引重建期间延迟问题，请尝试以下操作：

- 使用 BULK\_LOGGED 恢复模式进行离线重建，以减少任务必须处理的事件。
- 如果可能，请在索引重建期间停止任务。或者，尝试将索引重建安排在非高峰时段，以减轻延迟峰值的影响。
- 尝试找出导致 DMS 读取速度减慢的资源瓶颈，例如磁盘延迟或 I/O 吞吐量，并加以解决。

## 大型事务

具有大量事件的事务或长时间运行的事务会导致事务日志增长。这会导致 DMS 读取时间更长，从而产生延迟。这类似于索引重建对复制性能的影响。

如果您不熟悉源数据库上的典型工作负载，则可能很难识别此问题。要排除此问题，请执行以下操作：

- 首先，使用 ReadThroughput 和 WriteThroughput CloudWatch 指标或通过查看任务日志中的吞吐量监控消息来确定延迟激增的时间。
- 检查源数据库上在延迟高峰期期间是否有任何长时间运行的查询。有关长时间运行的查询的信息，请参阅 [SQL Server 技术文档](#) 中的 [SQL Server 中运行缓慢的查询故障排除](#)。
- 检查活动事务日志或日志备份的大小是否增加。有关更多信息，请参阅 [SQL Server 技术文档](#) 中的 [监控日志空间使用情况](#)。

要修复这个问题，请执行以下操作之一：

- 最好的解决办法是在应用程序端重构您的事务，使它们能够快速完成。
- 如果您无法重构事务，则短期解决方法是检查是否存在资源瓶颈，例如磁盘等待或 CPU 争用。如果您发现源数据库中存在瓶颈，则可以通过增加源数据库的磁盘、CPU 和内存资源来减少延迟。这可减少系统资源的争用，使得 DMS 查询可以更快地完成。

## Amazon RDS SQL Server 的 MS-CDC 轮询间隔配置错误

Amazon RDS 实例上错误配置的轮询间隔设置可能会导致事务日志增长。这是因为复制会防止日志截断。虽然正在运行的任务可能会以最小的延迟继续复制，但停止和恢复任务或启动仅 CDC 的任务会导致任务失败。这是由于扫描大型事务日志时超时所造成。

要排除错误配置的轮询间隔，请执行以下操作：

- 检查活动的事务日志大小是否在增加，以及日志使用率是否接近 100%。有关更多信息，请参阅 [SQL Server 技术文档](#) 中的 [监控日志空间使用情况](#)。
- 检查日志截断是否延迟，并且 `log_reuse_wait_desc` value 为 REPLICATION。有关更多信息，请参阅 [SQL Server 技术文档](#) 中的 [事务日志 \(SQL Server\)](#)。

如果您发现前面列表中的任何项目有问题，请调整 MS-CDC 轮询间隔。有关调整轮询间隔的信息，请参阅 [使用适用于 SQL Server 的 Amazon RDS 作为来源时的推荐设置 AWS DMS](#)。

### 从同一个源数据库复制多个 CDC 任务

在完全加载阶段，我们建议在任务之间拆分表以提高性能，以逻辑方式分隔依赖表，并减轻任务失败的影响。但是，在 CDC 阶段，我们建议整合任务以尽可能减少 DMS 扫描。在 CDC 阶段，每个 DMS 任务每分钟都会扫描几次事务日志以查找新事件。由于每个任务都是独立运行的，因此每个任务都会单独扫描每个事务日志。这会增加源 SQL Server 数据库的磁盘和 CPU 使用率。因此，大量并行运行的任务可能会导致 SQL Server 限制 DMS 读取，从而导致延迟增加。

如果有多个任务逐步开始，您可能很难确定此问题。此问题最常见的症状是大多数任务扫描开始需要花费更长的时间。这会导致这些扫描的延迟更高。SQL Server 会优先处理一些任务扫描，因此其中一些任务显示正常延迟。要排除此问题，请检查所有任务的 `CDCLatencySource` 指标。如果有些任务的 `CDCLatencySource` 增加，而少数几个任务的 `CDCLatencySource` 较低，则很可能是 SQL Server 限制了某些任务的 DMS 读取。

如果 SQL Server 在 CDC 期间限制任务读取，请整合任务以尽可能减少 DMS 扫描次数。可以连接到源数据库而不造成争用的最大任务数取决于源数据库容量、事务日志增长速率或表数量等因素。要确定复制场景的理想任务数量，请在与生产环境相似的测试环境中测试复制。

## 排除目标延迟问题

本节包含可能导致目标延迟的场景。

### 主题

- [索引问题](#)

- [任务日志中的 SORTER 消息](#)
- [数据库锁定](#)
- [LOB 查找缓慢](#)
- [多可用区、审计日志记录和备份](#)

## 索引问题

在 CDC 阶段，AWS DMS 通过在目标系统上执行 DML 语句（插入、更新和删除）来复制源上的更改。对于使用 DMS 的异构迁移，源端和目标端在索引优化方面的不同，可能会需要导致花费更长的时间写入目标系统。这会引发目标延迟和性能问题。

要排除此索引问题，请执行以下操作：这些步骤的过程因不同的数据库引擎而异。

- 监控目标数据库的查询时间。比较目标和源上的查询执行时间可以指明哪些索引需要优化。
- 为运行缓慢的查询启用日志记录。

要修复长时间运行复制的索引问题，请执行以下操作：

- 调整源数据库和目标数据库上的索引，使源数据库和目标数据库上具有相似的查询执行时间。
- 比较源和目标的 DML 查询中使用的二级索引。确保目标上的 DML 性能与源 DML 性能相当或更好。

请注意，优化索引的过程特定于数据库引擎。没有用于调整源索引和目标索引的 DMS 功能。

## 任务日志中的 SORTER 消息

如果目标端点无法跟上 AWS DMS 写入其中的更改量，则该任务会在复制实例上缓存更改。如果缓存的增长大于内部阈值，则任务将停止从源读取更多的更改。DMS 这样做是为了防止复制实例的存储空间不足，或者在读取大量待处理事件时任务停滞不前。

要解决此问题，请检查 CloudWatch 日志中是否有类似于以下任一内容的消息：

```
[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)
```

如果您的日志包含的消息与第一条消息类似，请禁用该任务的所有跟踪日志记录，并增加复制实例存储空间。有关增加复制实例存储空间的信息，请参阅 [修改复制实例](#)。

如果日志包含的消息与第二条消息类似，请执行以下操作：

- 如果包含大量事务或长时间运行 DML 操作的表，与任务中的其他表没有任何依赖关系，则将其移至单独的任务中。
- 增加 MemoryLimitTotal 和 MemoryKeepTime 的设置，使事务在内存中保留更长的时间。如果延迟一直保持较高，则这种方法没有帮助，但它有助于在短暂的交易量突发期间降低延迟。有关这些任务设置的信息，请参阅[更改处理优化设置](#)。
- 通过将 BatchApplyEnabled 设置为 true，评估是否可以对事务使用批量应用。有关 BatchApplyEnabled 设置的信息，请参阅[目标元数据任务设置](#)。

## 数据库锁定

如果应用程序访问 AWS DMS 用作复制目标的数据库，则应用程序可能会锁定 DMS 尝试访问的表。这会导致锁争用。由于 DMS 按照更改在源数据库上的发生顺序将更改写入目标数据库，因此由于锁争用而导致写入一个表出现延迟，会导致写入所有表出现延迟。

要排除此问题，请查询目标数据库以检查锁争用是否在阻止 DMS 写入事务。如果目标数据库阻止 DMS 写入事务，请执行以下一项或多项操作：

- 重构查询以更频繁地提交更改。
- 修改您的锁定超时设置。
- 对表进行分区以尽可能减少锁争用。

请注意，优化锁争用的过程特定于数据库引擎。没有可用于调整锁争用的 DMS 功能。

## LOB 查找缓慢

AWS DMS 复制大型对象 (LOB) 列时，它会在向目标写入更改之前对源执行查找。这种查找通常不会在目标上造成任何延迟，但是如果源数据库由于锁定而导致查找的延迟，则可能会出现目标延迟峰值。

这个问题通常很难诊断。要排除此问题，请在任务日志上启用详细调试，并比较 DMS LOB 查找调用的时间戳。有关启用详细调试的信息，请参阅[查看和管理 AWS DMS 任务日志](#)。

要修复这一问题，请尝试以下操作：

- 提高源数据库上的 SELECT 查询性能。
- 调整 DMS LOB 设置。有关调整 LOB 设置的更多信息，请参阅[迁移大型二进制对象 \(LOB\)](#)。

## 多可用区、审计日志记录和备份

对于 Amazon RDS 目标，在以下情况中目标延迟可能会增加：

- 备份
- 启用多可用区 (Multi-AZ) 后
- 启用数据库日志记录后，例如审计日志或慢速查询日志。

这些问题通常很难诊断。要排除这些问题，请监控在 Amazon RDS 维护时段或数据库负载繁重期间，延迟的周期性峰值。

要修复这些问题，请尝试以下操作：

- 如果可能，在短期迁移期间，请禁用多可用区、备份或日志记录。
- 将维护时段重新安排到活动量较低的时段。

## 使用 AWS DMS 中的诊断支持脚本

如果您在使用 AWS DMS 时遇到问题，支持工程师可能需要有关源数据库或目标数据库的更多信息。我们希望确保 AWS Support 在尽可能短的时间内获得尽可能多的所需信息。因此，我们开发了脚本，用于查询几个主要的关系数据库引擎的这些信息。

如果您的数据库有支持脚本可用，则可以使用下文所述相应脚本主题中的链接下载脚本。验证并查看脚本（如下所述）后，可以按照脚本主题中描述的步骤运行该脚本。脚本运行完成后，您可以将其输出上传到您的 AWS Support 案例中（描述同样见下文）。

在运行脚本之前，您可以检测下载或存储支持脚本时可能引入的任何错误。为此，请将脚本文件的校验和与 AWS 提供的值进行比较。AWS 使用 SHA256 算法计算校验和。

### 使用校验和验证支持脚本文件

1. 打开 <https://d2pwp9zz55emqw.cloudfront.net/sha256Check.txt> 中提供的最新校验和文件以验证这些支持脚本。例如，该文件可能包含如下内容。

```
MYSQL dfafd0d511477c699f96c64693ad0b1547d47e74d5c5f2f2025b790b1422e3c8
ORACLE 6c41ebcfc99518cfa8a10cb2ce8943b153b2cc7049117183d0b5de3d551bc312
POSTGRES 6ccd274863d14f6f3146fbdabbba43f2d8d4c6a4c25380d7b41c71883aa4f9790
SQL_SERVER 971a6f2c46aec8d083d2b3b6549b1e9990af3a15fe4b922e319f4fdd358debe7
```



- 在包含支持文件的目录中，运行适用于您的操作系统的 SHA256 验证命令。例如，在 macOS 操作系统上，您可以在本主题后面介绍的 Oracle 支持脚本上运行以下命令。

```
shasum -a 256 awsdms_support_collector_oracle.sql
```

- 将命令的结果与您打开的最新 sha256Check.txt 文件中显示的值进行比较。这两个值应该匹配。如果这些值不同，请联系您的支持工程师，告知不相符问题以及如何获取干净的支持脚本文件。

如果您有一个干净的支持脚本文件，则在运行脚本之前，请务必从性能和安全角度阅读并理解 SQL。如果您不愿意运行此脚本中的任何 SQL，则可以注释掉或删除有问题的 SQL。您也可以向支持工程师咨询任何可接受的解决方法。

成功完成后，除非另有说明，否则脚本将以可读的 HTML 格式返回输出。该脚本设计为从此 HTML 中排除任何可能危及您业务的数据或安全细节。它也不会对您的数据库或其环境进行任何修改。但是，如果您在 HTML 中发现任何不愿意分享的信息，请在上传 HTML 之前随意删除问题信息。当 HTML 的内容可以接受时，请在支持案例的案例详细信息中，使用附件进行上传。

以下各个主题描述了支持的 AWS DMS 数据库的可用脚本，以及如何运行这些脚本。您的支持工程师将引导您使用下面记录的特定脚本。

## 主题

- [Oracle 诊断支持脚本](#)
- [SQL Server 诊断支持脚本](#)
- [适用于 MySQL 兼容数据库的诊断支持脚本](#)
- [PostgreSQL 诊断支持脚本](#)

## Oracle 诊断支持脚本

接下来，您可以找到诊断支持脚本，可用于分析 AWS DMS 迁移配置中的本地数据库或 Amazon RDS for Oracle 数据库。这些脚本可以与源端点或目标端点一起使用。这些脚本均编写为在 SQL\*Plus 命令行实用程序中运行。有关使用此实用程序的更多信息，请参阅 Oracle 文档中的[使用 SQL 命令行](#)。

在运行脚本之前，请确保您使用的用户账户具有访问 Oracle 数据库所需的权限。所示权限设置假设用户按如下方式创建。

```
CREATE USER script_user IDENTIFIED BY password;
```

对于本地数据库，请按如下所示为 *script\_user* 设置最低权限。

```
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;
```

对于 Amazon RDS 数据库，请按如下所示设置最低权限。

```
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');
```

接下来，您可以找到有关如何下载、查看和运行可用于 Oracle 的每个 SQL\*Plus 支持脚本的说明。您还可以了解如何查看输出并将其上传到您的 AWS Support 案例中。

## 主题

- [awsdms\\_support\\_collector\\_oracle.sql 脚本](#)

## awsdms\_support\_collector\_oracle.sql 脚本

下载 [awsdms\\_support\\_collector\\_oracle.sql](#) 脚本。

此脚本收集有关您的 Oracle 数据库配置的信息。请记住验证脚本的校验和，如果校验和验证成功，请查看脚本中的 SQL 代码，注释掉任何您不愿意运行的代码。在您确定脚本的完整性和内容没有问题之后，可以运行该脚本。

运行脚本并将结果上传到您的支持案例

1. 使用以下 SQL\*Plus 命令行在数据库环境中运行脚本。

```
SQL> @awsdms_support_collector_oracle.sql
```

<result>

该脚本会显示简短的描述以及继续或中止运行的提示。按 [Enter] 键以继续。

</result>

2. 在随后出现的提示中，输入您要迁移的唯一一个架构的名称。
3. 在随后出现的提示中，输入您已定义用于连接数据库的用户的名称 (*script\_user*)。
4. 在随后出现的提示中，输入要检查数据的天数，或者接受默认值。然后，该脚本会从您的数据库中收集指定的数据。

<result>

脚本完成后，它会显示输出 HTML 文件的名称，例如

dms\_support\_oracle-2020-06-22-13-20-39-ORCL.html。该脚本将此文件保存在您的工作目录中。

</result>

5. 您可以查看此 HTML 文件并删除任何不愿意分享的信息。当您觉得 HTML 可以分享时，请将该文件上传到您的 AWS Support 案例中。有关上传此文件的更多信息，请参阅[使用 AWS DMS 中的诊断支持脚本](#)。

## SQL Server 诊断支持脚本

接下来，您可以找到诊断支持脚本的说明，该脚本可用于分析 AWS DMS 迁移配置中的本地数据库或 Amazon RDS for SQL Server 数据库。这些脚本可以与源端点或目标端点一起使用。对于本地数据库，请在 sqlcmd 命令行实用程序中运行这些脚本。有关使用此实用程序的更多信息，请参阅 Microsoft 文档中的 [sqlcmd – 使用实用程序](#)。

对于 Amazon RDS 数据库，您无法使用 `sqlcmd` 命令行实用程序进行连接。而是使用连接到 Amazon RDS SQL Server 的任意客户端工具运行这些脚本。

在运行脚本之前，请确保您使用的用户账户具有所需的访问权限，可以访问 SQL Server 数据库。无论是本地数据库还是 Amazon RDS 数据库，您都可以使用与访问 SQL Server 数据库相同的权限，无需 SysAdmin 角色。

## 主题

- [为本地 SQL Server 数据库设置最低权限](#)
- [为 Amazon RDS SQL Server 数据库设置最低权限](#)
- [在独立 SQL Server 上设置持续复制：不使用 sysadmin 角色](#)
- [在可用性组环境中的 SQL Server 上设置持续复制：无 sysadmin 角色](#)
- [SQL Server 支持脚本](#)

## 为本地 SQL Server 数据库设置最低权限

### 为运行本地 SQL Server 数据库设置最低权限

1. 使用 SQL Server Management Studio (SSMS) 创建采用密码身份验证的新 SQL Server 账户，例如 *on-prem-user*。
2. 在 SSMS 的用户映射部分中，选择 MSDB 和 MASTER 数据库（将提供公有权限），然后为您要运行脚本的数据库分配 DB\_OWNER 角色。
3. 打开新账户的上下文（右键单击）菜单，选择安全性，然后明确授予 Connect SQL 权限。
4. 运行以下授权命令。

```
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;
```

## 为 Amazon RDS SQL Server 数据库设置最低权限

以最低权限运行 Amazon RDS SQL Server 数据库

1. 使用 SQL Server Management Studio (SSMS) 创建采用密码身份验证的新 SQL Server 账户，例如 *rds-user*。
2. 在 SSMS 的用户映射部分中，选择 MSDB 数据库（将提供公有权限），然后为您要运行脚本的数据库分配 DB\_OWNER 角色。
3. 打开新账户的上下文（右键单击）菜单，选择安全性，然后明确授予 Connect SQL 权限。
4. 运行以下授权命令。

```
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;
```

## 在独立 SQL Server 上设置持续复制：不使用 sysadmin 角色

本节介绍如何针对 SQL Server 数据库源设置不需要用户账户具有 sysadmin 权限的持续复制。

### Note

运行本节中的步骤后，非 sysadmin DMS 用户将有权执行以下操作：

- 从联机事务日志文件中读取更改
- 通过访问磁盘从事务日志备份文件中读取更改
- 添加或更改 DMS 使用的发布内容
- 向发布内容中添加文章

1. 按照[捕获本地或 Amazon EC2 上的自管理 SQL Server 的数据更改](#)中所述设置用于复制的 Microsoft SQL Server。
2. 在源数据库上启用 MS-REPLICATION。这可以手动完成，也可以通过 sysadmin 用户身份运行一次任务来完成。
3. 使用以下脚本，在源数据库上创建 awsdms 架构：

```
use master
go
create schema awsdms
go

-- Create the table valued function [awsdms].[split_partition_list] on the Master
  database, as follows:
USE [master]
GO

set ansi_nulls on
go

set quoted_identifier on
go

if (object_id('[awsdms].[split_partition_list]','TF')) is not null

drop function [awsdms].[split_partition_list];

go

create function [awsdms].[split_partition_list]

(

@plist varchar(8000), -A delimited list of partitions

@dlim nvarchar(1) -Delimiting character

)

returns @partitionsTable table -Table holding the BIGINT values of the string
  fragments

(

pid bigint primary key

)

as
```

```
begin

declare @partition_id bigint;

declare @dml_pos integer;

declare @dml_len integer;

set @dml_len = len(@dml);

while (charindex(@dml,@plist)>0)

begin

set @dml_pos = charindex(@dml,@plist);

set @partition_id = cast( ltrim(rtrim(substring(@plist,1,@dml_pos-1))) as bigint);

insert into @partitionsTable (pid) values (@partition_id)

set @plist = substring(@plist,@dml_pos+@dml_len,len(@plist));

end

set @partition_id = cast (ltrim(rtrim(@plist)) as bigint);

insert into @partitionsTable (pid) values ( @partition_id );

return

end

GO
```

4. 使用以下脚本，在 Master 数据库上创建 [awsdms].[rtm\_dump\_dblog] 过程：

```
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,

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. 使用以下脚本，在 Master 数据库上创建证书：

```
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. 使用以下脚本，从证书创建登录名：

```
Use [master]
Go

CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
[awsdms_rtm_dump_dblog_cert];
```

7. 使用以下脚本，将登录名添加到 sysadmin 服务器角色：

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

8. 通过以下脚本，使用证书将签名添加到 [master].[awsdms].[rtm\_dump\_dblog]。

```

Use [master]
GO
ADD SIGNATURE
TO [master].[awsdms].[rtm_dump_dblog] BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
WITH PASSWORD = '@5trongpassword';

```

### Note

如果重新创建存储过程，您需要再次添加签名。

## 9. 使用以下脚本在 Master 数据库上创建 [awsdms].[rtm\_position\_1st\_timestamp] :

```

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

```





```
Go
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
[awsdms_rtm_position_1st_timestamp_cert];
```

12. 使用以下脚本，将登录名添加到 sysadmin 角色：

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_position_1st_timestamp_login];
```

13. 通过以下脚本，使用证书将签名添加到 [master].[awsdms].[rtm\_position\_1st\_timestamp]：

```
Use [master]
GO
ADD SIGNATURE
TO [master].[awsdms].[rtm_position_1st_timestamp]
BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
WITH PASSWORD = '@5trongpassword';
```

14. 使用以下脚本向 DMS 用户授予对新存储过程的执行访问权限：

```
use master
go
GRANT execute on [awsdms].[rtm_position_1st_timestamp] to dms_user;
```

15. 在以下每个数据库中，创建具有以下权限和角色的用户：

#### Note

您应该在每个副本上使用相同的 SID 创建 dmsnosysadmin 用户账户。以下 SQL 查询可以帮助验证每个副本上的 dmsnosysadmin 账户 SID 值。有关创建用户的更多信息，请参阅 [Microsoft SQL Server 文档](#) 中的 [CREATE USER \(Transact-SQL\)](#)。有关为 Azure SQL 数据库创建 SQL 用户账户的更多信息，请参阅 [活动异地复制](#)。

```
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]
```

在源数据库上运行以下脚本：

```
EXEC sp_addrolemember N'db_owner', N'DMS_user'
use Source_DB
go
```

16. 最后，向源 SQL Server 端点添加额外连接属性 (ECA)：

```
enableNonSysadminWrapper=true;
```

## 在可用性组环境中的 SQL Server 上设置持续复制：无 sysadmin 角色

本节介绍如何在不需要用户账户具有 sysadmin 权限的可用性组环境中为 SQL Server 数据库源设置持续复制。


### Note

运行本节中的步骤后，非 sysadmin DMS 用户将有权执行以下操作：

- 从联机事务日志文件中读取更改
- 通过访问磁盘从事务日志备份文件中读取更改
- 添加或更改 DMS 使用的发布内容
- 向发布内容中添加文章

在可用性组环境中设置持续复制而不使用 sysadmin 用户

1. 按照[捕获本地或 Amazon EC2 上的自管理 SQL Server 的数据更改](#)中所述设置用于复制的 Microsoft SQL Server。
2. 在源数据库上启用 MS-REPLICATION。这可以手动完成，也可以使用 sysadmin 用户身份运行一次任务来完成。

 Note

您应该将 MS-REPLICATION 分发服务器配置为本地分发服务器，或者配置为允许非 sysadmin 用户通过关联的链接服务器进行访问。

3. 如果启用在单个任务中独占使用 sp\_repldone 端点选项，请停止 MS-REPLICATION 日志读取器作业。
4. 在每个副本上，执行以下步骤：
  1. 在 Master 数据库中创建 [awsdms][awsdms] 架构：

```
CREATE SCHEMA [awsdms]
```

2. 在 Master 数据库中创建 [awsdms].[split\_partition\_list] 表值函数：

```
USE [master]
GO

SET ansi_nulls on
GO

SET quoted_identifier on
GO

IF (object_id('[awsdms].[split_partition_list]','TF')) is not null
    DROP FUNCTION [awsdms].[split_partition_list];
GO

CREATE FUNCTION [awsdms].[split_partition_list]
(
    @plist varchar(8000),    --A delimited list of partitions
    @dlim nvarchar(1)      --Delimiting character
)

```

```
RETURNS @partitionsTable table --Table holding the BIGINT values of the string
    fragments
(
    pid bigint primary key
)
AS
BEGIN
    DECLARE @partition_id bigint;
    DECLARE @d1m_pos integer;
    DECLARE @d1m_len integer;
    SET @d1m_len = len(@d1m);
    WHILE (charindex(@d1m,@plist)>0)
    BEGIN
        SET @d1m_pos = charindex(@d1m,@plist);
        SET @partition_id = cast( ltrim(rtrim(substring(@plist,1,@d1m_pos-1))) as
            bigint);
        INSERT into @partitionsTable (pid) values (@partition_id)
        SET @plist = substring(@plist,@d1m_pos+@d1m_len,len(@plist));
    END
    SET @partition_id = cast (ltrim(rtrim(@plist)) as bigint);
    INSERT into @partitionsTable (pid) values ( @partition_id );
    RETURN
END
GO
```

### 3. 在 Master 数据库中创建 [awsdms].[rtm\_dump\_dblog] 过程：

```
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.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 -- 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. 在 Master 数据库上创建证书：

```

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. 从证书创建登录名：

```
USE [master]
GO
CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
    [awsdms_rtm_dump_dblog_cert];
```

## 6. 将登录名添加到 sysadmin 服务器角色：

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

## 7. 使用证书将签名添加到 [master].[awsdms].[rtm\_dump\_dblog] 过程：

```
USE [master]
GO

ADD SIGNATURE
    TO [master].[awsdms].[rtm_dump_dblog]
    BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
    WITH PASSWORD = '@hardpassword1';
```

### Note

如果重新创建存储过程，您需要再次添加签名。

## 8. 在 Master 数据库中创建 [awsdms].[rtm\_position\_1st\_timestamp] 过程：

```
USE [master]
IF object_id('[awsdms].[rtm_position_1st_timestamp]','P') is not null
    DROP PROCEDURE [awsdms].[rtm_position_1st_timestamp];
GO
CREATE PROCEDURE [awsdms].[rtm_position_1st_timestamp]
(
    @dbname          sysname,          -- Database name
    @seqno           integer,          -- Backup set sequence/position number
    within file
    @filename        varchar(260),    -- The backup filename
    @1stTimeStamp    varchar(40)      -- The timestamp to position by
)
AS
BEGIN
    SET NOCOUNT ON          -- Disable "rows affected display"
```



```

DECLARE @firstMatching table
(
    cLsn varchar(32),
    bTim datetime
)
DECLARE @sql nvarchar(4000)
DECLARE @nl          char(2)
DECLARE @tb          char(2)
DECLARE @fnameVar   sysname = 'NULL'

SET @nl = char(10); -- New line
SET @tb = char(9)  -- Tab separator

IF (@filename is not null)
    SET @fnameVar = '''+@filename +''''
SET @filename = '''+@filename +''''
SET @sql='use ['+@dbname+'];'+@nl+
    'SELECT TOP 1 [Current LSN],[Begin Time]'+@nl+
    'FROM fn_dump_dblog (NULL, NULL, NULL, '+ cast(@seqno as varchar(10))+','+
@filename +','+@nl+
    @tb+'default, default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default, default,'+@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(''+@lstTimeStamp+'''+ 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. 在 Master 数据库上创建证书：

```
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. 从证书创建登录名：


```
USE [master]
GO
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
    [awsdms_rtm_position_1st_timestamp_cert];
```

11. 将登录名添加到 sysadmin 服务器角色：

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER
    [awsdms_rtm_position_1st_timestamp_login];
```


12. 使用证书将签名添加到 [master].[awsdms].[rtm\_position\_1st\_timestamp] 过程：

```
USE [master]
GO
ADD SIGNATURE
    TO [master].[awsdms].[rtm_position_1st_timestamp]
    BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
    WITH PASSWORD = '@hardpassword1';
```

 Note

如果重新创建存储过程，您需要再次添加签名。

13. 在以下每个数据库中，创建具有以下权限/角色的用户：

 Note

您应该在每个副本上使用相同的 SID 创建 dmsnosysadmin 用户账户。以下 SQL 查询可以帮助验证每个副本上的 dmsnosysadmin 账户 SID 值。有关创建用户的更多信息，请参阅 [Microsoft SQL Server 文档](#) 中的 [CREATE USER \(Transact-SQL\)](#)。有关为 Azure SQL 数据库创建 SQL 用户账户的更多信息，请参阅 [活动异地复制](#)。

```
SELECT @@servername servername, name, sid, create_date, modify_date
FROM sys.server_principals
WHERE name = 'dmsnosysadmin';
```

14.在每个副本上授予对 Master 数据库的权限：

```
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.在每个副本上授予对 MSDB 数据库的权限：

```
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.在源数据库上将 db\_owner 角色添加到 dmsnosysadmin。由于数据库已同步，您只能在主副本上添加角色。

```
use <source DB>
GO
EXEC sp_addrolemember N'db_owner', N'dmsnosysadmin'
```

## SQL Server 支持脚本

以下主题介绍如何下载、查看和运行可用于 SQL Server 的各个支持脚本。您还可以了解如何查看脚本输出并将其上传到您的 AWS Support 案例中。

### 主题

- [awsdms\\_support\\_collector\\_sql\\_server.sql 脚本](#)

awsdms\_support\_collector\_sql\_server.sql 脚本

下载 [awsdms\\_support\\_collector\\_sql\\_server.sql](#) 脚本。

#### Note

仅在 SQL Server 2014 及更高版本上运行此 SQL Server 诊断支持脚本。

此脚本收集有关您的 SQL Server 数据库配置的信息。请记住验证脚本的校验和，如果校验和验证成功，请查看脚本中的 SQL 代码，注释掉任何您不愿意运行的代码。在您确定脚本的完整性和内容没有问题之后，可以运行该脚本。

运行适用于本地 SQL Server 数据库的脚本

1. 使用以下 sqlcmd 命令行运行脚本。

```
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
```

指定的 sqlcmd 命令参数如下所示：

- -U – 数据库用户名
- -P – 数据库用户密码。
- -S – SQL Server 数据库服务器名称。
- -y – 从 sqlcmd 实用程序输出的最大列宽度。如果值为 0，则表示列宽度不受限制。
- -i – 要运行的支持脚本的路径，在本例中为  
awsdms\_support\_collector\_sql\_server.sql。
- -o – 输出 HTML 文件的路径，文件名由您指定，其中包含收集的数据库配置信息。

- -d – SQL Server 数据库名称。
2. 脚本完成后，查看输出的 HTML 文件并删除任何您不愿意分享的信息。当您觉得 HTML 可以分享时，请将该文件上传到您的 AWS Support 案例中。有关上传此文件的更多信息，请参阅[使用 AWS DMS 中的诊断支持脚本](#)。

对于 Amazon RDS for SQL Server，您无法使用 sqlcmd 命令行实用程序进行连接，因此请使用以下过程。

#### 运行适用于 RDS SQL Server 数据库的脚本

1. 使用任何允许您以 Master 用户身份连接到 RDS SQL Server 的客户端工具运行脚本，并将输出保存为 HTML 文件。
2. 您可以查看输出 HTML 文件并删除任何不愿意分享的信息。当您觉得 HTML 可以分享时，请将文件上传到您的 AWS Support 案例中。有关上传此文件的更多信息，请参阅[使用 AWS DMS 中的诊断支持脚本](#)。

## 适用于 MySQL 兼容数据库的诊断支持脚本

接下来，您可以找到诊断支持脚本，可用于分析 AWS DMS 迁移配置中的本地数据库或 Amazon RDS for MySQL 数据库。这些脚本可以与源端点或目标端点一起使用。这些脚本均编写为在 MySQL SQL 命令行中运行。

有关安装 MySQL 客户端的信息，请参阅 MySQL 文档中的[安装 MySQL Shell](#)。有关使用 MySQL 客户端的信息，请参阅 MySQL 文档中的[使用 MySQL Shell 命令](#)。

在运行脚本之前，请确保您使用的用户账户具有所需的访问权限，可以访问 MySQL 兼容数据库。使用以下过程创建用户账户并提供运行此脚本所需的最低权限。

#### 设置具有运行这些脚本的最低权限的用户账户

1. 创建要运行脚本的用户。

```
create user 'username'@'hostname' identified by password;
```

2. 在数据库上运行 select 命令以对其进行分析。

```
grant select on database-name.* to username;  
grant replication client on *.* to username;
```

3.

```
grant execute on procedure mysql.rds_show_configuration to username;
```

以下主题介绍如何下载、查看和运行可用于 MySQL 兼容数据库的各个支持脚本。您还可以了解如何查看脚本输出并将其上传到您的 AWS Support 案例中。

## 主题

- [awsdms\\_support\\_collector\\_MySQL.sql 脚本](#)

## awsdms\_support\_collector\_MySQL.sql 脚本

下载 [awsdms\\_support\\_collector\\_MySQL.sql](#) 脚本。

此脚本收集有关您的 MySQL 兼容数据库配置的信息。请记住验证脚本的校验和，如果校验和验证成功，请查看脚本中的 SQL 代码，注释掉任何您不愿意运行的代码。在您确定脚本的完整性和内容没有问题之后，可以运行该脚本。

在使用命令行连接到数据库环境后，运行脚本。

运行脚本并将结果上传到您的支持案例

1. 使用以下 `mysql` 命令连接到数据库。

```
mysql -h hostname -P port -u username database-name
```

2. 使用以下 `mysql source` 命令运行脚本。

```
mysql> source awsdms_support_collector_MySQL_compatible_DB.sql
```

您可以查看生成的报告并删除任何不愿意分享的信息。当您觉得内容可以分享时，请将该文件上传到您的 AWS Support 案例中。有关上传此文件的更多信息，请参阅[使用 AWS DMS 中的诊断支持脚本](#)。

### Note

- 如果您已经拥有具有 [适用于 MySQL 兼容数据库的诊断支持脚本](#) 中描述的所需权限的用户账户，则也可以使用现有的用户账户来运行脚本。

- 请记住在运行脚本之前连接到您的数据库。
- 脚本生成文本格式的输出。
- 请记住安全最佳实践，如果您创建的新用户账户只用于执行此 MySQL 诊断支持脚本，建议您在成功执行该脚本后删除此用户账户。

## PostgreSQL 诊断支持脚本

接下来，您可以找到诊断支持脚本，可用于分析 AWS DMS 迁移配置中的任意 PostgreSQL RDBMS（本地、Amazon RDS 或 Aurora PostgreSQL）。这些脚本可以与源端点或目标端点一起使用。这些脚本均编写为在 `psql` 命令行实用程序中运行。

在运行这些脚本之前，请确保您使用的用户账户具有以下必要权限，可以访问任何 PostgreSQL RDBMS：

- PostgreSQL 10.x 或更高版本 – 具有 `pg_catalog.pg_ls_waldir` 函数执行权限的用户账户。
- PostgreSQL 9.x 或更早版本 – 具有默认权限的用户账户。

我们建议使用具有相应权限的现有账户来运行这些脚本。

如果您需要创建新的用户账户或向现有账户授予权限以运行这些脚本，则可以对基于 PostgreSQL 版本的任何 PostgreSQL RDBMS 执行以下 SQL 命令。

授予账户权限以在 PostgreSQL 数据库版本 10.x 或更高版本中运行这些脚本

- 请执行以下操作之一：
  - 对于新用户账户，请运行以下命令。

```
CREATE USER script_user WITH PASSWORD 'password';  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

- 对于现有用户账户，请运行以下命令。

```
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

授予账户权限以在 PostgreSQL 9.x 或更早版本中运行这些脚本

- 请执行以下操作之一：
  - 对于新用户账户，请使用默认权限运行以下命令。

```
CREATE USER script_user WITH PASSWORD password;
```

- 对于现有用户账户，请使用现有权限。

#### Note

对于 PostgreSQL 9.x 及更早版本的数据库，这些脚本不支持与查找 WAL 大小相关的特定功能。有关更多信息，请联系 AWS Support。

以下主题介绍如何下载、查看和运行可用于 PostgreSQL 的各个支持脚本。它们还描述了如何查看脚本输出并将其上传到您的 AWS Support 案例中。

主题

- [awsdms\\_support\\_collector\\_postgres.sql 脚本](#)

### awsdms\_support\_collector\_postgres.sql 脚本

下载 [awsdms\\_support\\_collector\\_postgres.sql](#) 脚本。

此脚本收集有关您的 PostgreSQL 数据库配置的信息。请记住验证脚本的校验和。如果校验和验证成功，请查看脚本中的 SQL 代码，注释掉任何您不愿意运行的代码。在您确定脚本的完整性和内容没有问题之后，可以运行该脚本。

#### Note

您可以使用 psql 客户端版本 10 或更高版本运行此脚本。

您可以使用以下过程，从数据库环境或从命令行运行此脚本。无论哪种方法，您都可以稍后将文件上传到 AWS Support。



## 运行脚本并将结果上传到您的支持案例

### 1. 请执行以下操作之一：

- 使用以下 `psql` 命令行在数据库环境中运行脚本。

```
dbname=# \i awsdms_support_collector_postgres.sql
```

在随后出现的提示中，输入您要迁移的唯一一个架构的名称。

在随后出现的提示中，输入您已定义用于连接数据库的用户的名称 (*script\_user*)。

- 直接从命令行运行以下脚本。此选项可避免在脚本执行之前出现任何提示。

```
psql -h database-hostname -p port -U script_user -d database-name -f  
awsdms_support_collector_postgres.sql
```

- ### 2. 您可以查看输出 HTML 文件并删除任何不愿意分享的信息。当您觉得 HTML 可以分享时，请将该文件上传到您的 AWS Support 案例中。有关上传此文件的更多信息，请参阅[使用 AWS DMS 中的诊断支持脚本](#)。

## 使用 AWS DMS 诊断支持 AMI

如果您在使用时遇到与网络相关的问题 AWS DMS，您的支持工程师可能需要有关您的网络配置的更多信息。我们希望确保 S AWS support 在尽可能短的时间内获得尽可能多的所需信息。因此，我们开发了一个带有诊断工具的预构建的 Amazon EC2 AMI，用于测试您的 AWS DMS 网络环境。

该亚马逊机器映像 (AMI) 上安装的诊断测试包括：

- Virtual Private Cloud (VPC)
- 网络数据包丢失
- 网络延迟
- 最大传输单元 (MTU) 大小

### 主题

- [启动新的 AWS DMS 诊断 Amazon EC2 实例](#)
- [创建 IAM 角色](#)
- [运行诊断测试](#)

- [后续步骤](#)
- [按区域列出的 AMI ID](#)

#### Note

如果您的 Oracle 源遇到性能问题，则可以评估 Oracle 重做或存档日志的读取性能，以寻找提高性能的方法。有关更多信息，请参阅[评估 Oracle 重做或存档日志的读取性能](#)。

## 启动新的 AWS DMS 诊断 Amazon EC2 实例

在本节中，您将启动新的 Amazon EC2 实例。有关如何启动 Amazon EC2 实例的信息，请参阅[Amazon EC2 用户指南](#)中的[Amazon EC2 Linux 实例入门](#)。

使用以下设置启动 Amazon EC2 实例：

- 对于应用程序和 OS 映像（亚马逊机器映像），请搜索 DMS-DIAG-AMI AMI。如果您已登录控制台，则可以使用[此查询](#)搜索 AMI。有关您所在地区的 AWS 诊断 AMI 的 AMI ID，请参阅[按区域列出的 AMI ID](#)以下内容。
- 对于实例类型，建议您选择 t2.micro。
- 对于网络设置，请选择您的复制实例使用的同一个 VPC。

实例激活之后，连接到该实例。有关连接到 Amazon EC2 Linux 实例的信息，请参阅[连接到 Linux 实例](#)。

## 创建 IAM 角色

如果您想使用所需的最低权限在复制实例上运行诊断测试，请创建一个使用以下权限策略的 IAM 角色：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "dms:DescribeEndpoints",
```

```
        "dms:DescribeTableStatistics",
        "dms:DescribeReplicationInstances",
        "dms:DescribeReplicationTasks",
        "secretsmanager:GetSecretValue"
    ],
    "Resource": "*"
}
]
```

将该角色附加到新的 IAM 用户。有关创建 IAM 角色、策略和用户的信息，请参阅 [IAM 用户指南](#) 中的以下部分：

- [IAM 入门](#)
- [创建 IAM 角色](#)
- [创建 IAM 策略](#)

## 运行诊断测试

创建 Amazon EC2 实例并连接到该实例后，请执行以下操作，对您的复制实例运行诊断测试。

1. 配置 C AWS LI：

```
$ aws configure
```

提供您要用于运行诊断测试的 AWS 用户帐户的访问凭证。向 VPC 和复制实例提供区域。

2. 显示您所在地区的可用 AWS DMS 任务。将示例区域替换为您的区域。

```
$ dms-report -r us-east-1 -l
```

此命令会显示任务的状态。





```

#####
#
#
#   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-04829b2beb8214602-postgres-dev-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
    Looks good with no issue. <<<<<

3.) Check Target endpoint (rds-postgres-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
    Looks good with no issue. <<<<<

==== End network packet check ====

==== Network Latency Check ====
1.) Check DMS MetaData Service
>>>>Result: round-trip min/avg/max = 0.4/0.4/0.5 ms
    Looks good with no issue. <<<<<

2.) Check Source endpoint (dms-04829b2beb8214602-postgres-dev-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: round-trip min/avg/max = 1.0/1.1/1.2 ms
    Looks good with no issue. <<<<<

3.) Check Target endpoint (rds-postgres-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: round-trip min/avg/max = 1.4/1.4/1.5 ms
    Looks good with no issue. <<<<<

==== End network latency check ====

==== Network MTU Check ====
1.) Check DMS MetaData Service
>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<

2.) Check Source endpoint (dms-04829b2beb8214602-postgres-dev-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<

3.) Check Target endpoint (rds-postgres-instance-1.cucdvzaur7nk.us-east-2.rds.amazonaws.com:5432)
>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<

==== End network MTU check ====

```

**Perform AMI Diag EC2 VPC Check****Perform Network Packet Test****Returns Test Results and Recommendation****Perform Network Latency Test****Perform Network Maximum Transmission Unit (MTU) Check**

## 后续步骤

以下部分介绍基于网络诊断测试结果的故障排除信息：

### VPC 测试

此测试验证诊断 Amazon EC2 实例与复制实例位于同一 VPC 中。如果诊断 Amazon EC2 实例与您的复制实例不在同一 VPC 中，请将其终止，然后在正确的 VPC 中重新创建。Amazon EC2 在创建之后无法更改其 VPC。

### 网络数据包丢失测试

此测试向以下端点发送 10 个数据包并检查数据包是否丢失：

- 端口 80 上的 AWS DMS Amazon EC2 元数据服务
- 源端点
- 目标端点

所有数据包都应成功到达。如果有任何数据包丢失，请咨询网络工程师以确定问题并寻找查找方案。

### 网络延迟测试

此测试将 10 个数据包发送到之前测试所用的同一个端点，并检查数据包延迟。所有数据包的延迟应小于 100 毫秒。如果任何数据包的延迟超过 100 毫秒，请咨询网络工程师以确定问题并寻找解决方案。

### 最大传输单元 ( MTU ) 大小测试

此测试在之前测试所用的同一个端点上，使用 Traceroute 工具来检测 MTU 的大小。测试中的所有数据包都应具有相同的 MTU 大小。如果任何数据包的 MTU 大小不同，请咨询系统专家以确定问题并寻找解决方案。

## 按区域列出的 AMI ID

要查看您 AWS 所在地区可用的 DMS 诊断 AMI 列表，请运行以下 CL AWS I 示例。

```
aws ec2 describe-images --owners 343299325021 --filters "Name=name, Values=DMS-DIAG*"
--query "sort_by(Images, &CreationDate)[-1].[Name, ImageId, CreationDate]" --output
text
```

如果输出未显示任何结果，则表示 DMS 诊断 AMI 在您所在的 AWS 地区不可用。解决方法是按照以下步骤从其他区域复制诊断 AMI。有关更多信息，请参阅[复制 AMI](#)。

- 在可用区域启动实例。
- 创建镜像。图片将归您所有。
- 将 AMI 复制到您所在的地区，例如中东 (UAE) 地区。
- 在您的本地区域启动实例。



## AWS DMS 参考

在此参考部分中，您可以查找在使用 AWS Database Migration Service (AWS) 时可能需要的其他信息，包括数据类型转换信息。

AWS DMS 在您执行同构数据库迁移（源和目标使用相同的引擎类型）时保留数据类型。当您执行异构迁移（从一种数据库引擎类型迁移到另一种数据库引擎）时，数据类型将转换为中间数据类型。要了解数据类型如何在目标数据库上显示，请参阅源和目标数据库引擎的数据类型表。

在迁移数据库时，了解有关数据类型的几个重要事项：

- FLOAT 数据类型本身是一个近似值。在插入特定 FLOAT 值时，它可能会在数据库中以不同方式呈现。这一区别是因为 FLOAT 不是准确的数据类型，如 NUMBER 或 NUMBER(p,s) 之类的十进制数据类型。因此，存储在数据库中的内部 FLOAT 值可能与您插入的值不同。因此，迁移后的 FLOAT 值可能与源数据库中的值不完全匹配。

有关此问题的更多信息，请参阅以下文章：

- Wikipedia 中的 [IEEE 浮点](#)
- Microsoft Learn 上的 [IEEE 浮点表示形式](#)
- Microsoft Learn 上的 [为什么浮点数可能丢失精度](#)

### 主题

- [AWS Database Migration Service 的数据类型](#)


## AWS Database Migration Service 的数据类型

AWS Database Migration Service 使用内置的数据类型，将数据从一个数据库引擎类型迁移到另一个数据库引擎类型。下表显示了内置的数据类型及其说明。

AWS DMS 数据类型	描述
STRING	一个字符串。
WSTRING	一个双字节字符串。
BOOLEAN	布尔值。

AWS DMS 数据类型	描述
BYTE	一个二进制数据值。
DATE	一个日期值：年、月、日。
TIME	一个时间值：小时、分钟、秒。
DATETIME	一个时间戳值：年、月、日、小时、分钟、秒、小数秒。最多具有 9 位数的小数秒。支持以下格式：YYYY:MM:DD HH:MM:SS.F(9)。对于 Amazon S3 Select 和 Amazon S3 Glacier Select，DATETIME 数据类型的格式不同。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中 <a href="#">支持的数据类型</a> 中的 timestamp 基元数据类型。
INT1	一个有符号的 1 字节整数。
INT2	一个有符号的 2 字节整数。
INT4	一个有符号的 4 字节整数。
INT8	一个有符号的 8 字节整数。
NUMERIC	带固定精度和小数位数的精确数字值。
REAL4	一个单精度浮点值。
REAL8	一个双精度浮点值。
UINT1	一个无符号的 1 字节整数。
UINT2	一个无符号的 2 字节整数。
UINT4	一个无符号的 4 字节整数。
UINT8	一个无符号的 8 字节整数。
BLOB	二进制大对象。

AWS DMS 数据类型	描述
CLOB	字符大对象。
NCLOB	本机字符大对象。

 Note

AWS DMS 无法将任何 LOB 数据类型迁移到 Apache Kafka 终端节点。

# AWS DMS 发行说明

接下来，您可以找到当前和以前版本的 AWS Database Migration Service (AWS DMS) 的发行说明。

AWS DMS 当您为复制实例启用自动版本升级时，不会区分主要版本和次要版本。如果复制实例的版本已过时，DMS 会在维护时段内自动升级该版本。

请注意，要手动将复制实例的版本（使用 API 或 CLI）从 3.4.x 版本升级到 3.5.x，必须将参数设置为 `AllowMajorVersionUpgrade true`。有关 `AllowMajorVersionUpgrade` 参数的信息，请参阅 DMS API 文档 [ModifyReplicationInstance](#) 中的。

## Note

的当前默认引擎版本 AWS DMS 为 3.5.1。

下表显示了 DMS 有效版本的以下日期：

- 该版本的发布日期
- 在此日期之后您无法使用该版本创建新实例
- DMS 自动更新该版本实例的日期（停产日期）

版本	发行日期	没有新的实例日期	EOL 日期
3.5.3	2024 年 5 月 17 日	2025 年 8 月 31 日	2025 年 10 月 31 日
3.5.2	2023 年 10 月 29 日	2025 年 3 月 30 日	2025 年 4 月 29 日
3.5.1	2023 年 6 月 30 日	2024 年 11 月 30 日	2025 年 1 月 30 日
3.4.7	2022 年 5 月 31 日	2024 年 7 月 30 日	2024 年 8 月 29 日
3.4.6	2021 年 11 月 30 日	2024 年 5 月 26 日	2024 年 6 月 27 日

# AWS Database Migration Service 3.5.3 发行说明

## AWS DMS 3.5.3 中的新功能

新功能或增强功能	描述
增强了 PostgreSQL 源端点以支持 Babelfish	AWS DMS 增强了其 PostgreSQL 源端点以支持 Babelfish 数据类型。有关更多信息，请参阅 <a href="#">将 PostgreSQL 数据库作为 AWS DMS 源</a> 。
支持 S3 Parquet 作为来源	AWS DMS 支持 S3 Parquet 作为来源。有关更多信息，请参阅 <a href="#">使用亚马逊 S3 作为来源 AWS DMS</a>
支持 PostgreSQL 16.x	AWS DMS 支持 PostgreSQL 版本 16.x。有关更多信息，请参阅 <a href="#">将 PostgreSQL 数据库作为 AWS DMS 源</a> 和 <a href="#">将 PostgreSQL 数据库作为 AWS Database Migration Service 的目标</a> 。
提高了 Oracle 到 Amazon Redshift 的满负荷迁移的吞吐量	AWS DMS 对于从 Oracle 到 Amazon Redshift 的满负荷迁移，Serverless 显著提高了吞吐量性能。有关更多信息，请参阅 <a href="#">提高了 Oracle 到 Amazon Redshift 的满负荷迁移的吞吐量</a> 。

AWS DMS 版本 3.5.3 包括以下已解决的问题：

2024 年 5 月 17 日发布的 DMS 3.5.3 版本中已解决的问题

解决的问题	描述
数据验证覆盖函数	修复了数据验证功能的一个问题，即在表映射 <code>override-validation-function</code> 中将规则操作设置为时，DMS 不支持源筛选。
MySQL 来源 CDC 错误	修复了 MySQL 作为源的问题，即使用 UTF16 编码，CDC 迁移会失败。
数据验证排序规则的差异	修复了使用列筛选时 DMS 无法正确应用 <code>HandleCollationDiff</code> 任务设置的数据验证功能的问题。
数据验证任务暂停。	修复了数据验证功能的一个问题，即 DMS 任务会挂起并出现“target 为空”错误。

解决的问题	描述
PostgreSQL 到 PostgreSQL 复制中的任务失败。	修复了 PostgreSQL 到 PostgreSQL 迁移的问题，即在 CDC 复制期间，在向目标中插入 LOB 数据时，DMS 任务会失败。
以 PostgreSQL 为源的数据丢失	修复了 PostgreSQL 作为数据源的问题，该问题在某些边缘情况下会发生数据丢失。
MySQL 5.5 来源 CDC 错误	修复了 MySQL 作为来源的问题，在 MySQL 版本 5.5 中，CDC 复制会失败。
甲骨文来源 IOT 表问题。	修复了 Oracle 作为来源的问题，即在所有列上都启用了补充日志的情况下，DMS 无法正确复制物联网表的 UPDATE 语句。
MySQL 源代码 LOBS	修复了 MySQL 到 Redshift 迁移的问题，即由于 LOB 超过 Redshift 允许的最大大小，DMS 任务会失败。
的验证问题 SkipLobColumns	修复了数据验证功能的一个问题，即当主键位于源表的最后一列 SkipLobColumns = true 时，DMS 任务会失败。
跳过唯一密钥所在位置的验证 null	修复了数据验证功能的一个问题，即 DMS 无法正确跳过具有空唯一键的行。
针对 Oracle COLLATE 操作员的数据验证进行了改进。	修复了数据验证功能的问题，即在 12.2 之前的 Oracle 版本上，验证会因语法错误而失败。
满载期间的错误处理	修复了 PostgreSQL 作为目标的问题，即在无效数据导致表错误后，任务在满载阶段会挂起。
重新验证 CDC 仅限验证的任务	增强了数据验证功能，允许对仅限 CDC 验证的任务进行重新验证。
S3 作为目标 CdcMaxBatchInterval Out of Memory 问题	修复了 S3 作为目标的问题，即 CdcMaxBatchInterval 设置后出现内存不足的情况，DMS 任务会失败。
甲骨文源码驱动程序	将 DMS Oracle 源驱动程序从 v12.2 升级到 19.18 版本。
带有 SQL Server 源代码的 LOB 截断警告	增强了 SQL Server 的日志记录，以显示 CDC 期间 LOB 截断的警告。

解决的问题	描述
甲骨文二进制阅读器增强	<p>增强了 Oracle 源代码二进制读取器以支持以下内容：</p> <ul style="list-style-type: none"> <li>• Big Endian 平台</li> <li>• 使用 HCC 压缩的并行 DML 提示</li> <li>• 启用 Golden Gate 的高级甲骨文压缩</li> </ul>

## AWS Database Migration Service 3.5.2 版本说明

### AWS DMS 3.5.2 中的新功能

新功能或增强功能	描述
Redshift 数据验证	AWS DMS 现在支持验证 Redshift 目标中的数据。
支持将 Microsoft SQL Server 版本 2022 作为源和目标。	AWS DMS 现在支持使用微软 SQL Server 版本 2022 作为源和目标。
将 IBM Db2 LUW 作为目标	AWS DMS 现在支持 IBM Db2 LUW 作为目标。使用 AWS DMS，您现在可以执行从 IBM Db2 LUW 到 IBM Db2 LUW 的实时迁移。

AWS DMS 版本 3.5.2 包括以下已解决的问题：

在 2024 年 4 月 29 日的 DMS 3.5.2 维护版本中解决的问题

解决的问题	描述
IBM Db2 的目标是分段满负荷	增加了对以 IBM Db2 为目标的分段满载的支持。
将亚马逊 Timestream 作为目标设置	增强了对 Timestream 作为目标的无效时间戳设置和不支持的表操作的处理。
列筛选器导致任务崩溃	修复了在 DMS 使用转换规则动态添加的列上使用筛选器时任务崩溃的问题。

解决的问题	描述
记录交易交换文件读取	添加了日志记录以显示 DMS 何时从事务交换文件中读取数据。
将 S3 作为目标 CdcInsertsAndUpdates	修复了 S3 作为目标时任务在当前 true 和 CdcInsertsAndUpdates PreserveTransactions 现在时会崩溃的问题 true。
源过滤器负面运算符	修复了如果同一列定义了转换规则，则源过滤运算符设置为负数运算符时行为不正确的问题。
添加了 DMS 暂停从源端读取时的日志记录	增强了日志功能，可显示 DMS 何时暂停从源端读取以提高性能。
带有转义字符的源过滤器	修复了源过滤器的问题，即在 CDC 期间，DMS 会将转义字符应用于新创建的表。
PostgreSQL 作为目标，错误地复制了删除内容	修复了 PostgreSQL 作为目标时，DMS 会将删除内容复制为空值的问题。
Oracle 作为源日志记录的改进	增强了 Oracle 作为来源的日志记录，可删除多余的错误代码。
改进了 XMLTYPE 限制的日志记录	改进了 Oracle 作为来源的日志记录，以显示 DMS 不支持该 XMLTYPE 数据类型的完整 LOB 模式。
MySQL 数据丢失	修复了 MySQL 作为目标的问题，即损坏的列元数据可能导致任务崩溃或数据丢失。
应用于新列的筛选器	修复了满载期间 DMS 会忽略转换规则添加到新列的过滤器的问题。
以 S3 为目标：验证问题	修复了 S3 作为目标的问题，即在迁移具有不同验证分区定义的多个表时，数据验证会失败。
仅限 CDC 的任务崩溃	修复了仅限 CDC 的任务在处于状态时 TaskRecoveryTableEnabled 任务会崩溃的问题。true
MySQL 到 MariaDB 不兼容的排序规则	修复了 MySQL 到 MariaDB 迁移时出现的问题，即 DMS 无法使用排序规则迁移 MySQL v8 表。tf8mb4_0900_ai_ci



解决的问题	描述
任务崩溃 BatchApplyEnabled	修复了 Batch Apply 功能在某些条件下任务会失败的问题。
亚马逊 DocumentDB 中的非 UTF-8 角色	为亚马逊 DocumentDB 终端节点添加了对非 UTF-8 字符的支持。
Batch Apply 任务崩溃	修复了 Batch Apply 功能的一个问题，即在复制大型交易时，DMS 任务会崩溃。
Db2 事务回滚处理	修复了 Db2 作为源的问题，即尽管已在源上回滚，但 DMS 仍会将复制 INSERT 到目标。
使用源过滤器进行验证	修复了验证不符合源过滤器的问题。

## AWS Database Migration Service 3.5.1 发行说明

下表显示了 AWS Database Migration Service (AWS DMS) 版本 3.5.1 中引入的新功能和增强功能。

新功能或增强功能	描述
支持 PostgreSQL 15.x	AWS DMS 版本 3.5.1 支持 PostgreSQL 版本 15.x。有关更多信息，请参阅 <a href="#">将 PostgreSQL 作为源</a> 和 <a href="#">将 PostgreSQL 作为目标</a> 。
支持带有分片集合的 Amazon DocumentDB 弹性集群	AWS DMS 版本 3.5.1 支持带有分片集合的亚马逊 DocumentDB 弹性集群。有关更多信息，请参阅 <a href="#">将 Amazon DocumentDB 作为 AWS Database Migration Service 的目标</a> 。
Redshift Serverless 作为目标	增加了将 Amazon Redshift Serverless 作为目标端点的支持。有关更多信息，请参阅 <a href="#">将 Amazon Redshift 数据库作为 AWS Database Migration Service 目标</a> 。
Babelfish 端点设置	增强了 PostgreSQL 目标端点设置，用于提供 Babelfish 支持。有关更多信息，请参阅 <a href="#">将 PostgreSQL 数据库作为 AWS Database Migration Service 的目标</a> 。

新功能或增强功能	描述
Oracle 源未完成事务	AWS DMS 3.5.1 改进了从 Oracle 来源的“起始位置”启动仅限 CDC 的任务时处理未结交易的方法。有关更多信息，请参阅 <a href="#">使用 Oracle 作为来源时的终端节点设置 AWS DMS</a> 部分中的 <code>OpenTrans actionWindow</code> 。
以亚马逊 Timestream 为目标	支持使用亚马逊 Timestream 作为目标终端节点。有关更多信息，请参阅 <a href="#">将 Amazon Timestream 作为 AWS Database Migration Service 的目标</a> 。

AWS DMS 版本 3.5.1 包括以下已解决的问题：

解决的问题	描述
Oracle 是不断增长的非活动会话的来源	修复了 Oracle 来源的一个问题，即仅限 CDC 的任务的非活动会话不断增加，从而导致以下异常： <code>ORA-00020: maximum number of processes exceeded on the source database</code>
将更新更改复制到 DocumentDB	修复了 DocumentDB 作为目标的问题，即在某些情况下无法正确复制 UPDATE 语句。
仅限验证的任务	改进了数据验证功能的错误处理，当仅限验证的任务禁用数据验证时，可以正确地使任务失败。
连接终止后的 Redshift 复制	修复了 Redshift 目标的一个问题，即当目标在连接终止后 <code>ParallelApplyThreads</code> 设置大于零时，DMS 任务不会重试对目标应用更改，这会导致数据丢失。
MySQL 文本到中等文本的复制	修复了 MySQL 到 MySQL 在全 LOB 模式下复制中文本数据类型的问题。
无法使用轮换密钥复制 CDC 任务	修复了 DMS 任务 <code>BatchApplyEnabled</code> 设置为在 Secrets Manager 轮换密码后停止复制数据的问题。 <code>true</code>
mongodb/documentDB 分	修复了 MongoDB /docDB 源的一个问题，即当主键列包含较大的值时，范围分割无法正常工作。

解决的问题	描述
未绑定数值的 Oracle 数据验证	修复了 Oracle 目标的一个问题，即在数据验证STRING期间，DMS 会将未绑定数据类型的值识别NUMERIC为 a。
SQL 服务器数据验证	修复了 SQL Server 端点存在的问题，其中 DMS 数据验证会生成无效 SQL 语句。
MongoDB 自动分段	改进了在将 MongoDB 作为源时，从中并行迁移文档时自动对数据分区的功能。
Amazon S3 Apache Parquet 格式	修复了一个问题，以便使用 Python 和 Apache Arrow C++ 查看以 S3 为目标写入的 Apache Parquet 文件。
PostgreSQL 作为源时的 DDL 处理	修复了 PostgreSQL 作为源时未正确忽略不支持的 DDL 操作的问题。
PostgreSQL timestamp tz 数据错误	修复了 PostgreSQL 迁移到 PostgreSQL 时出现的问题，在该问题中，CDC 期间启用了批量应用时，带有时区数据的时间戳无法正确迁移。
Oracle 到 PostgreSQL 的验证失败	修复了 Oracle 迁移到 PostgreSQL 时，NUMERIC(38,30) 数据类型的数据验证失败的问题。
Oracle 扩展数据类型错误	修复了 Oracle 源中扩展 varchar 数据类型被截断的问题。
组合筛选器运算符	修复了列筛选功能中无法将 null 列运算符与其他运算符类型组合使用的问题。
过多日志记录导致的 CDC 延迟。	修复了 PostgreSQL 源中 pglogical 插件警告的过多日志记录会导致源 CDC 延迟的问题。
创建表 DDL 的双向复制处理	修复了从 PostgreSQL 到 PostgreSQL 的双向复制时无法正确复制创建表 DDL 更改的问题。
使用筛选器时 CDC 失败	修复了 CDC 复制失败时的筛选功能问题。
Kafka 端点的证书颁发机构主机名验证	通过添加用于禁用证书颁发机构的主机名验证 ( SslEndpointIdentificationAlgorithm ) 的选项，增强了 Kafka 端点的功能。

解决的问题	描述
IBM Db2 LUW 验证	修复了在数据验证期间未正确处理 Db2 LUW 源日期、时间戳和时间数据类型的问题。
S3 验证	修复了从 Db2 LUW 迁移到 S3 时，验证功能无法正确处理 timestamp(0) 数据类型的问题。
DMS 任务重启失败	修复了 PostgreSQL 源代码中的一个问题，AWS DMS 即在使用 pglogical 插件时，任务无法重启且无法使用关系事件。
SQL Server 的 HIERARCHY 数据类型的验证	修复了 SQL Server 源中 HIERARCHY 数据类型验证会失败的问题。
SQL Server 中带有控制字符的字符串	修复了 SQL Server 源中无法正确复制带有控制字符的字符串的问题。
Redshift 和 Secrets Manager	修复了 Redshift 目标中在使用 Secrets Manager 时测试端点会失败的问题。
MySQL ParallelLoadThreads 设置不一致	修复了 MySQL 目标中更改任务设置后 ParallelLoadThreads 设置未正确保留的问题。
PostgreSQL 到 Oracle 数据类型映射时出错	修复了从 PostgreSQL 迁移到 Oracle 时，从 TEXT 数据类型复制到 VARCHAR2(2000) 数据类型会失败的问题。
Oracle 到 PostgreSQL 的数据验证	修复了从 Oracle 迁移到 PostgreSQL，当 NULL 字符复制为 SPACE 字符时，数据验证会出现误报的问题。
AlwaysOn 配置中的 SQL Server 源	修复了 AlwaysOn 配置中 SQL Server 源代码的问题，即当副本名称与实际服务器名称不完全匹配时，AWS DMS 任务会失败。
Oracle 源端点测试失败	修复了 Oracle 源的一个问题，即在检索 Oracle 会话 ID (SID) 时，由于权限不足，AWS DMS 端点连接测试会失败。
CDC 不选取新表	修复了仅 CDC 任务在某些情况下，任务启动后在源上创建的表没有复制的问题。

解决的问题	描述
将 Oracle 作为源时的未完成事务	针对 Oracle 源，改进了从起始位置启动仅 CDC 任务时处理未完成事务的方法。
缺失数据问题	修复了在应用缓存更改后停止的任务（StopTaskCachedChangesApplied 选项设置为 true），在恢复任务时丢失数据的问题。如果由于源上有大量更改而导致对 AWS DMS 复制实例磁盘的缓存更改 AWS DMS 持续存在，则很少会出现此问题。
扩展数据类型的数据验证问题	修复了 PostgreSQL 到 Oracle 的数据验证问题，在该问题中，扩展数据类型的验证失败。
字符编码不一致导致的数据验证问题	修复了 SQL Server 到 PostgreSQL 的数据验证问题，在该问题中，当源和目标之间的字符编码不一致时，验证失败。
数据验证问题 ORA-01455	修复了 PostgreSQL integer 映射到 Oracle number(10) 时，验证期间出现 ORA-01455 错误的问题。
SQL Server IDENTITY 支持	修复了 SQL Server 到 SQL Server 的数据复制问题，在该问题中，当目标列具有 IDENTITY 属性时，迁移身份列会失败。
ALTER 语句的字符集问题	修复了 MySQL 到 MySQL 复制时在 CDC 期间迁移 ALTER 语句时将字符集 AWS DMS 更改为 UTF16 的问题。
PostgreSQL 到 Redshift 空间数据类型支持	增加了从 PostgreSQL 迁移到 Amazon Redshift 时对 spatial 数据类型的支持。
.parquet 文件的 GZIP 压缩	修复了 AWS DMS 无法生成以 S3 为目标的 GZIP 压缩的 .parquet 文件的问题。
MongoDB/DocDB 源迁移	修复了 AWS DMS 无法从 MongoDB 源迁移部分分区的问题。
表统计数据问题	修复了当复制实例上的至少一个任务包含的表超过 1001 个时，不显示表统计数据的问题。
IBM Db2 LUW 版本 10.1.0 及更低版本的表暂停	修复了 Db2 LUW 源的一个问题，在该问题中，当源数据库版本为 10.1.0 或更低版本时，表迁移会暂停并出现错误 TYPESTRIN GUNITS is not valid 。

解决的问题	描述
MongoDB 分区问题	修复了 MongoDB/DocDB 中源分区缺少一个或多个区段的问题。
MongoDB 分区问题	修复了由于类型转换错误而导致基于具有 NumberLong () 类型的列进行分割失败的问题。
MongoDB 分区问题	提高了以 MongoDB 为源的大型数据集的自动分段性能。
MongoDB 驱动程序版本	将 MongoDB 驱动程序降级到 1.20.0，以继续支持 MongoDB 3.6 及更低版本。
Amazon S3 Apache Parquet 时间戳数据类型	修复了 Amazon S3 实木复合地板目标的问题。AWS DMS 现在将 format 参数设置为，isAdjustedToUTC true 以匹配先前版本中的行为 AWS DMS。
Amazon Redshift 作为目标复制命令	修复了将 Amazon Redshift 作为目标，将数据从 Amazon S3 复制到 Amazon Redshift 时，大型表的复制命令会失败的问题。
PostgreSQL 几何体数据类型	修复从 PostgreSQL 迁移到 PostgreSQL 时，在大型几何体数据类型上迁移失败的问题。
Oracle 到 PostgreSQL XML	修复了从 Oracle 复制到 PostgreSQL 时，迁移会在 XML 上增加额外空间的问题。
在支持的引擎中更新目标检查点	AWS DMS 现在更新目标数据库 awsdms_txn_state 表中的目标检查点。
MongoDB/DocDB 记录发送到错误的集合	修复了 MongoDB/DocDB 中数据发送到错误目标集合的问题。
Oracle source 使用 EscapeCharacter 端点设置来选择新的表	修复 AWS DMS 了 Oracle 源的一个问题，即在设置 EscapeCharacter 终端节点设置时任务停止并恢复时，只有在任务停止和恢复时，才会选择新的表进行复制。
CDC 恢复检查点	修复了在目标数据存储和 AWS DMS 控制台之间观察到的 CDC 恢复检查点不一致的问题。
CDC 仅验证任务	修复了 CDC 仅验证任务的问题，该问题导致即便任务中的所有表都遇到故障，任务也不会失败。

解决的问题	描述
源或目标出现连接问题时的验证行为	修复了数据验证中的一个问题，即在连接断开时 AWS DMS 会暂停源或目标上的表。
Oracle 迁移到 PostgreSQL 的数据验证误报	修复了 Oracle 到 PostgreSQL 数据验证 AWS DMS 中报告误报的问题。由于除了 VARCHAR 之外，基于文本的数据类型没有考虑源 NULL 字符在目标上的表示形式的差异，因而会产生误报。
Oracle 迁移到 PostgreSQL 的数据截断	修复了 Oracle 作为源和 PostgreSQL 作为目标的问题，当 Oracle 的 NLS_NCHAR_CHARACTERSET 设置为 AL16UTF16 时，AWS DMS 会截断 NVARCHAR 列的数据。
数据验证错误	修复了数据验证中，同时使用源筛选和添加列转换规则时会引发 unable to create where filter clause 错误的问题。
Redshift 目标错误处理	修复了 Redshift 作为目标时，当 CDC 任务将 ParallelApplyThreads 任务设置的值设置为大于零时，错误处理无法按配置运行的问题。
Oracle 作为源时通信失败	修复了 Oracle 作为源时，任务保持为 RUNNING 状态，但在通信失败后无法迁移任何数据的问题。
使用列筛选器时 CDC 表暂停	修复了完全加载 + CDC 任务中，应用列筛选器时，在 CDC 阶段会暂停表的问题。
S3 作为目标时特殊字符的数据验证失败	修复了 S3 目标数据验证中，如果表名中包含除下划线之外的特殊字符，任务将失败的问题。
MongoDB 源中的完全加载和 CDC 故障	修复了 MongoDB 作为源时，迁移大型集合的缓存事件处理期间，完全加载 + CDC 任务会失败的问题。
BatchApplyEnabled 设置为 true 时的升级问题	修复了任务设置为 true 的 BatchApplyEnabled 任务在某些情况下从 AWS DMS 版本 3.4.6 迁移到 3.5.1 后会失败的问题。
具有区分大小写归类的 SQL Server AlwaysOn 源代码	修复了 SQL Server AlwaysOn 作为源时任务因区分大小写的排序规则而失败的问题。

解决的问题	描述
MySQL 源中的任务挂起	修复了将 MySQL 作为源时，源配置不正确时任务会挂起而不是失败的问题。
S3 源中的完全加载任务失败	修复了从 3.4.6 或 3.4.7 版本升级到 3.5.1 AWS DMS 版本后，将 S3 作为源的问题，任务在恢复时会失败。
将 CaptureDDLs 设置为 false 时的 PostgreSQL 源	修复了将 PostgreSQL 作为源时，将 CaptureDDLs 端点设置为 false 时未正确处理 DDL 的问题。
Oracle 源中任务在恢复期间崩溃	修复了 Oracle 作为源时，由于列名中的数据不正确，会导致任务在恢复时崩溃的问题。
MySQL 源中的 LOB 查找失败	修复了将 MySQL 作为源，当 ParallelApplyThreads 任务设置被设置为大于零的值时，LOB 查找会失败的问题。
SQL 服务器源中的不合逻辑的 LSN 错误	修复了将 SQL Server 作为源的问题，即从 AWS DMS 版本 3.4.7 升级到版本 3.5.1 后，任务会因 illogical LSN sequencing state error 错误而失败。
PostgreSQL 源和 pglogical	修复了 PostgreSQL 作为源时的问题，在该问题中，使用 pglogical 插件的任务，在任务停止时、从选择规则中移除表时、任务恢复时以及对已删除的表进行更改时会失败。
Aurora MySQL 恢复检查点不正确。	修复了将 Aurora MySQL 作为源时出现的问题，即，由于 Aurora 失效转移或 Aurora 源停止和启动，会保存不正确的恢复检查点。
SQL Server 作为源时任务崩溃。	修复了 SQL Server 作为源时出现的问题，即，当 Safeguard Policy 设置为 RELY_ON_SQL_SERVER_REPLICATION_AGENT 时，任务会崩溃。
MySQL 作为目标时数据类型转换不正确	修复了 MySQL 作为目标时出现的问题，即，由于批处理应用阶段的数据类型转换不正确，CDC 复制会失败。
将 PostgreSQL 作为源时，如果将 CaptureDDLs 设置为 false，任务将失败。	修复了 PostgreSQL 作为源时出现的问题，即，在 CaptureDDLs 端点设置为 false 时，由于将 DDL 视为 DML，任务会失败。



解决的问题	描述
MongoDB 空集合崩溃	修复了 MongoDB 作为源时，由于空集合而导致任务崩溃的问题。
Redshift 作为目标时完全加载任务崩溃	修复了 Redshift 作为目标时出现的问题，即，启用恢复检查点控制表后，任务在完全加载阶段会崩溃。
S3 到 S3 - 无数据移动。	修复了 S3 到 S3 复制的问题，AWS DMS 如果未指定，bucketFolder 则无法复制数据。
GlueCatalogGeneration 设置为 true 时 CDC 延迟	修复了 S3 作为目标时出现的问题，即，GlueCatalogGeneration 设置为 true 时会出现延迟过长。
Oracle 作为目标时数据截断	修复了 Oracle 作为目标会 AWS DMS 截断 VARCHAR2 列中的数据的问题。
PostgreSQL 下划线通配符行为	修复了 PostgreSQL 作为源时出现的问题，即，选择规则中通配符“_”的行为与文档所述不符。
PostgreSQL 作为源时出现空 WAL 标头问题。	修复了 PostgreSQL 作为源时出现的问题，即，由于从复制时隙收到空 WAL 标头导致任务失败。
MySQL 或 MariaDB 作为源时出现压缩二进制日志	修复了 MySQL 和 MariaDB 作为源代码的问题，即在 AWS DMS 检测到 BINLOG 压缩时未发出正确的错误消息。
S3 数据验证特殊字符	改进了 S3 数据验证，以处理主键列和非主键列中的特殊字符。
Redshift 作为目标时出现误导性任务日志条目。	修复了 Redshift 作为目标时出现的问题，即，任务日志中出现误导性条目，报告对 UPDATES 和 DELETES 应用批处理语句失败。
SQL Server 到 S3 迁移任务崩溃。	修复了 SQL Server 到 S3 迁移的问题，即，在应用缓存更改时任务崩溃。
批量应用时出现缺少数据错误。	修复了批量应用功能的问题，即，应用批处理时出错会导致数据丢失。

# AWS Database Migration Service 3.5.0 测试版发行说明

## Important

AWS DMS 3.5.0 是复制实例引擎的测试版。AWS DMS 支持此版本与之前的所有版本相同。但我们建议您在将 AWS DMS 3.5.0 Beta 版用于生产目的之前先对其进行测试。

下表显示了 AWS Database Migration Service (AWS DMS) 版本 3.5.0 测试版中引入的新功能和增强功能。

新功能或增强功能	描述
Oracle 和 Microsoft SQL Server 的 Time Travel	现在，您可以通过 DMS 支持的 Oracle、Microsoft SQL Server 和 PostgreSQL 源端点以及 DMS 支持的 PostgreSQL 和 MySQL 目标端点在所有 AWS 区域使用时空旅行。
S3 验证	AWS DMS 现在支持验证 Amazon S3 目标终端节点中复制的数据。有关验证 Amazon S3 目标数据的信息，请参阅 <a href="#">Amazon S3 目标数据验证</a> 。
Glue 目录集成	AWS Glue 是一项提供简单数据分类方法的服务，由名为 AWS Glue Data Catalog 的元数据存储库组成。现在，您可以将 AWS Glue Data Catalog 与您的 Amazon S3 目标终端节点集成，并通过其他 AWS 服务（例如亚马逊 Athena）查询亚马逊 S3 数据。有关更多信息，请参阅 <a href="#">将 AWS Glue Data Catalog 与 Amazon S3 目标一起用于 AWS DMS</a> 。
将 DocumentDB 作为目标时的并行应用	使用 DocumentDB 作为具有新 ParallelApply* 任务设置的目标，在 CDC 复制期间 AWS DMS 现在最多支持每秒 5000 条记录。有关更多信息，请参阅 <a href="#">将 Amazon DocumentDB 作为 AWS Database Migration Service 的目标</a> 。
以客户为中心的日志记录	现在，在 3.5.0 AWS DMS 版本中，您可以更有效地检查和管理任务日志。有关查看和管理 AWS DMS 任务日志的信息，请参阅 <a href="#">查看和管理 AWS DMS 任务日志</a> 。

新功能或增强功能	描述
Kafka 目标端点的 SASL_PLAIN 机制	现在，您可以使用 SASL_PLAIN 身份验证来支持 Kafka MSK 目标端点。
在 MySQL 中复制 XA 事务	现在，您可以在 MySQL DMS 源上使用 XA 事务。在 DMS 3.5.0 之前，作为 XA 事务的一部分应用的 DML 更改无法正确复制。
Oracle 扩展数据类型	AWS DMS 现在支持在 Oracle 12.2 及更高版本中复制扩展数据类型。
Db2 LUW 环境 PureScale	AWS DMS 现在支持从 Db2 LUW PureScale 环境进行复制。仅当使用“从源更改位置开始处理更改”选项时才支持此功能。
SQL 服务器源和 READ_COMMITTED_SNAPSHOT 选项	在 READ_COMMITTED_SNAPSHOT 选项设置为的情况下使用 Microsoft SQL Server 源数据库时 TRUE，可以通过设置强制 DataRow 查找连接属性来正确复制 DML 更改。

AWS DMS 3.5.0 包括以下已解决的问题：

202 AWS DMS 3 年 3 月 17 日发布的 3.5.0 中已解决的问题

主题	解决方案
Oracle – 比较从数字转换而来的字符串的特殊大小写	修复了 Oracle 源中，当数字列中将数据类型转换为字符串时，同一列的筛选规则无法按预期工作的问题。
本地 SQL Server AG 增强功能	通过消除与 DMS 未使用的副本的不必要连接，提高了在 AlwaysOn 配置中使用 SQL Server 源的连接处理效率。
SQL Server HIERARCHYID 内部转换	修复了 SQL Server 源中的问题，在该问题中，复制到 SQL Server 目标时，HIERARCHYID 数据类型复制为 VARCHAR(250) 而不是 HIERARCHYID。
S3 目标移动任务修复	修复了使用 S3 目标的任务在移动时需要很长时间、显示为冻结或永远无法完成的问题。

主题	解决方案
Kafka 的 SASL Plain 机制	为 Kafka MSK 目标端点引入了对 SASL Plain 身份验证方法的支持。
由于 Opensearch 2.x 中有 <code>_type</code> 参数，并行加载/应用失败	修复了 Opensearch 2.x 目标由于缺少对 <code>_type</code> 参数的支持，并行加载或并行应用会失败的问题。
支持使用混合运算符的表映射筛选器	删除了只能对一个列应用一个筛选器的限制。
S3、Kinesis、Kafka 端点 – CDC 阶段中基于变更的 LOB 列迁移	修复了 Kinesis、Kafka 和 S3 目标在 CDC 期间不复制添加的 LOB 列中数据的问题。
MongoDB 驱动程序升级	MongoDB 驱动程序已升级到 v1.23.2。
Kafka 驱动程序更新	Kafka 驱动程序已从 1.5.3 升级到 1.9.2。
S3 端点设置无法正常工作	修复了 S3 目标的一个问题，在该问题中，当数据包含指定作为 S3 目标的分隔符的字符时， <code>AddTrailingPaddingCharacter</code> 端点设置不起作用。
Kinesis 目标任务会崩溃	修复了 Kinesis 目标的问题，在该问题中，当 PK 值为空且启用了详细调试时，任务会崩溃。
当 S3 目标的列名偏移一个位置时	修复了 S3 目标的问题，在该问题中，当 <code>AddColumnName</code> 设置为 <code>true</code> 且 <code>TimestampColumnName</code> 设置为 <code>""</code> 时，列名会移动一个位置。
改进了日志记录 LOB 截断警告	改进了 SQL Server 源的 LOB 截断警告日志记录，以包括用于检索 LOB 的选择语句。
添加致命错误以避免在 TDE 密码错误时 DMS 任务崩溃。	引入了有意义的错误消息，并消除了将 Oracle 作为源时，在由于 TDE 密码不正确而导致 DMS 任务失败且没有错误消息的情况下，任务崩溃的问题。

主题	解决方案
允许在 CDC 期间 PostgreSQL CTAS ( Create Table As Selected , 根据选定内容创建表 ) DDL 的迁移。	移除了在 CDC 期间 , DMS 无法复制 PostgreSQL CTAS ( Create Table As Selected , 按选定内容创建表 ) DDL 的限制。
修复 CDC 中删除表列时的 pg_logical 任务崩溃问题。	修复了带有 S3 目标的 PostgreSQL 源的一个问题 , 在该问题中 , 禁用了 LOB 支持且存在 LOB 时 , 目标上的列未对齐。
修复 MySQL 连接处理中的内存泄漏问题	修复了 MySQL 源中任务内存消耗持续增加的问题。
Oracle 源端点设置 - ConvertTimestampWithZoneToUTC	将此属性设置为 true , 以便将“TIMESTAMP WITH TIME ZONE”和“TIMESTAMP WITH LOCAL TIME ZONE”列的时间戳值转换为 UTC。默认情况下 , 此属性的值为“false” , 数据将使用源数据库时区进行复制。
Oracle 源 - DataTruncationErrorPolicy 设置为 SUSPEND_TABLE 时不起作用	修复了将作为 S3 目标的 Oracle 源的问题 , 在该问题中 , DataTruncationErrorPolicy 任务设置被设置为 SUSPEND_TABLE 时表不暂停。
构建查询子句时 , SQL Server 在长架构/表上失败	修复了 SQL Server 源的一个问题 , 在该问题中 , 当选择规则包含以逗号分隔的表列表时 , 任务会失败或无响应。
MongoDB 端点的 Secret Manager 身份验证	修复了 MongoDB 和 DocumentDB 端点中基于 Secret Manager 的身份验证不工作的问题。

主题	解决方案
当 NLS_NCHAR_CHARACTERSET 设置为 UTF8 时，DMS 会在 CDC 期间截断多字节 varchar 列的数据	修复了在 Oracle 源和 Oracle 目标中，当 NLS_NCHAR_CHARACTERSET 设置为 UTF8 时，多字节 VARCHAR 列的数据被截断的问题。
filterTransactionsOfUser 甲骨文版 ECA LogMiner	添加了额外连接属性 (ECA)filterTransactionsOfUser，允许 DMS 在使用从 Oracle 进行复制时忽略来自指定用户的交易。LogMiner
SQL Server 在备份中丢失 lsn 时设置可恢复错误	修复了 SQL Server 在缺少 LSN 时任务不会失败的问题。

## AWS Database Migration Service 3.4.7 版本说明

下表显示了 AWS Database Migration Service (AWS DMS) 版本 3.4.7 中引入的新功能和增强功能。

新功能或增强功能	描述
支持将 Babelfish 作为目标	<p>AWS DMS 现在支持 Babelfish 作为目标。使用 AWS DMS，您现在可以将实时数据从任何 AWS DMS 支持的来源迁移到 Babelfish，停机时间最短。</p> <p>有关更多信息，请参阅 <a href="#">使用 Babelfish 作为 AWS Database Migration Service 的目标</a>。</p>
支持将 IBM Db2 z/OS 数据库作为仅完全加载的源	<p>AWS DMS 现在支持 IBM Db2 z/OS 数据库作为源。使用 AWS DMS，您现在可以执行从 Db2 大型机到任何 AWS DMS 支持的目标的实时迁移。</p> <p>有关更多信息，请参阅 <a href="#">将 IBM Db2 for z/OS 数据库用作 AWS DMS 的源</a>。</p>

新功能或增强功能	描述
支持将 SQL Server 只读副本作为源	<p>AWS DMS 现在支持 SQL Server 只读副本作为源。现在 AWS DMS，您可以使用执行从 SQL Server 只读副本到任何 AWS DMS 支持的目标的实时迁移。</p> <p>有关更多信息，请参阅 <a href="#">使用微软 SQL Server 数据库作为来源 AWS DMS</a>。</p>
Support EventBridge 或 DMS 活动	<p>AWS DMS 支持使用 EventBridge 针对 DMS 事件管理事件订阅。</p> <p>有关更多信息，请参阅 <a href="#">在 AWS Database Migration Service 中使用 Amazon EventBridge 事件和通知</a>。</p>
支持 VPC 源端点和目标端点	<p>AWS DMS 现在支持亚马逊虚拟私有云 (VPC) Virtual Private Cloud 终端节点作为源和目标。AWS DMS 现在，如果在 AWS 服务的 VPC 中定义了明确定义的路由，则可以通过 AWS DMS VPC 终端节点连接到任何服务。</p> <div data-bbox="544 957 1507 1318" style="border: 1px solid #add8e6; border-radius: 15px; padding: 15px; margin: 10px 0;"> <p> <b>Note</b></p> <p>升级到 3.4.7 及更高 AWS DMS 版本要求您首先配置 AWS DMS 为使用 VPC 终端节点或使用公共路由。此要求适用于亚马逊 S3、亚马逊 Kinesis Data Streams、亚马逊 DynamoDB AWS Secrets Manager、Amazon Redshift 和亚马逊服务的源端点和目标终端节点。OpenSearch</p> </div> <p>有关更多信息，请参阅 <a href="#">将 VPC 端点配置为 AWS DMS 源端点和目标端点</a>。</p>
新 PostgreSQL 版本	<p>现在支持将 PostgreSQL 版本 14.x 作为源和目标。</p>
支持将 Aurora Serverless v2 作为目标	<p>AWS DMS 现在支持 Aurora Serverless v2 作为目标。使用后 AWS DMS，您现在可以实时迁移到 Aurora Serverless v2。</p> <p>有关支持的 AWS DMS 目标的信息，请参见 <a href="#">数据迁移的目标</a>。</p>

新功能或增强功能	描述
新 IBM Db2 for LUW 版本	<p>AWS DMS 现在支持 IBM Db2 for LUW 版本 11.5.6 和 11.5.7 作为来源。现在 AWS DMS，您可以使用最新版本的 IBM DB2 for LUW 执行实时迁移。</p> <p>有关 AWS DMS 来源的信息，请参见<a href="#">数据迁移的源</a>。</p> <p>有关支持的 AWS DMS 目标的信息，请参见<a href="#">数据迁移的目标</a>。</p>

AWS DMS 3.4.7 包括以下新的或已更改的行为以及已解决的问题：

- 现在，在使用 Amazon S3 作为源时，您可以使用表定义中的日期格式将数据字符串解析为日期对象。
- 新的表统计数据计数器现已推出：AppliedInserts、AppliedDdls、AppliedDeletes 和 AppliedUpdates。
- 现在，当 OpenSearch 用作目标时，您可以选择默认的映射类型。
- 通过 Oracle、PostgreSQL 和 SQLServer 源的新 TrimSpaceInChar 端点设置，您可以指定是否修剪 CHAR 和 NCHAR 数据类型的数据。
- 使用 S3 作为源或目标时，Amazon S3 的新 ExpectedBucketOwner 端点设置可防止探测。
- 对于 RDS SQL Server、Azure SQL Server 和自管理 SQL Server，DMS 现在可以在为迁移任务选择的所有表上自动设置 MS-CDC，不论这些表是否有主键，或者在具有唯一索引时，考虑在具有主键的自管理 SQL Server 表上为 MS-REPLICATION 启用优先级。
- 增加了对在 Oracle 同构迁移期间，对复制 Oracle 分区和子分区 DDL 操作的支持。
- 修复了使用 Oracle 作为源和目标时，复合主键导致数据验证任务崩溃的问题。
- 修复了一个问题，在使用 Redshift 作为目标时，当目标列预先创建为布尔值时，能够将不同的字符类型正确转换为布尔值。
- 修复了当使用 PostgreSQL 作为目标时，由于已知的 ODBC 问题，将 varchar 数据类型作为 varchar(255) 迁移时会导致数据截断的问题。
- 修复了使用 Oracle 作为目标时，在 BatchApplyEnabled 设置为 true 且 BatchApplyPreserveTransaction 设置为 false 时，不遵守 DELETE 操作并行提示的问题。
- 使用 S3 作为目标时，Amazon S3 的新 AddTrailingPaddingCharacter 端点设置增加了对字符串数据进行填充的功能。



- 新的 `max_statement_timeout_seconds` 任务设置延长了端点查询的默认超时。MySQL 端点元数据查询目前使用此设置。
- 修复了使用 PostgreSQL 作为目标时，CDC 任务未正确使用错误处理任务设置的问题。
- 修复了 DMS 无法正确识别 Redis Enterprise 实例的 Redis 模式的问题。
- 扩展了对 S3 目标 Parquet 格式的 `includeOpForFullLoad` 额外连接属性 (ECA) 的支持。
- 引入了一个新的 PostgreSQL 端点设置 `migrateBooleanAsBoolean`。对于从 PostgreSQL 迁移到 Redshift，当此设置被设置为 `true` 时，布尔值将作为 `varchar(1)` 进行迁移。如果将其设置为 `false`，则布尔值将作为 `varchar(15)` 进行迁移，这是默认行为。
- 修复了使用 SQL Server 源时，`datetime` 数据类型的迁移问题。此修复解决了精度为毫秒时插入 `Null` 的问题。
- 对于带有 PGLOGICAL 的 PostgreSQL 源，修复了一个迁移问题，在该问题中，在 CDC 阶段使用 `pglogical` 并从源表中删除字段时，删除字段之后的值不会迁移到目标表。
- 修复了 SQL Server 环回迁移中双向复制获得重复记录的问题。
- 为 PostgreSQL 作为源添加了一个新的 ECA `mapBooleanAsBoolean`。使用这个额外的连接属性，你可以覆盖 PostgreSQL 布尔值到 RedShift 布尔数据类型的默认数据类型映射。
- 修复了使用 SQL Server 作为源时的迁移问题，解决了 `ALTER DECIMAL/NUMERIC SCALE` 未复制到目标的情况。
- 修复了 SQL Server 2005 的连接问题。
- 自 2022 年 10 月 17 日起，DMS 3.4.7 现在支持将第 6 代 Amazon EC2 实例类用作复制实例。
- 从 2022 年 11 月 25 日起，通过 DMS 3.4.7，您可以使用 DMS 架构转换来转换数据库架构和代码对象，并使用 DMS Fleet Advisor 在网络环境中发现适合迁移的数据库。
- 自 2022 年 11 月 25 日起，DMS Studio 停用。
- 自 2023 年 1 月 31 日起，DMS 架构转换支持将 Aurora MySQL 和 Aurora PostgreSQL 作为目标数据提供程序。
- 从 2023 年 3 月 6 日起，您可以使用 DMS Fleet Advisor 为源数据库生成合适目标大小的建议。
- 自 2023 年 3 月 6 日 AWS DMS 起，支持允许向亚马逊发布指标数据点的 AWS 托管政策 CloudWatch。

## 2023 年 5 月 5 日发布的 DMS 3.4.7 维护版本中解决的问题

主题	解决方案
PostgreSQL 源任务失败	修复了 PostgreSQL 源中，当单个事件中超过允许的 DDL 操作最大数时，任务会失败的问题。
PostgreSQL 源数据验证误报	修复了 PostgreSQL 为源和 Oracle 为目标时，时间戳字段的转换不正确会导致误报数据验证错误的问题。
MySQL 源错误处理	修复了 MySQL 源中下一个 BIN 日志不可用时 DMS 任务不失败的问题。
MySQL 源 ROTATE_EVENT 日志记录	改进了与 ROTATE_EVENT 相关的 MySQL 源的日志记录，包括所读取的 BIN 日志名称。
数据验证超时问题	修复了数据验证功能的一个问题，在该问题中，与数据验证相关的查询不遵守 <code>executeTimeout</code> 端点设置。
PostgreSQL 目标并行完全加载问题	修复了 PostgreSQL 目标中由于“连接中断”错误而导致分段（并行）完全加载失败的问题。
DMS 任务移动问题	修复了 S3 目标中 DMS 任务移动操作花费很长时间或从未完成的问题。
PostgreSQL 源中重复记录问题	修复了 PostgreSQL 源的一个问题，在该问题中，任务停止并恢复后，DMS 任务会引发与目标上重复项相关的错误。
Oracle 目标数据验证误报	修复了 Oracle 目标的一个问题，在该问题中，由于时间戳字段的复制时区不正确，数据验证会出现误报的错误。

## 2023 年 2 月 22 日发布的 DMS 3.4.7 维护版本中解决的问题

主题	解决方案
将 SQL Server AG 副本作为源	在侦听器 TCP 端口与副本 TCP 端口不同的AlwaysOn配置中添加了对 SQL Server 源的支持。
将 Amazon Redshift 作为目标时的数据丢失	修复了 Redshift 目标中，在极少数情况下 Redshift 意外重启可能会导致目标上的数据丢失的问题。
SQL Server 源保护支持	修复了 SQL Server 源的一个问题，在该问题中，当指定了端点设置 "SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION" 时，DMS 任务可能会失败，并显示错误，说明无法读取事务日志备份。
Oracle 作为源时数据验证任务失败	修复了 Oracle 源中，由于主键值识别不正确，DMS 任务在数据验证时可能失败的问题。
图像数据发布之前的 Kinesis	修复了流式传输目标 ( Kinesis、Kafka ) 中 "EnableBeforeImage" 任务设置仅对字符数据类型起作用的问题。
Time Travel 日志文件	修复了当源处于空闲状态时，DMS 会创建零字节 Time Travel 日志文件的问题。

## 2022 年 12 月 16 日发布的 DMS 3.4.7 维护版本中解决的问题

主题	解决方案
BatchApply已启用	修复了设置 BatchApplyEnabled 为 True 时日志记录过多的问题。
新的 MongoDB 端点设置-超时 FullLoad NoCursor	MongoDB 端点 FullLoadNoCursorTimeout 设置为满载 NoCursorTimeout 光标指定。NoCursorTimeout 是一个 MongoDB 连接设置，用于防止服务器在空闲时关闭光标。

主题	解决方案
MongoDB – 用于单列分段的筛选功能	新的筛选功能提高了使用单列进行分段来迁移 MongoDB 数据库的性能。
MongoDB 到 Redshift	修复了从 MongoDB 迁移到 Redshift 时，如果 MongoDB 集合具有二进制数据类型，DMS 无法在 Redshift 上创建目标表的问题。
新的 Mon SocketTimeout goDB MS 连接属性	新的 M SocketTimeout ongoDB MS 额外连接属性以毫秒为单位配置 MongoDB 客户端的连接超时。如果该值小于或等于零，则使用 MongoDB 客户端默认值。
修复了导致 Amazon Kinesis 任务崩溃的问题	修复了当迁移到 Amazon Kinesis Data Streams 目标时，在表中不存在主键的情况下处理 null 值的问题。
支持 Oracle NULL PK/UK 数据验证	删除了不支持 NULL PK/UK 值的数据验证的限制。
Oracle 到 Amazon S3	修复了从 Oracle 迁移到 Amazon S3 时，将一些记录错误地迁移为 NULL 的问题。
Oracle Standby	使用 Oracle Standby 作为源时，增加了 DMS 处理未完成事务的功能。
Oracle 迁移到 Oracle，使用 SDO_GEOMETRY 空间数据类型	修复了从 Oracle 迁移到 Oracle 时，表的 DDL 中存在 SDO_GEOMETRY 列时任务会失败的问题。
Oracle 作为源	修复了使用 Oracle 作为源时，DMS 偶尔会跳过 Oracle 重做日志序列号的问题。
Oracle 作为源 – 缺少存档/联机重做日志	修复了使用 Oracle 作为源时，当存档日志丢失时 DMS 任务会失败的问题。

主题	解决方案
已修复 – DMS 偶尔会跳过 Oracle Standby 重做日志	修复了使用 Oracle 作为源时，DMS 偶尔会跳过 Oracle 重做日志序列号的问题。
已修复 – Oracle 复制到 Oracle 在 CDC 期间无法复制空间数据类型	修复了从 Oracle 复制到 Oracle 时，在 CDC 期间无法复制空间数据类型的问题。
Oracle 作为目标	使用 Oracle 作为目标时，修复了目标应用失败并出现 ORA-01747 错误的问题。
Amazon S3 – 修复了重新加载表数据丢失的问题	修复了使用 Amazon S3 作为目标时，表重新加载操作未生成 CDC 文件的问题。
已修复 – 将主服务器作为源时 SQL Server Always On 的上下文初始化	使用 SQL Server Always On 作为源时，修复了在源为主源且设置 AlwaysOnSharedSyncedBackupsEnabled 为 true 时无法初始化可用性组 (AG) 的问题。
更新了 SQL 服务器端点设置	当源端点是 SQL Server Always On 可用性组并且是辅助副本时，修复了设置 AlwaysOnSharedSyncedBackupsEnabled 为 True 时复制任务失败的问题。
将 PostgreSQL 作为源	修复了 CDC 无法迁移 PostgreSQL 源上的删除/更新操作的问题，该操作是在 3.4.7 中引入的，支持布尔值。 mapBooleanAs

## AWS Database Migration Service 3.4.6 版本说明

下表显示了 AWS Database Migration Service (AWS DMS) 版本 3.4.6 中引入的新功能和增强功能。

新功能或增强功能	描述
AWS DMS 时光旅行	<p>AWS DMS 推出了 <a href="#">Time Travel</a>，该功能为客户提供了灵活的日志记录功能，并增强了他们的故障排除体验。借助 Time Travel，您可以使用 Amazon S3 存储和加密 AWS DMS 日志，并在特定时间范围内查看、下载和混淆日志。</p>
支持将 Microsoft Azure SQL 托管实例作为源	<p>AWS DMS 现在支持微软 Azure SQL 托管实例作为源。使用后 AWS DMS，你现在可以执行从 Microsoft Azure SQL 托管实例到任何 AWS DMS 支持的目标的实时迁移。</p> <p>有关 AWS DMS 来源的信息，请参见<a href="#">数据迁移的源</a>。</p> <p>有关支持的 AWS DMS 目标的信息，请参见<a href="#">数据迁移的目标</a>。</p>
支持将 Google Cloud SQL for MySQL 作为源	<p>AWS DMS 现在支持将适用于 MySQL 的谷歌云 SQL 作为源。通过使用 AWS DMS，您现在可以执行从适用于 MySQL 的 Google Cloud SQL 到任何 AWS DMS 支持的目标的实时迁移。</p> <p>有关 AWS DMS 来源的信息，请参见<a href="#">数据迁移的源</a>。</p> <p>有关支持的 AWS DMS 目标的信息，请参见<a href="#">数据迁移的目标</a>。</p>
支持将分区数据并行加载到 S3	<p>AWS DMS 现在支持将分区数据并行加载到 Amazon S3，从而缩短了将分区数据从支持的数据库引擎源数据迁移到 Amazon S3 的加载时间。此功能为数据库源中表的每个分区创建 Amazon S3 子文件夹，从而允许 AWS DMS 运行并行进程来填充每个子文件夹。</p>
支持单个任务中的多个 Apache Kafka 目标主题	<p>AWS DMS 现在支持使用单个任务的 Apache Kafka 多主题目标。使用 AWS DMS，您现在可以使用同一个任务，将多个架构从单个数据库复制到不同的 Apache Kafka 目标主题。这样，在需要将来自同一源数据库的多个表迁移到不同的 Kafka 目标主题时，就不需要创建多个单独的任务。</p>

AWS DMS 3.4.6 中解决的问题包括以下内容：

- 修复了使用 Amazon S3 作为目标且使用 CSV 格式时，如果主键列不是第一列，UPDATE 语句中的列会填充到错误列的问题。

- 修复了使用 PostgreSQL 作为源时，在受限 LOB 模式下使用带有 BYTEA 列 NULL 值的 pglogical 插件时，AWS DMS 任务可能会崩溃的问题。
- 修复了在使用 PostgreSQL 作为源时删除大量源表时 AWS DMS 任务可能会崩溃的问题。
- 通过引入新的 Amazon S3 设置 DatePartitionTimezone 来允许在非 UTC 日期上进行分区，从而改进了 Amazon S3 基于日期的文件夹分区。
- 使用 Redshift 作为目标时，支持从源到 TIMESTAMPTZ 的数据类型 TIMESTAMP WITH TIME ZONE 之间的映射
- 提高了在使用 MongoDB 或 Amazon DocumentDB 作为源时，在没有通配符选择规则的任务中 CDC 的性能。
- 修复了在使用 Db2 LUW 作为源时，AWS DMS 任务无法捕获使用下划线通配符且长度小于 8 的架构名称的问题。
- 修复了使用 OpenSearch 服务作为目标时，AWS DMS 实例在大数据量下内存不足的问题。
- 通过支持仅完全加载验证任务，提高了数据验证的性能。
- 修复了使用 Sybase 作为源时，强制故障切换后 AWS DMS 任务无法恢复的问题。
- 修复了警告 AWS DMS 发送 Invalid BC timestamp was encountered in column 错误的问题。

在 DMS 3.4.6 维护版本中解决了以下问题：

- 修复了使用 Oracle 作为源和目标时，启用了批量应用模式时任务崩溃的问题。
- 修复了一个问题，使得完全加载任务能够正确地使用以 PostgreSQL 为源的 ExecuteTimeout 端点设置。
- 修复了在使用 PostgreSQL 作为源时，在将任务设置为受限 LOB 模式时，迁移数组数据类型列出现的问题。
- 修复了在使用 PostgreSQL 作为源时，迁移带有时区的时间戳在 1970-01-01 之前时出现的问题。
- 修复了使用 SQL Server 作为源和目标时，DMS 在复制过程中将 null 字符串视为空字符串的问题。
- 修复了使用 MySQL 源/目标时遵守会话读取和写入超时端点设置的问题。
- 修复了使用 Amazon S3 作为源时，DMS CDC 任务下载完全加载相关文件时的问题。
- 修复了使用 Amazon S3 作为目标时，将 CdcInsertsAndUpdates 和 PreserveTransactions 均设置为 true 时日志崩溃的问题。
- 修复了在使用 ParallelApply \* 功能时任务崩溃，但在使用 Amazon Kinesis Data Streams 作为源时，某些表没有默认主键的问题。
- 修复了在使用 Amazon Kinesis Data Streams 作为来源 StreamArn 时未因错误而出现错误的问题。

- 修复了使用主键值作为空字符串会导致任务在 OpenSearch 用作目标时崩溃的问题。
- 修复了数据验证使用过多磁盘空间的问题。

## 2022 年 12 月 13 日发布的 DMS 3.4.6 维护版本中解决的问题

主题	解决方案
SAP ASE odbc 驱动程序	修复了 SAP ASE 作为源时的一个问题，以便 ODBC 驱动程序可以支持字符集。
用于 lob 查找的 Sqlserver 日期时间主键错误	修复了 SQL Server 作为源，在主键具有日期时间数据类型且精度为毫秒时，LOB 查找无法正常工作的问题。
SQL Server 到 Redshift –“datetimeoffset”映射到“timestamptz”	对于从 SQL Server 到 Redshift 的迁移，改进了映射，使 SQL Server 的“datetimeoffset”格式映射到 Redshift 的“timestamptz”格式。
数据验证- SkipLobColumns 是真的	修复了当 SkipLobColumns 为 True、源上有 LOB、主键位于最后一列且通过验证检测到数据差异时 DMS 任务崩溃的问题。
以 MySQL 为源时的数据验证	修复了 MySQL 作为数据源且启用了数据验证时，使用具有 null 值的复合唯一键的表会发生 DMS 任务崩溃的问题。
MySQL 作为源	修复了将 MySQL 作为源时，在更改列以增加精度时，表会因溢出错误而暂停的问题。
MySQL ODBC 驱动程序升级到 8.0.23	修复了 MySQL 作为源时，排序规则“utf8mb4_0900_bin”与 DMS 使用的 mysql 驱动程序不兼容的问题。
MySQL – 支持分区表的 DDL 更改	引入了新的 MySQL 端点设置，skipTableSuspensionForPartitionDdl 允许用户在 CDC 期间跳过暂停表以更改分区 DDL，这样 DMS 现在可以支持对已分区的 MySQL 表进行 DDL 更改。



主题	解决方案
MongoDB 到 Redshift 的迁移	修复了从 MongoDB 迁移到 Redshift 时，如果 MongoDB 集合具有二进制数据类型，DMS 无法在 Redshift 上创建目标表的问题。
Redshift 目标 – 批量应用中的 Time Travel Segfault	修复了 Redshift 作为目标的问题，即 BatchApplyEnabled 设置为 true 时，DMS 任务会崩溃。
Redshift 作为目标	修复了 Redshift 作为目标时的问题，在该问题中，当并行加载设置为 type=partitions-auto 时，并行分段会将批量 CSV 文件写入同一个表目录并相互干扰。
Redshift 作为目标	修复了将 Redshift 作为目标时的问题，在该问题中，CDC 期间目标列的类型为布尔值，但源的类型为可变字符。
Redshift 作为目标	改进了任务日志，以识别无法复制到 Redshift 目标的 DDL 更改。
使用 PostgreSQL 时的数据验证	修复了 PostgreSQL 的验证中，当存在布尔数据类型时验证会失败的问题。
将 PostgreSQL 作为源	修复了 PostgreSQL 作为源的问题，因此满载时会使用额外连接属性 ExecuteTimeout 中的字段。
将 PostgreSQL 作为源	修复了 PostgreSQL 作为源时的问题，在该问题中，如果任务读取的 LSN 大于请求的任务，任务将失败，在超过 60 分钟之后恢复 LSN，以表明正在使用的复制时隙有问题。
PostgreSQL 作为源 – 1970-01-01 之前的 timestampz	修复了在 PostgreSQL 作为源时，1970-01-01 之前的 timestampz 在 CDC 期间无法正确迁移的问题。
将 PostgreSQL 作为源	修复了在 PostgreSQL 作为源时，CDC 期间 DMS 会截断可变字符数据类型值的问题。

主题	解决方案
PostgreSQL 作为源 – 恢复已停止的任务	修复了 PostgreSQL 作为源时，恢复之前停止的任务重放会在 CDC 期间错过一个或多个事务的问题。
Amazon S3 作为目标	修复了 S3 作为目标的问题，即当为 true 且 AddColumnName TimestampColumnName 为 "" 时，生成的 CSV 文件标题会偏离一列。
Amazon S3 作为源 – 任务的完全加载阶段的内存使用行为	修复了 S3 作为源时，处于完全加载阶段的 DMS 任务仅在整个表加载到目标数据库后才释放所用内存的问题。
Amazon S3 作为目标 – 表重新加载操作	修复了使用 S3 作为目标时，表重新加载操作未生成 CDC 文件的问题。

## AWS Database Migration Service 3.4.5 版本说明

下表显示了 AWS Database Migration Service (AWS DMS) 版本 3.4.5 中引入的新功能和增强功能。

新功能或增强功能	描述
支持将 Redis 作为目标	AWS DMS 现在支持 Redis 作为目标。现在 AWS DMS，您可以使用，将实时数据从任何 AWS DMS 支持的来源迁移到 Redis 数据存储，停机时间最短。有关 AWS DMS 目标的信息，请参见 <a href="#">数据迁移的目标</a> 。
支持将 MongoDB 4.2 和 4.4 作为源	AWS DMS 现在支持 MongoDB 4.2 和 4.4 作为源代码。使用 AWS DMS，您现在可以将数据从 MongoDB 4.2 和 4.4 集群迁移到 AWS DMS 任何支持的目标，包括亚马逊 DocumentDB (兼容 MongoDB)，停机时间最短。有关 AWS DMS 来源的信息，请参见 <a href="#">数据迁移的源</a> 。
使用 MongoDB 作为源时支持多个数据库	AWS DMS 现在支持使用 MongoDB 作为源在一个任务中迁移多个数据库。现在 AWS DMS，您可以使用 MongoDB 集群的多个数据库进行分组，并使用一个数据库迁移任务进行迁移。您可以迁移到

新功能或增强功能	描述
	任何 AWS DMS 支持的目标，包括 Amazon DocumentDB (兼容 MongoDB)，停机时间最短。
支持使用 MongoDB 或 Amazon DocumentDB (与 MongoDB 兼容) 作为源时的自动分段	AWS DMS 现在支持使用 MongoDB 或 Amazon DocumentDB 作为来源进行自动分段。使用 AWS DMS，您可以配置数据库迁移任务，以自动分段 MongoDB 或 DocumentDB 集群的集合。然后，您可以将区段并行迁移到任何 AWS DMS 支持的目标，包括 Amazon DocumentDB，同时最大限度地减少停机时间。
Amazon Redshift 完全加载性能改进	AWS DMS 现在支持在满负荷期间使用 Amazon Redshift 作为目标时使用并行线程。通过利用多线程满载任务设置，您可以提高从任何 AWS DMS 支持的来源到 Amazon Redshift 的初始迁移性能。有关 AWS DMS 目标的信息，请参见 <a href="#">数据迁移的目标</a> 。

AWS DMS 3.4.5 中解决的问题包括以下内容：

- 修复了在使用 PostgreSQL 作为源且具有高事务并发度时，恢复操作后数据可能会丢失或重复的问题。
- 修复了在使用 PostgreSQL 作为源且启用了 pglogical 插件时，数据库迁移任务失败并显示错误找不到关系 ID... 的问题。
- 修复了使用 PostgreSQL 作为源并使用 Oracle 作为目标时，VARCHAR 列未正确复制的问题。
- 修复了使用 PostgreSQL 作为源，当主键不是表定义中的第一列时无法正确捕获删除操作的问题。
- 修复了在使用 MySQL 作为源时，在特殊元数据设置下数据库迁移任务会错过 LOB 更新的问题。
- 修复了使用 MySQL 版本 8 作为源时，在完整 LOB 模式下 TIMESTAMP 列被视为 DATETIME 问题。
- 修复了使用 MySQL 5.6.4 及更高版本作为源时，解析 NULL DATETIME 记录时数据库迁移任务失败的问题。
- 修复了使用 Amazon Redshift 作为目标且使用并行应用时，遇到线程正在退出错误后数据库迁移任务卡住的问题。
- 修复了在批量应用 CDC 期间，数据库迁移任务与 Amazon Redshift 目标端点断开连接时可能会丢失数据的问题。
- 使用 Amazon Redshift 作为目标时，通过调用 ACCEPTINVCHARS 来提高完全加载的性能。
- 修复了使用 Amazon Redshift 作为目标从 one-by-one 模式恢复到并行应用模式时复制重复记录的问题。

- 修复了在使用 Amazon S3 作为目标时，数据库迁移任务不会通过 `cannedAclForObjects=bucket_owner_full_control` 将 Amazon S3 对象所有权切换为存储桶所有者的问题。
- 使用 Oracle 作为来源 `additionalArchivedLogDestId` 时，AWS DMS 通过使用 ECA 支持多个存档目标而得到改进。
- 修复了完整 LOB 模式下更新 LOB 列时，数据库迁移任务失败并出现错误 `OCI_INVALID_HANDLE` 的问题。
- 修复了使用 Oracle 作为源时，CDC 期间无法正确迁移 `NVARCHAR2` 列的问题。
- AWS DMS 通过 `SafeguardPolicy` 在使用适用于 SQL Server 的 RDS 作为源时启用而得到改进。
- 修复了使用非 RDS SQL Server 源时，数据库迁移任务报告有关 `rdsadmin` 的错误的问题。
- 修复了使用 SQL Server 作为源时，在分区设置中将 `UUID` 作为主键时数据验证失败的问题。
- 修复了使用 Db2 LUW 作为源时，如果在数据库日志中找不到所需的 `LSN`，则完全加载 + CDC 任务可能会失败的问题。
- AWS DMS 通过在使用 MongoDB 作为源时支持自定义 CDC 时间戳进行了改进。
- 修复了使用 MongoDB 作为源，MongoDB 驱动程序出现错误 `endSessions` 而停止时，数据库迁移任务会卡住的问题。
- 修复了使用 DynamoDB 作为目标时 AWS DMS 无法更新非主要字段的问题
- 修复了数据验证报告在 `CLOB` 和 `NLOB` 列上的误报不匹配的问题。
- 修复了使用 Oracle 作为源时，只有空格的记录的数据验证失败的问题。
- 修复了截断分区表时数据库迁移任务崩溃的问题。
- 修复了创建 `awsdms_apply_exceptions` 控制表时数据库迁移任务失败的问题。
- 扩展了使用 MySQL 版本 8 时，对 `caching_sha2_password` 身份验证插件的支持。

## AWS Database Migration Service 3.4.4 发行说明

下表显示了 AWS DMS 版本 3.4.4 中引入的新功能和增强功能。

新功能或增强功能	描述
使用 Kafka 作为目标时，支持 TLS 加密以及 TLS 或 SASL 身份验证，	AWS DMS 现在支持使用 Amazon MSK 和本地 Kafka 集群作为目标的 TLS 加密以及 TLS 或 SASL 身份验证。有关对 Kafka 端点使用加密和身份验证的更多信息，请参阅 <a href="#">使用传输层安全性协议 ( TLS ) 连接到 Kafka</a> 。

AWS DMS 3.4.4 中解决的问题包括以下内容：

- 改进了使用 Oracle 端点时任务失败的 AWS DMS 日志记录。
- 在 Oracle Data Guard 故障转移后，当 Oracle 源端点切换角色时，改进的 AWS DMS 任务执行会继续处理。
- 在使用 Oracle 端点时，改进的错误处理将 ORA-12561 视为可恢复的错误。
- 修复了使用 Oracle 作为源时，EMPTY\_BLOB() 和 EMPTY\_CLOB() 列作为 null 迁移的问题。
- 修复了使用 SQL Server 作为源时，添加列 DDL 更改后，AWS DMS 任务无法更新记录的问题。
- 通过支持 TIMESTAMP WITH TIME ZONE 数据类型改进了 PostgreSQL 作为源时的迁移功能。
- 修复了使用 PostgreSQL 作为目标时，afterConnectScript 设置在完全加载期间不起作用的问题。
- 引入了一个新 mapUnboundedNumericAsString 设置，以便在使用 PostgreSQL 端点时，更好地处理没有精度和小数位数的 NUMERIC 日期类型。
- 修复了在使用 PostgreSQL 作为源时，停止并恢复 AWS DMS 任务后任务失败并显示“0 行受到影响”的问题。
- 修复了使用 PostgreSQL 作为源时 AWS DMS 无法迁移带 BC 后缀 TIMESTAMP 的数据类型的问题。
- 修复了使用 PostgreSQL 作为源时 AWS DMS 无法迁移 TIMESTAMP 值“±infinity”的问题。
- 修复了使用 S3 作为源时，csvNullValue 设置被设置为其他值的情况下将空字符串作为 NULL 进行处理的问题。
- 改进了使用 S3 作为目标时，完全加载和 CDC 中的 timestampColumnName 额外连接属性在 CDC 期间可进行排序。
- 改进了使用 S3 作为源时，对十六进制格式的二进制数据类型（例如 BYTE、BINARY 和 BLOB）的处理。
- 修复了使用 S3 作为目标时，已删除的记录会使用特殊字符迁移的问题。
- 修复了使用 Amazon DocumentDB（与 MongoDB 兼容）作为目标时处理空键值的问题。
- 修复了使用 MongoDB NumberDecimal 或 Amazon DocumentDB（兼容 MongoDB）作为源时 AWS DMS 无法复制或 Decimal128 列的问题。
- 修复了在使用 MongoDB 或 Amazon DocumentDB（与 MongoDB 兼容）作为源时，其上出现失效转移时允许 CDC 任务重试的问题。
- 添加了一个选项，用于在使用 Kinesis、Kafka 或作为目标时删除 RAW 数据类型值的十六进制“0x”前缀。OpenSearch
- 修复了使用 Db2 LUW 作为源时，对固定长度的字符列进行验证失败的问题。

- 修复了只有源数据类型或目标数据类型为 FLOAT 或 DOUBLE 时验证失败的问题。
- 修复了使用 Oracle 作为源时，NULL 字符的验证失败的问题。
- 修复了使用 Oracle 作为源时，XML 列的验证失败的问题。
- 修复了使用 MySQL 作为源的复合键中存在可为空的列时 AWS DMS 任务崩溃的问题。
- 修复了 AWS DMS 无法验证 SQL Server 源端点UNIQUEIDENTIFIER中的两列和 PostgreSQL 目标端点的 UUID 列的问题。
- 修复了在源表定义修改之后，CDC 任务不使用更新的源表定义的问题。
- 改进了 AWS DMS 故障转移功能，可将由无效的用户名或密码导致的任务失败视为可恢复的错误。
- 修复了使用适用于 SQL Server 的 RDS 作为源时，由于缺少 LSN 而导致 AWS DMS 任务失败的问题。

## AWS Database Migration Service 3.4.3 版本说明

下表显示了 AWS DMS 版本 3.4.3 中引入的新功能和增强功能。

新功能或增强功能	描述
新 Amazon DocumentDB 版本	现在支持将 Amazon DocumentDB 版本 4.0 作为源。
新 MariaDB 版本	现在支持将 MariaDB 版本 10.4 作为源和目标。
Support 对 AWS Secrets Manager 集成	您可以将支持的端点的数据库连接详细信息（用户凭证）安全地存储在 AWS Secrets Manager 中。然后，您可以在创建或修改终端节点 AWS DMS 时向其提交相应的密钥而不是纯文本凭证。AWS DMS 然后使用密钥连接到端点数据库。有关为 AWS DMS 终端节点创建密钥的更多信息，请参阅 <a href="#">使用密钥访问 AWS Database Migration Service 端点</a> 。
C5 和 R5 复制实例的更大选项	现在，您可以创建更大的复制实例大小，如下所示：C5 最大大小为 96 个 vCPU 和 192 GiB 内存，R5 最大大小为 96 个 vCPU 和 768 GiB 内存。
Amazon Redshift 性能改进	AWS DMS 现在在使用 Redshift 作为目标时支持并行应用，以提高正在进行的复制的性能。有关更多信息，请参阅 <a href="#">Amazon Redshift 的多线程任务设置</a> 。


AWS DMS 3.4.3 中解决的问题包括以下内容：

- 修复了使用 Db2 LUW 作为源时，延迟事件的提交时间戳变为“1970-01-01 00:00:00”的问题。
- 修复了在完整 LOB 模式下使用 SQL Server 作为源时，使用 NVARCHAR 列作为主键的 AWS DMS 任务失败的问题。
- 修复了使用 SQL Server 作为源时，在缓存更改阶段缺少记录的问题。
- 修复了使用适用于 SQL Server 的 RDS 作为源时，AWS DMS 任务恢复后会跳过记录的问题。
- 修复了 ASSERTIO AWS DMS N 日志组件为 SQL Server 生成大型日志的问题。
- 修复了使用 MySQL 作为源时，由于列解析溢出而在 CDC 阶段数据验证失败的问题。
- 修复了在使用 PostgreSQL 作为目标时，由于数据验证期间出现分段错误而导致 AWS DMS 任务崩溃的问题。
- 修复了使用 PostgreSQL 作为源和目标时，CDC 期间 DOUBLE 数据类型的数据验证失败的问题。
- 修复了使用 PostgreSQL 作为源并使用 Redshift 作为目标时，复制命令插入的记录未正确复制的问题。
- 修复了使用 PostgreSQL 作为源时，缓存更改阶段的数据丢失问题。
- 修复了使用 PostgreSQL 作为源时可能导致数据丢失或记录重复的问题。
- 修复了使用 PostgreSQL 作为源时，无法使用 pglogical 迁移混合大小写的架构的问题。
- 修复了使用 Oracle 作为源时，最后一条失败消息不包含 ORA 错误的问题。
- 修复了使用 Oracle 作为目标时 AWS DMS 任务无法构建 UPDATE 语句的问题。
- 修复了使用 Oracle 12.2 作为带有 ASM 和可插拔数据库配置的源时，AWS DMS 任务无法复制数据的问题。
- 使用 S3 作为源时，通过保留限额以遵守 RFC 4180，改进了记录解析。
- 改进了 timestampColumnName 的处理，使得完全加载中的列可以使用来自 CDC 的该值排序。
- 通过引入新的端点设置 MessageMaxBytes，修复了 LOB 元素大于 1MB 时 AWS DMS 任务失败的问题。
- 修复了使用 Redshift 作为目标时 AWS DMS 任务因分段错误而崩溃的问题。
- 改进了 Redshift 测试连接的错误日志记录。
- 修复了在满载期间 AWS DMS 未将所有文档从 MongoDB 传输到 DocumentDB 的问题。
- 修复了当表映射规则中没有表时，AWS DMS 任务会报告致命错误的问题。
- 修复了使用 MySQL 作为源时，在重新启动 AWS DMS 任务之前创建的架构和表无法复制到目标的问题。

- 修复了使用 MySQL 作为源时，通配符转义 [ ] 无法转义排除规则中的通配符“\_”的问题。
- 修复了使用 MySQL 作为源时，数据类型为 UNSIGNED BIGINT 的列无法正确复制的问题。

## AWS Database Migration Service 3.4.2 版本说明

下表显示了 AWS DMS 版本 3.4.2 中引入的新功能和增强功能。

新功能或增强功能	描述
<p>支持以私密方式将您的亚马逊虚拟私有云 ( Amazon VPC ) 连接到 AWS 数据库迁移服务 ( DMS ) ，而无需互联网网关、NAT 设备、VPN 连接或 AWS Direct Connect 连接。</p>	<p>现在，您可以通过自己创建的 VPC 接口终端节点连接和访问 AWS DMS 您的 Amazon VPC。此接口终端节点允许您将 AWS DMS 复制实例的所有网络活动隔离在 Amazon 网络基础设施中。通过在使用 AWS CLI 或 SDK 的所有 API 调用中引 AWS DMS 用此接口终端节点，可以确保所有 AWS DMS 活动对公共 Internet 不可见。有关更多信息，请参阅 <a href="#">AWS Database Migration Service 中的基础设施安全性</a>。</p> <div data-bbox="545 974 1507 1146" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b> 此功能适用于所有支持的 AWS DMS 引擎版本。</p> </div>
<p>使用 Amazon S3 作为目标时的基于 CDC 日期的文件夹分区</p>	<p>AWS DMS 在使用 S3 作为目标复制数据时，现在支持基于日期的文件夹分区。有关更多信息，请参阅 <a href="#">使用基于日期的文件夹分区</a>。</p>

AWS DMS 3.4.2 中解决的问题包括以下内容：

- 使用 Redshift 作为目标时，执行迁移时添加了一个 STATUPDATE 选项。
- 通过引入新设置改进了验证任务。ValidQueryCdcDelaySecond 会延迟源端点和目标端点上的第一个验证查询，以帮助在迁移延迟较高时减少资源争用。
- 修复 AWS DMS 了启动验证任务需要很长时间的问题。
- 修复了使用 S3 作为目标，在启动或停止复制任务时生成空记录的问题。
- 修复了任务在完全加载完成后卡住的问题。
- 修复了使用 S3 作为源，在源表出现数据错误时任务卡住的问题。



- 修复了禁用了源端点的用户账户时，任务在启动时卡住的问题。
- 修复了使用 PostgreSQL 作为源，在 REPLICAS IDENTITY FULL 时任务崩溃的问题。
- 修复了使用 PostgreSQL 作为源且带有 pglogical 插件时，任务会错过事务的问题。
- 修复了使用 Redshift 作为目标时 AWS DMS 未删除压缩源文件的问题。
- 修复了使用 MySQL 作为源和目标且数据类型为 BIGINT UNSIGNED 时，验证任务会出现误报的问题。
- 修复了使用 SQL Server 作为源且主键列为 CHAR 类型时，验证任务出现误报的问题。
- 修复了使用 start-replication S3 作为目标启动复制任务时 AWS DMS 无法清除目标对象的问题。
- 修复了使用 Db2 作为源时数据验证的几个问题。
- 修复了使用 SQL Server 作为源且使用 VARCHAR 列作为主键时，验证任务卡住的问题。
- 增加了使用 PostgreSQL 作为源时，对数据类型 TIMESTAMP WITH TIMEZONE 的支持。

## AWS Database Migration Service 3.4.1 测试版发行说明

下表显示了 AWS DMS 版本 3.4.1 中引入的新功能和增强功能。

新功能或增强功能	描述
新 MongoDB 版本	现在支持将 MongoDB 版本 4.0 作为源。
SQL Server 的 TLS 1.2 支持	AWS DMS 现在支持 SQL 服务器端点的 TLS 1.2。


AWS DMS 3.4.1 测试版中解决的问题包括以下内容：

- 改进了 Oracle 19c TDE 支持。
- 改进了使用 Redshift 作为目标时，对 utf8mb4 字符集和身份数据类型的支持。
- 改进了使用 MySQL 作为源且没有二进制日志时的复制任务失败处理。
- 改进了对各种数据类型和字符集的数据验证支持。
- 使用 Kinesis 和 Kafka 作为目标时，通过新的端点设置 IncludeNullAndEmpty 改进了 null 值处理。
- 改进了使用 Kafka 作为目标时的错误日志记录和处理。

- 改进了使用 SQL Server 作为源时的 DST 时间偏移。
- 修复了将 Oracle 作为目标时，复制任务尝试创建现有表的问题。
- 修复了使用 Oracle 作为源时，数据库连接终止后复制任务会卡住的问题。
- 修复了使用 SQL Server 作为源并使用 AlwaysON 设置时，复制任务无法检测并重新连接到新主服务器的问题。
- 修复了在 S3 作为目标时，特定条件下复制任务不会为 "OP" 列添加 "D" 的问题。

## AWS Database Migration Service 3.4.0 测试版发行说明

下表显示了 AWS DMS 版本 3.4.0 中引入的新功能和增强功能。

新功能或增强功能	描述
新 MySQL 版本	AWS DMS 现在支持 MySQL 8.0 版本作为源，但压缩事务负载时除外。
MySQL 支持 TLS 1.2	AWS DMS 现在支持 MySQL 端点的 TLS 1.2。
新 MariaDB 版本	AWS DMS 现在支持 MariaDB 版本 10.3.13 作为源代码。
SysAdmin 无法访问自行管理的 Microsoft SQL Server 源代码	AWS DMS 现在支持非 SysAdmin 用户访问本地和 EC2 托管的 SQL Server 源端点。  <div data-bbox="544 1228 1510 1449" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>此功能目前为测试版。如果您想试一试，请联系 AWS 支持人员以获取更多信息。</p> </div>
使用 CREATE TABLE AS 创建的 CDC 任务和 Oracle 源表	AWS DMS 现在支持针对使用该语句创建的 Oracle 源表运行满载任务以及仅限 CDC 和 CDC 的任务。CREATE TABLE AS

AWS DMS 3.4.0 中解决的问题包括以下内容：

- 改进了迁移前任务评测。有关更多信息，请参阅 [为任务启用和使用迁移前评估](#)。
- 改进了对浮点数、实数和双精度数据类型的数据验证。

- 通过更好地处理以下错误，改进了将 Amazon Redshift 作为目标时的性能：“指定的密钥不存在”。
- 支持以亚马逊 OpenSearch 服务 ( OpenSearch 服务 ) 为目标的多线程 CDC 加载任务设置，包括ParallelApplyThreadsParallelApplyQueuesPerThread、和。ParallelApplyBufferSize
- 通过支持使用复合主键来改进 OpenSearch 服务即目标。
- 修复了使用 PostgreSQL 作为源且密码中包含特殊字符时，测试连接失败的问题。
- 修复了使用 SQL Server 作为源时某些 VARCHAR 列被截断的问题。
- 修复了使用 Amazon RDS SQL Server 作为来源时无法关闭 AWS DMS 未结交易的问题。如果轮询间隔参数设置不正确，这可能会导致数据丢失。有关如何设置建议的轮询间隔值的更多信息，请参阅[使用微软 SQL Server 数据库作为来源 AWS DMS](#)。
- 修复了使用 Oracle Standby 作为源，在使用 Binary Reader 时 CDC 任务会意外停止的问题。
- 修复了 IBM DB2 for LUW 中，任务失败并显示消息“数字文本 0 无效，因为其值超出范围”的问题。
- 修复了在以下情况中出现的问题：从 PostgreSQL 迁移到 PostgreSQL，当 PostgreSQL 源中添加了新列，并且创建该列使用的数据类型，与最初在源上创建该列的数据类型不同时。
- 修复了 MySQL 源中，当迁移任务在无法获取二进制日志时意外停止的问题。
- 修复了使用 BatchApply 时与 Oracle 目标相关的问题。
- 修复了迁移 TIME 数据类型时 MySQL 和 MariaDB 中的问题。
- 修复了将 IBM DB2 LUW 作为源，当表没有主键或唯一键时迁移带有 LOB 的表会失败的问题。

## AWS Database Migration Service 3.3.4 版本说明

AWS DMS 3.3.4 中解决的问题包括以下内容：

- 修复了使用 PostgreSQL 作为源时事务被丢弃或重复的问题。
- 改进了对在架构名称中使用美元符号 ( \$ ) 的支持。
- 修复了使用 RDS SQL Server 作为源时，复制实例不关闭未完成事务的问题。
- 修复了使用 PostgreSQL 作为源且密码中包含特殊字符时，测试连接失败的问题。
- 通过更好地处理以下错误，改进了将 Amazon Redshift 作为目标时的性能：“指定的密钥不存在”。
- 改进了对各种数据类型和字符集的数据验证支持。
- 修复了将 Oracle 作为目标时，复制任务尝试创建现有表的问题。
- 修复了将 Amazon S3 作为目标时，在特定条件下复制任务不会为 "OP" 列添加 "D" 的问题。

## AWS Database Migration Service 3.3.3 发行说明

下表显示了 AWS DMS 版本 3.3.3 中引入的新功能和增强功能。

新功能或增强功能	描述
新 PostgreSQL 版本	现在支持将 PostgreSQL 版本 12 作为源和目标。
支持以亚马逊 OpenSearch 服务为目标的复合主密钥	从 AWS DMS 3.3.3 开始，OpenSearch 服务目标支持使用复合主键。
支持 Oracle 扩展数据类型	现在支持 Oracle 源和目标采用 Oracle 扩展数据类型。
增加每个账户的 AWS DMS 资源数量	您可以创建的 AWS DMS 资源数量限制已增加。有关更多信息，请参阅 <a href="#">AWS Database Migration Service 的限额</a> 。

AWS DMS 3.3.3 中解决的问题包括以下内容：

- 修复了在 Amazon Kinesis 中通过“并行应用”使用特定更新语句时任务崩溃的问题。
- 修复了以 Amazon S3 为目标执行 ALTER TABLE 语句时任务崩溃的问题。
- 修复了使用 Microsoft SQL Server 作为源时多边形列上的值被截断的问题。
- 修复了使用 Oracle 作为源时 JA16SJISTILDE 和 JA16EUCTILDE 的 Unicode 转换器存在的问题。
- 修复了 MEDIUMTEXT 列和 LONGTEXT 列无法从 MySQL 迁移到 S3 逗号分隔值 (CSV) 格式的问题。
- 修复了 Apache Parquet 输出中布尔列被转换为不正确类型的问题。
- 修复了 Oracle 中扩展 varchar 列的问题。
- 修复了特定的时间戳组合导致数据验证任务失败的问题。
- 修复了 Sybase 数据定义语言 (DDL) 复制的问题。
- 修复了 Oracle Binary Reader 发生 Oracle Real Application Clusters (RAC) 源崩溃的问题。
- 修复了 Oracle 目标验证中与架构名称大小写有关的问题。
- 修复了 IBM Db2 版本 9.7 和 10 的验证问题。
- 修复了启用 StopTaskCachedChangesNotApplied 和 StopTaskCachedChangesApplied 后任务不会停止两次的问题。

# 文档历史记录

下表说明了自 2018 年 1 月以来对《AWS Database Migration Service 用户指南》文档的重要更改。

您可以订阅 RSS 源获取此文档更新的通知。有关 AWS DMS 发行版本的更多详细信息，请参阅 [AWS DMS 发行说明](#)。

变更	说明	日期
<a href="#">AWS DMS 添加了对将 RDS IBM DB2 作为目标的支持</a>	AWS DMS 现在支持使用 Amazon RDS IBM DB2 作为目标。	2023 年 12 月 4 日
<a href="#">AWS DMS 添加了对将 Timestream 作为目标的支持。</a>	AWS DMS 现在支持将 Timestream 作为目标。	2023 年 11 月 17 日
<a href="#">AWS DMS 添加了对 Redshift 目标数据验证的支持</a>	AWS DMS 现在支持验证 Redshift 目标中的数据。	2023 年 11 月 14 日
<a href="#">AWS DMS 增加了对四种新端点类型的支持</a>	AWS DMS 现在支持将 Microsoft Azure Database for PostgreSQL、Microsoft Azure Database for MySQL、OCI MySQL Heatwave 和 Google Cloud for PostgreSQL 作为源。	2023 年 10 月 26 日
<a href="#">AWS DMS 增加了对新的 AWS 服务相关角色的支持</a>	AWS DMS 现在支持 AWS 服务相关角色 <code>AWSServiceRoleForDMSServerless</code> ，该角色允许 AWS DMS 代表您创建和管理资源，例如将指标数据点发布到 Amazon CloudWatch。	2023 年 5 月 22 日
<a href="#">AWS DMS 增加了对新的 AWS 托管策略的支持</a>	AWS DMS 现在支持允许将无服务器复制日志发布到	2023 年 5 月 22 日

	CloudWatch Logs 的 AWS 托管策略。	
<a href="#">AWS DMS 增加了对新的 AWS 托管策略的支持</a>	AWS DMS 现在支持允许将指标数据点发布到 Amazon CloudWatch 的 AWS 托管策略。此外，AWS DMS 为其 AWS 托管策略开启了跟踪更改。	2023 年 3 月 6 日
<a href="#">支持 VPC 源端点和目标端点</a>	AWS DMS 现在支持将虚拟私有云 (VPC) 端点作为源和目标。AWS DMS 现在可通过 VPC 端点连接到任何 AWS 服务，只要在 AWS DMS VPC 中明确定义了到服务的路由。	2022 年 6 月 30 日
<a href="#">支持将 SQL Server 只读副本作为源</a>	AWS DMS 现在支持将 SQL Server 只读副本作为源。使用 AWS DMS，您现在可以执行从 SQL Server 只读副本到任何 AWS DMS 支持的目标的实时迁移。	2022 年 6 月 30 日
<a href="#">支持将 IBM Db2 z/OS 数据库作为仅完全加载的源</a>	AWS DMS 现在支持将 IBM Db2 z/OS 数据库作为源。使用 AWS DMS，您现在可以执行从 Db2 大型机到任何支持 AWS DMS 的目标的实时迁移。	2022 年 6 月 30 日
<a href="#">支持 EventBridge DMS 事件</a>	AWS DMS 支持使用 EventBridge 管理 DMS 事件的事件订阅。	2022 年 6 月 30 日

<a href="#">支持将 Babelfish 作为目标</a>	AWS DMS 现在支持将 Babelfish 作为目标。借助 AWS DMS，您现在可以将实时数据从任何 AWS DMS 支持的源迁移到 Babelfish，同时最大程度减少停机时间。	2022 年 6 月 30 日
<a href="#">支持将 Aurora Serverless v2 作为目标</a>	AWS DMS 现在支持将 Aurora Serverless v2 作为目标。使用 AWS DMS，您现在可以执行到 Aurora Serverless v2 的实时迁移。	2022 年 6 月 30 日
<a href="#">入门教程</a>	AWS DMS 入门教程的更新。本教程使用 MySQL 数据库作为源，使用 PostgreSQL 数据库作为目标。	2021 年 5 月 20 日
<a href="#">支持将 Amazon Neptune 作为目标</a>	增加了对将 Amazon Neptune 作为数据迁移目标的支持。	2020 年 6 月 1 日
<a href="#">支持将 Apache Kafka 作为目标</a>	增加了对 Apache Kafka 作为数据迁移目标的支持。	2020 年 3 月 20 日
<a href="#">更新了安全性内容</a>	更新并标准化安全内容，以响应客户请求。	2019 年 12 月 20 日
<a href="#">使用 AWS Snowball Edge 进行迁移</a>	增加了对使用 AWS Snowball Edge 迁移大型数据库的支持。	2019 年 1 月 24 日
<a href="#">支持将 Amazon DocumentDB (与 MongoDB 兼容) 作为目标</a>	增加了对将 Amazon DocumentDB (与 MongoDB 兼容) 作为数据迁移目标的支持。	2019 年 1 月 9 日

---

<a href="#">支持将 Amazon OpenSearch Service 和 Amazon Kinesis Data Streams 作为目标</a>	增加了对将 OpenSearch Service 和 Kinesis Data Streams 作为数据迁移目标的支持。	2018 年 11 月 15 日
<a href="#">CDC 本机开始支持</a>	增加了在使用更改数据捕获 (CDC) 时对本机开始点的支持。	2018 年 6 月 28 日
<a href="#">Db2 LUW 支持</a>	增加了对将 IBM Db2 LUW 用作数据迁移源的支持。	2018 年 4 月 26 日
<a href="#">支持将 SQL Server 作为目标</a>	增加了对将 Amazon RDS for Microsoft SQL Server 用作源的支持。	2018 年 2 月 6 日



# AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 [AWS 词汇表](#)。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。