Manual do usuário

AWS Database Migration Service

AWS Database Migration Service: Manual do usuário

Table of Contents

O que é o AWS Database Migration Service?

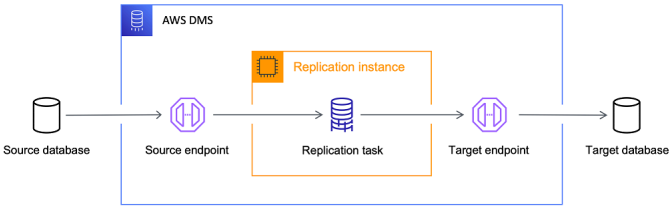
O AWS Database Migration Service (AWS DMS) é um serviço em nuvem que possibilita a migração de bancos de dados relacionais, data warehouses, bancos de dados NoSQL e outros tipos de datastores. É possível utilizar o AWS DMS para migrar os dados para a Nuvem AWS ou entre combinações configurações de nuvem e on-premises.

Com o AWS DMS, é possível descobrir os datastores de origem, converter os esquemas de origem e migrar os dados.

- Para descobrir a infraestrutura dos dados de origem, é possível utilizar o DMS Fleet Advisor. Esse serviço coleta dados do banco de dados e de servidores analíticos on-premises e cria um inventário de servidores, bancos de dados e esquemas que podem ser migrados para a nuvem AWS.
- Para migrar para um mecanismo de banco de dados diferente, é possível utilizar a DMS Schema Conversion. Esse serviço avalia e converte automaticamente os esquemas de origem em um novo mecanismo de destino. Como alternativa, é possível baixar o AWS Schema Conversion Tool (AWS SCT) no PC local para converter os esquemas de origem.
- Depois de converter os esquemas de origem e aplicar o código convertido ao banco de dados de destino, é possível utilizar o AWS DMS para migrar os dados. É possível executar migrações avulsas e replicar as alterações em andamento para manter as origens e os destinos em sincronia. Como o AWS DMS faz parte da Nuvem AWS, você obtém eficiência de custos, velocidade de comercialização, segurança e flexibilidade que os serviços da AWS oferecem.

Em um nível básico, o AWS DMS é um servidor na Nuvem AWS que executa software de replicação. Você cria uma conexão de origem e de destino para informar ao AWS DMS de onde extrair e para onde carregar. E programa uma tarefa que é executada nesse servidor para mover os dados. O AWS DMS criará as tabelas e as chaves primárias associadas se ainda não existirem no destino. É possível criar as tabelas de destino manualmente, se preferir. Ou utilizar o AWS Schema Conversion Tool (AWS SCT) para criar algumas ou todas as tabelas, índices, visualizações, acionadores e assim por diante de destino.

O diagrama a seguir ilustra o processo de replicação do AWS DMS.



Referências

- Regiões da AWS compatíveis com o AWS DMS: para obter informações sobre quais regiões da AWS são compatíveis com o AWS DMS, consulte Trabalhando com uma instância de AWS DMS replicação.
- Custo da migração do banco de dados: para obter mais informações sobre o custo da migração do banco de dados, consulte a página Preços do AWS Database Migration Service.
- Recursos e benefícios do AWS DMS: para obter informações sobre os recursos e benefícios do AWS DMS, consulte Recursos do AWS Database Migration Service.
- Opções dos bancos de dados disponíveis: para saber mais sobre a variedade de opções de bancos de dados disponíveis na Amazon Web Services, consulte Como escolher o banco de dados certo para a sua organização.

Tarefas de migração executadas pelo AWS DMS

O AWS DMS assume muitas das tarefas difíceis ou tediosas envolvidas em um projeto de migração:

- Em uma solução tradicional, você precisa executar análises de capacidade, adquirir hardware e software, instalar e administrar sistemas e testar e depurar a instalação. O AWS DMS gerencia automaticamente a implantação, o gerenciamento e o monitoramento de todo o hardware e de todo o software necessários para a migração. A migração pode estar ativa e em execução em minutos depois do início do processo de configuração do AWS DMS.
- Com o AWS DMS, é possível aumentar a escala verticalmente (ou reduzir) os recursos de migração conforme necessário para que correspondam à sua workload real. Por exemplo, se você determinar que precisa de armazenamento adicional, poderá aumentar facilmente o armazenamento alocado e reiniciar a migração, geralmente em minutos.
- O AWS DMS utiliza um modelo de pagamento conforme o uso. Você paga pelos recursos do AWS DMS apenas quando os usa, em vez dos modelos de licenciamento tradicionais com custos de compra iniciais e cobranças contínuas de manutenção.

- O AWS DMS gerencia automaticamente toda a infraestrutura que é compatível com o servidor de migração, incluindo hardware e software, aplicação de patches de software e relatórios de erros.
- O AWS DMS fornece failover automático. Se seu servidor de replicação primário falhar por qualquer motivo, um servidor de replicação de backup poderá assumir com pouca ou nenhuma interrupção do serviço.
- O AWS DMS Fleet Advisor faz o inventário automático da sua infraestrutura de dados. Ele cria relatórios que ajudam a identificar candidatos à migração e a planejar a migração.
- A AWS DMS Schema Conversion avalia automaticamente a complexidade da migração para o provedor de dados de origem. Ela também converte esquemas do banco de dados e objetos de código em um formato compatível com o banco de dados de destino e aplica o código convertido.
- O AWS DMS pode ajudar você a alternar para um mecanismo de banco de dados moderno, talvez mais econômico, do que o mecanismo de banco de dados que você está executando agora. Por exemplo, o AWS DMS pode ajudar a aproveitar os serviços de banco de dados gerenciados fornecidos pelo Amazon Relational Database Service (Amazon RDS) ou pelo Amazon Aurora. Ou pode ajudar a migrar para o serviço de data warehouse gerenciado fornecido pelo Amazon Redshift, plataformas NoSQL, como o Amazon DynamoDB, ou plataformas de armazenamento de baixo custo, como o Amazon Simple Storage Service (Amazon S3). Por outro lado, se você quiser migrar da infraestrutura antiga, mas continuar a utilizar o mesmo mecanismo de banco de dados, o AWS DMS também é compatível com esse processo.
- O AWS DMS é compatível com quase todos os mecanismos de DBMS mais populares atuais como endpoints de origem. Para obter mais informações, consulte Origens para a migração de dados.
- O AWS DMS fornece uma ampla cobertura de mecanismos destino disponíveis. Para obter mais informações, consulte Destinos para a migração de dados.
- É possível migrar de qualquer uma das fontes de dados compatíveis para qualquer um dos destinos de dados compatíveis. O AWS DMS é compatível com migrações de dados completamente heterogêneas entre os mecanismos compatíveis.
- O AWS DMS garante que a migração dos seus dados seja segura. Os dados em repouso são criptografados com criptografia AWS Key Management Service (AWS KMS). Durante a migração, é possível utilizar o Secure Socket Layers (SSL) para criptografar os dados em trânsito enquanto viajam da origem para o destino.

Como funciona o AWS Database Migration Service

AWS Database Migration Service (AWS DMS) é um serviço web que você pode usar para migrar dados de um armazenamento de dados de origem para um armazenamento de dados de destino. Esses dois datastore são chamados de endpoints. É possível migrar entre endpoints de origem e de destino que usam o mesmo mecanismo de banco de dados, como de um banco de dados Oracle para um banco de dados Oracle. Você também pode migrar entre endpoints de origem e de destino que usam mecanismos de banco de dados diferentes, como de um banco de dados Oracle para um banco de dados PostgreSQL. O único requisito a ser usado AWS DMS é que um dos seus endpoints esteja em um AWS serviço. Você não pode usar AWS DMS para migrar de um banco de dados local para outro banco de dados local.

Para obter mais informações sobre o custo da migração de bancos de dados, consulte a página Definição de preços do AWS Database Migration Service.

Use os tópicos a seguir para entender melhor AWS DMS.

Tópicos

Visão de alto nível de AWS DMS

Para realizar uma migração de banco de dados, AWS DMS conecte-se ao armazenamento de dados de origem, lê os dados de origem e formata os dados para consumo pelo armazenamento de dados de destino. Ele carrega os dados no datastore de destino. A maior parte desse processo ocorre na memória, mas grandes transações podem exigir buffer para o disco. Transações armazenadas em cache e arquivos de log também são gravados no disco.

Em um nível alto, ao usar, AWS DMS você faz o seguinte:

- Descubra os bancos de dados no ambiente de rede que são bons candidatos à migração.

- Converta automaticamente os esquemas do banco de dados de origem e a maioria dos objetos de código do banco de dados em um formato compatível com o banco de dados de destino.
- Crie um servidor de replicação.
- Crie endpoints de origem e de destino que tenham informações de conexão sobre os datastores.
- Crie uma ou mais tarefas de migração para migrar dados entre os datastores de origem e de destino.

Uma tarefa pode consistir em três fases principais:

- Migração de dados existentes (carga máxima)
- A aplicação de alterações armazenadas em cache
- Replicação contínua (captura de dados de alteração)

Durante uma migração de carga total, em que os dados existentes da origem são movidos para o destino, AWS DMS carrega dados das tabelas no armazenamento de dados de origem para as tabelas no armazenamento de dados de destino. Enquanto a carga máxima está em andamento, as alterações feitas nas tabelas que estão sendo carregadas são armazenadas em cache no servidor de replicação: essas são as alterações armazenadas em cache. É importante observar que AWS DMS não captura as alterações de uma determinada tabela até que o carregamento completo dessa tabela seja iniciado. Em outras palavras, o ponto onde a captura de alterações começa é diferente para cada tabela individual.

Quando a carga completa de uma determinada tabela é concluída, AWS DMS imediatamente começa a aplicar as alterações em cache para essa tabela. Depois que a tabela é carregada e as alterações em cache são aplicadas, AWS DMS começa a coletar as alterações como transações para a fase de replicação contínua. Se uma transação tiver tabelas ainda não totalmente carregadas, as alterações serão armazenadas localmente na instância de replicação. Depois de AWS DMS aplicar todas as alterações em cache a todas as tabelas, as tabelas ficam transacionalmente consistentes. Nesse ponto, AWS DMS passa para a fase de replicação contínua, aplicando as alterações como transações.

No início dessa fase, uma lista de pendências de transações costuma causar atrasos entre os bancos de dados de origem e de destino. A migração acaba alcançando um estado estável após trabalhar nessa lista de pendências de transações. Nesse momento, é possível desligar suas aplicações, permitir que as transações restantes sejam aplicadas ao destino e atualizar as aplicações, agora apontando para o banco de dados de destino.

AWS DMS cria os objetos do esquema de destino necessários para realizar uma migração de dados. Você pode usar AWS DMS para adotar uma abordagem minimalista e criar somente os objetos necessários para migrar os dados com eficiência. Usando essa abordagem, AWS DMS cria tabelas, chaves primárias e, em alguns casos, índices exclusivos, mas não cria nenhum outro objeto que não seja necessário para migrar com eficiência os dados da fonte.

Como alternativa, você pode usar a Conversão de Esquema DMS AWS DMS para converter automaticamente os esquemas do banco de dados de origem e a maioria dos objetos de código do banco de dados em um formato compatível com o banco de dados de destino. Essa conversão inclui tabelas, visualizações, procedimentos armazenados, perfis, tipos de dados, sinônimos e assim por diante. Todos os objetos que a DMS Schema Conversion não pode converter automaticamente são claramente marcados. Para concluir a migração, é possível converter esses objetos manualmente.

Componentes do AWS DMS

Esta seção descreve os componentes internos AWS DMS e como eles funcionam juntos para realizar sua migração de dados. Entender os componentes estruturais do AWS DMS pode ajudar você a migrar dados de forma mais eficaz e fornecer uma melhor compreensão ao solucionar ou investigar problemas.

Uma AWS DMS migração consiste em cinco componentes: descoberta de bancos de dados a serem migrados, conversão automática de esquemas, instância de replicação, endpoints de origem e destino e tarefa de replicação. Você cria uma AWS DMS migração criando a instância de replicação, os endpoints e as tarefas necessárias em um. Região da AWS

Descoberta de banco de dados

O DMS Fleet Advisor coleta dados de vários ambientes de banco de dados para fornecer insight da infraestrutura de dados. O DMS Fleet Advisor coleta dados do banco de dados on-premises e de servidores analíticos de um ou mais locais centrais sem a necessidade de instalá-los em todos os computadores. Atualmente, o DMS Fleet Advisor é compatível com o Microsoft SQL Server, o MySQL, o Oracle e os servidores de banco de dados PostgreSQL.

Com base nos dados descobertos na rede, o DMS Fleet Advisor cria um inventário que é possível analisar para determinar quais servidores e objetos do banco de dados devem ser monitorados. À medida que os detalhes sobre esses servidores, bancos de dados e esquemas são coletados, é possível analisar a viabilidade das migrações de banco de dados pretendidas.

Migração de código e schema

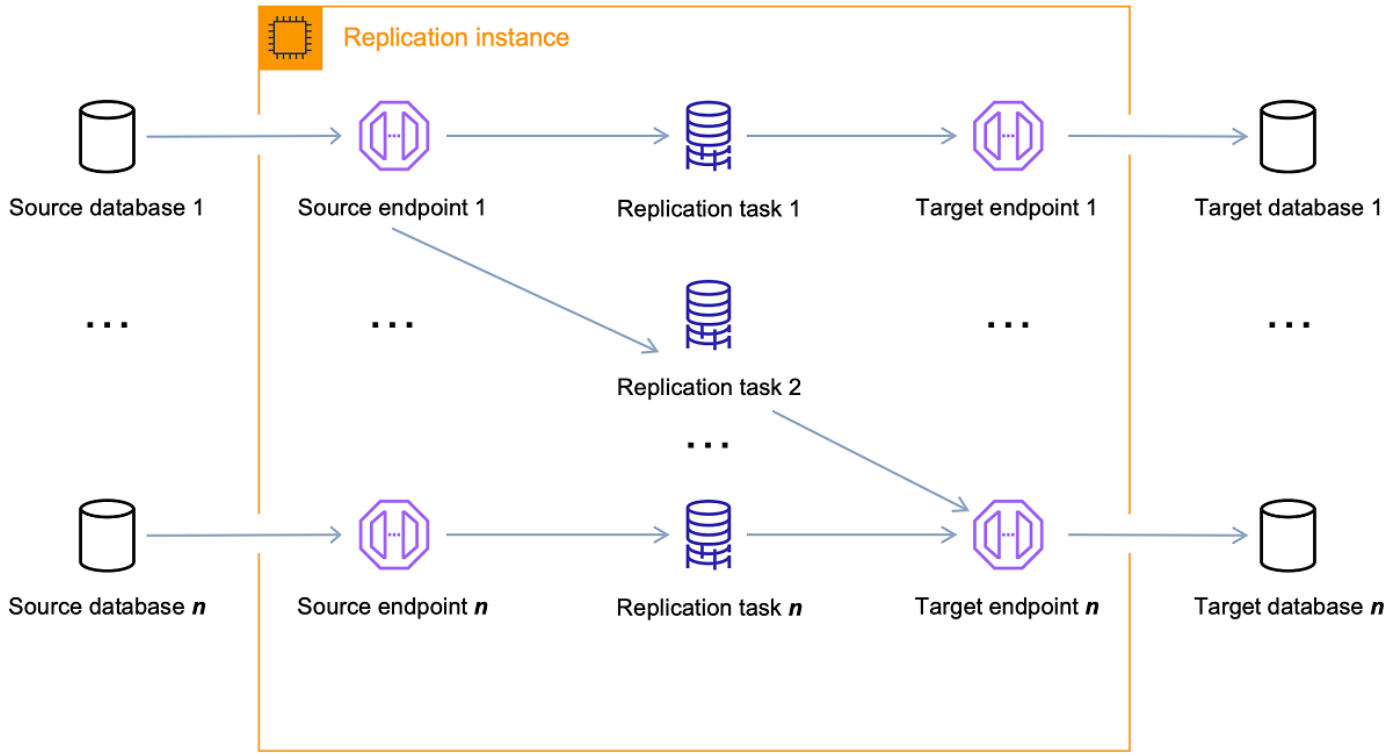
A conversão do esquema DMS AWS DMS torna as migrações de banco de dados entre diferentes tipos de bancos de dados mais previsíveis. É possível utilizar a DMS Schema Conversion para avaliar a complexidade da migração para o provedor de dados de origem e usá-la para converter esquemas de banco de dados e objetos de código. É possível aplicar o código convertido ao banco de dados de destino.

Em alto nível, a DMS Schema Conversion opera com os três componentes a seguir: perfis de instância, provedores de dados e projetos de migração. Um perfil de instância especifica as configurações da rede e da segurança. Um provedor de dados armazena as credenciais de conexão do banco de dados. Um projeto de migração contém provedores de dados, um perfil de instância e regras de migração. AWS DMS usa provedores de dados e um perfil de instância para criar um processo que converte esquemas de banco de dados e objetos de código.

Instância da replicação

Em um alto nível, uma instância de AWS DMS replicação é simplesmente uma instância gerenciada do Amazon Elastic Compute Cloud (Amazon EC2) que hospeda uma ou mais tarefas de replicação.

A figura a seguir mostra um exemplo de instância de replicação que executa várias tarefas de replicação associadas.



Uma única instância de replicação pode hospedar uma ou mais tarefas de replicação, dependendo das características da migração e da capacidade do servidor de replicação. AWS DMS fornece uma variedade de instâncias de replicação para que você possa escolher a configuração ideal para seu caso de uso. Para obter mais informações sobre as diversas classes de instâncias de replicação, consulte Escolhendo a instância de replicação AWS DMS certa para sua migração.

AWS DMS cria a instância de replicação em uma instância do Amazon EC2. Algumas das classes de instância menores são suficientes para testar o serviço ou para migrações pequenas. Se a migração envolver um grande número de tabelas ou se você pretender executar várias tarefas de replicação simultâneas, considere utilizar uma das instâncias maiores. É recomendável essa abordagem porque o AWS DMS pode consumir uma quantidade significativa de memória e de CPU.

Dependendo da classe da instância do Amazon EC2 selecionada, a instância de replicação vem com 50 GB ou 100 GB de armazenamento de dados. Essa quantidade normalmente é suficiente para a maioria dos clientes. No entanto, se a migração envolve grandes transações ou um grande volume de alterações de dados, talvez deseje aumentar a alocação de armazenamento de base. A captura de dados de alteração (CDC) pode fazer com que os dados sejam gravados em disco, de acordo com a velocidade com que o destino consegue gravar as alterações. Como os arquivos de log também são gravados em disco, aumentar o nível de gravidade do registro em log também resultará em um maior consumo de armazenamento.

AWS DMS pode fornecer alta disponibilidade e suporte de failover usando uma implantação Multi-AZ. Em uma implantação Multi-AZ, provisiona e mantém AWS DMS automaticamente uma réplica em espera da instância de replicação em uma zona de disponibilidade diferente. A instância de replicação primária é replicada em sincronia para a réplica em espera. Se a instância de replicação primária falhar ou parar de responder, a instância em espera retoma qualquer tarefa em execução com o mínimo de interrupção. Como a primária está replicando constantemente seu estado para a de espera, a implantação multi-AZ implica alguma sobrecarga no desempenho.

Para obter informações mais detalhadas sobre a instância AWS DMS de replicação, consulteTrabalhando com uma instância de AWS DMS replicação.

Em vez de criar e gerenciar uma instância de replicação, você pode deixar AWS DMS provisionar sua replicação automaticamente usando AWS DMS o Serverless. Para ter mais informações, consulte Trabalhando com AWS DMS Serverless.

Endpoint

AWS DMS usa um endpoint para acessar seu armazenamento de dados de origem ou destino. As informações de conexão específicas são diferentes, dependendo do seu armazenamento de dados, mas no geral deve-se fornecer as seguintes informações ao criar um endpoint.

- Tipo de endpoint: origem ou destino.
- Tipo de mecanismo: tipo de mecanismo do banco de dados, como Oracle ou PostgreSQL.
- Nome do servidor — Nome do servidor ou endereço IP que AWS DMS pode ser acessado.
- Porta: número da porta usada para conexões de servidor do banco de dados.
- Criptografia: modo Secure Socket Layer (SSL), se o SSL for usado para criptografar a conexão.
- Credenciais: nome de usuário e senha de uma conta com os direitos de acesso necessários.

Quando você cria um endpoint usando o AWS DMS console, o console exige que você teste a conexão do endpoint. O teste deve ser bem-sucedido antes de usar o endpoint em uma AWS DMS tarefa. Como as informações de conexão, os critérios de teste específicos são diferentes para diferentes tipos de mecanismos. No geral, o AWS DMS verifica se o banco de dados existe no determinado nome de servidor e porta, e se as credenciais fornecidas podem ser usadas para se conectar ao banco de dados com os privilégios necessários para executar uma migração. Se o teste de conexão for bem-sucedido, AWS DMS baixa e armazena as informações do esquema para uso posterior durante a configuração da tarefa. As informações de esquema podem incluir definições de tabela, definições de chave primária e definições de chave exclusiva, por exemplo.

Mais de uma tarefa de replicação pode utilizar um único endpoint. Por exemplo, é possível ter duas aplicações logicamente distintas hospedadas no mesmo banco de dados de origem que deseja migrar separadamente. Nesse caso, duas tarefas de replicação são criadas, uma para cada conjunto de tabelas de aplicações. Você pode usar o mesmo AWS DMS endpoint nas duas tarefas.

É possível personalizar o comportamento de um endpoint utilizando as configurações do endpoint. As Configurações do endpoint podem controlar vários comportamentos, como detalhes de registro em log, tamanho do arquivo e outros parâmetros. Cada tipo de mecanismo de datastore tem diferentes configurações de endpoint disponíveis. É possível encontrar as configurações específicas de endpoints para cada datastore na seção de origem e de destino do datastore. Para obter uma lista de datastores de origem e de destino compatíveis, consulte Fontes para AWS DMS e Metas para AWS DMS.

Para obter informações mais detalhadas sobre AWS DMS endpoints, consulteComo trabalhar com endpoints do AWS DMS.

Tarefas de replicação

Você usa uma tarefa de AWS DMS replicação para mover um conjunto de dados do endpoint de origem para o endpoint de destino. A criação de uma tarefa de replicação é a última etapa necessária antes de iniciar uma migração.

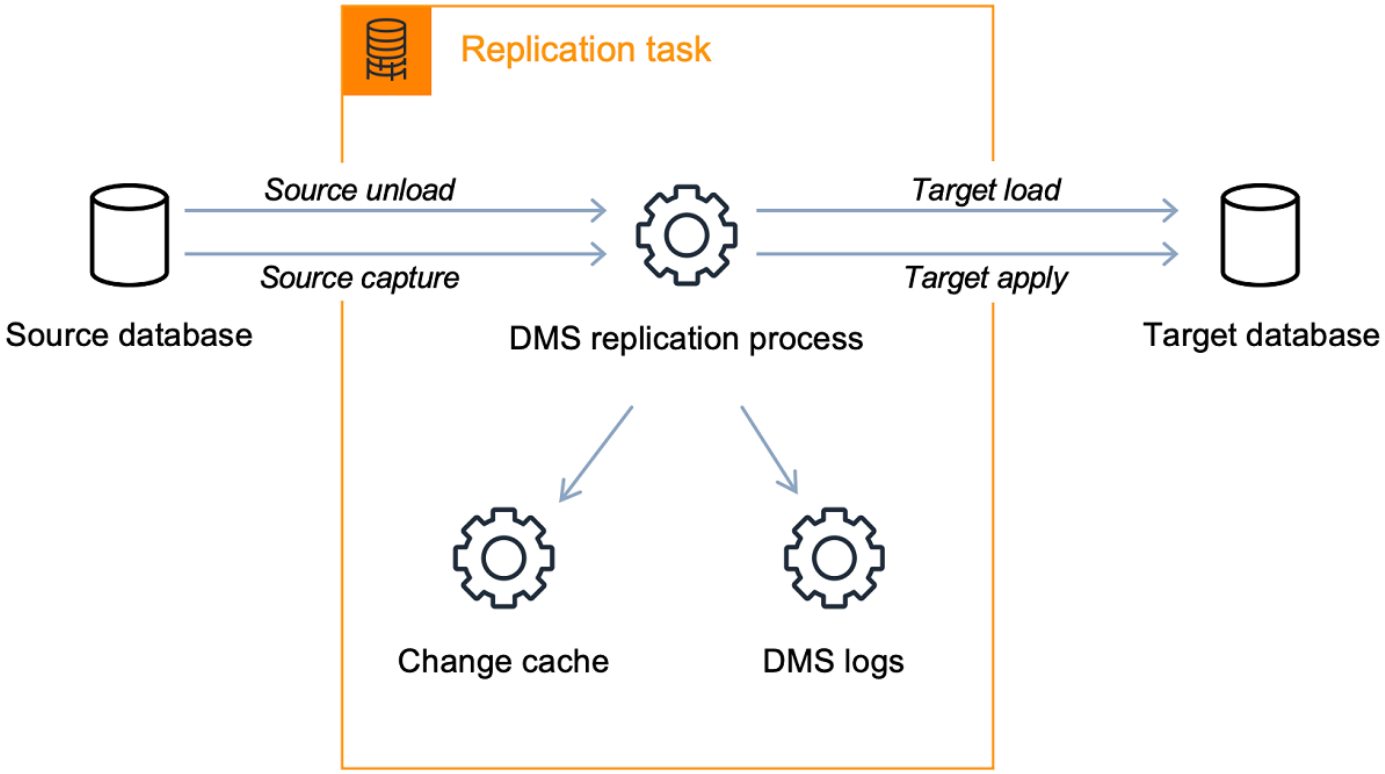
Ao criar uma tarefa de replicação, especifique as seguintes configurações de tarefa:

- Instância de replicação: a instância que hospeda e executa a tarefa
- Endpoint de origem
- Endpoint de destino
- Opções de tipo de migração, conforme listadas a seguir. Para obter uma explicação completa sobre as opções do tipo de migração, consulte Criar uma tarefa.
  - Carga máxima (migração de dados existentes): se houver condições para uma interrupção longa o suficiente para copiar os dados existentes, esta será uma boa opção. Essa opção migra somente os dados do banco de dados de origem para o banco de dados de destino, criando tabelas quando necessário.
  - Carga máxima + CDC (alterações da migração dos dados existentes e da replicação contínua): esta opção executa uma carga máxima de dados enquanto captura alterações na origem. Após a carga máxima ser concluída, as alterações capturadas são aplicadas ao destino. Por fim, a aplicação de alterações alcança um estado estável. Nesse momento, é possível encerrar as aplicações, permitir que as alterações restantes sejam transmitidas até o destino e reiniciar as aplicações apontando para o destino.
  - Somente CDC (somente replicação de alterações de dados): em algumas situações, copiar os dados existentes utilizando um método diferente do AWS DMS pode ser mais eficaz. Por exemplo, em uma migração homogênea, utilizar ferramentas de exportação/importação nativas pode ser mais eficaz para o carregamento de dados em massa. Nessa situação, você pode usar AWS DMS para replicar as alterações a partir do momento em que inicia o carregamento em massa para trazer e manter seus bancos de dados de origem e destino sincronizados.
- Opções de modo de preparação de tabela de destino, conforme listadas a seguir. Para obter uma explicação completa sobre os modos da tabela de destino, consulte Criar uma tarefa.
  - Não fazer nada — AWS DMS presume que as tabelas de destino foram pré-criadas no destino.
  - Solte as tabelas no alvo — AWS DMS derruba e recria as tabelas de destino.

    - Truncar: se você criou tabelas no destino, o AWS as truncará antes que a migração seja iniciada. Se nenhuma tabela existir e você selecionar essa opção, AWS DMS criará qualquer tabela ausente.
- Opções do modo LOB, conforme listadas a seguir. Para obter uma explicação completa sobre os modos de LOB, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS.
    - Não incluir colunas LOB: as colunas LOB são excluídas da migração.
    - Modo LOB completo — Migre LOBs completos, independentemente do tamanho. AWS DMS migra LOBs por partes em partes controladas pelo parâmetro Max LOB Size. Esse modo é mais lento que o modo LOB limitado.
    - Modo LOB limitado: trunca LOBs até o valor especificado pelo parâmetro Tamanho máximo de LOB. Esse modo é mais rápido que o modo LOB completo.
- Mapeamentos de tabelas: indica as tabelas a serem migradas e como elas são migradas. Para ter mais informações, consulte Utilizar o mapeamento de tabela para especificar as configurações da tarefa.
- Transformações de dados, conforme listadas a seguir. Para obter mais informações sobre transformações de dados, consulte Especificar a seleção de tabelas e as regras de transformação utilizando JSON.
    - Alteração de nomes de esquema, tabela e coluna.
    - Alteração de nomes de espaços de tabela (para endpoints de destino do Oracle).
    - Definição de chaves primárias e índices exclusivos no destino.
- Validação de dados
- CloudWatch Registro na Amazon

Utilize a tarefa para migrar dados do endpoint de origem para o endpoint de destino, e o processamento da tarefa acontece na instância de replicação. Especifique quais tabelas e esquemas serão migrados e qualquer processamento especial, como requisitos de registro em log, dados da tabela de controle e gerenciamento de erros.

Conceitualmente, uma tarefa de AWS DMS replicação executa duas funções distintas, conforme mostrado no diagrama a seguir.



O processo de carga máxima apresenta compreensão direta. Os dados são extraídos da origem em um modo de extração em lote e carregados diretamente no destino. Você pode especificar o número de tabelas a serem extraídas e carregadas paralelamente no AWS DMS console em Configurações avançadas.

Para obter mais informações sobre AWS DMS tarefas, consulteTrabalhar com tarefas do AWS DMS.

Replicação contínua ou captura de dados de alteração (CDC)

Você também pode usar uma AWS DMS tarefa para capturar alterações contínuas no armazenamento de dados de origem enquanto migra seus dados para um destino. O processo de captura de alterações AWS DMS usado ao replicar alterações contínuas de um endpoint de origem coleta as alterações nos registros do banco de dados usando a API nativa do mecanismo de banco de dados.

No processo de CDC, a tarefa de replicação é projetada para transmitir as alterações da origem para o destino, utilizando buffers na memória para manter os dados em trânsito. Se os buffers na memória se esgotarem por qualquer motivo, a tarefa de replicação repassará alterações pendentes para o Cache de alteração no disco. Isso pode ocorrer, por exemplo, se AWS DMS

estiver capturando alterações da fonte mais rápido do que elas podem ser aplicadas no destino. Nesse caso, você verá a latência de destino da tarefa exceder a latência de origem.

Você pode verificar isso navegando até sua tarefa no AWS DMS console e abrindo a guia Monitoramento de tarefas. Os LatencySource gráficos do CDC LatencyTarget e do CDC são mostrados na parte inferior da página. Se você tiver uma tarefa que mostra a latência de destino, então provavelmente há algum ajuste no endpoint de destino necessário para aumentar a taxa de aplicação.

A tarefa de replicação também utiliza armazenamento para logs de tarefas, conforme discutido anteriormente. O espaço em disco que vem pré-configurado com a instância de replicação normalmente é suficiente para o registro em log e alterações repassadas. Se precisar de espaço adicional em disco, por exemplo, ao utilizar a depuração detalhada para investigar um problema de migração, é possível modificar a instância de replicação para alocar mais espaço.

Fontes para AWS DMS

Você pode usar diferentes armazenamentos de dados de origem em diferentes AWS DMS recursos. As seções a seguir contêm as listas de armazenamentos de dados de origem suportados para cada AWS DMS recurso.

Tópicos

Endpoints de origem da migração de dados

É possível utilizar os seguintes datastores como endpoints de origem da migração de dados utilizando o AWS DMS.

Bancos de dados on-premises e de instância do EC2

- Oracle versões 10.2 e superior (para versões 10.x), 11g e até 12.2, 18c e 19c para as edições Enterprise, Standard, Standard One e Standard Two

- Microsoft SQL Server versões 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019 e 2022.
  - As edições Enterprise, Standard, Workgroup, Developer e Web oferecem suporte à replicação de carga total.
  - As edições Enterprise, Standard (versão 2016 e superior) e Developer oferecem suporte à replicação CDC (contínua), além da carga total.
  - A edição Express não é compatível.
- MySQL, versões 5.5, 5.6, 5.7 e 8.0

> **Note**
>
> O suporte para o MySQL 8.0 como fonte está disponível nas AWS DMS versões 3.4.0 e superiores, exceto quando a carga útil da transação é compactada. O suporte ao Google Cloud for MySQL 8.0 como fonte está disponível nas AWS DMS versões 3.4.6 e superiores.

- MariaDB (suportado como fonte de dados compatível com MySQL) versões 10.0 (somente 10.0.24 e superior), 10.2, 10.3, 10.4, 10.5 e 10.6.

> **Note**
>
> O suporte para o MariaDB como fonte está disponível em AWS DMS todas as versões em que o MySQL é suportado.

- PostgreSQL versão 9.4 e superior (para versões 9.x), 10.x, 11.x, 12.x, 13.x 14.x, 15.x e 16.x.

> **Note**
>
> AWS DMS suporta apenas a versão 15.x do PostgreSQL nas versões 3.5.1 e superiores. AWS DMS suporta apenas a versão 16.x do PostgreSQL nas versões 3.5.3 e superiores.

- MongoDB versões 3.x, 4.0, 4.2, 4.4, 5.0 e 6.0

> **Note**
>
> AWS DMS as versões 3.5.0 e superiores não oferecem suporte às versões do MongoDB anteriores à 3.6.

- SAP Adaptive Server Enterprise (ASE) versões 12.5, 15, 15.5, 15.7, 16 e superior.

- IBM Db2 para Linux, UNIX e Windows (Db2 LUW) versões:
  - Versão 9.7, todos os fix packs
  - Versão 10.1, todos os fix packs
  - Versão 10.5, todos os fix packs, exceto o Fix Pack 5.
  - Versão 11.1, todos os fix packs
  - Versão 11.5, Mods (0-8) apenas com o Fix Pack Zero
- IBM Db2 for z/OS versão 12

Serviços de banco de dados gerenciados por terceiros:

- Banco de dados Microsoft Azure SQL
- Microsoft Azure PostgreSQL Flexible Server versões 11.2, 12.15, 13.11, 14.8 e 15.3.
- Microsoft Azure MySQL Flexible Server versões 5.7 e 8.
- Google Cloud for MySQL versões 5.6, 5.7 e 8.0.
- Google Cloud for PostgreSQL versões 9.6, 10, 11, 12, 13, 14 e 15.
- OCI MySQL Heatwave versão 8.0.34.

Bancos de dados de instância do Amazon RDS e do Amazon Simple Storage Service (Amazon S3)

- Oracle versões 11g (versões 11.2.0.4 e superior) e até 12.2, 18c e 19c para as edições Enterprise, Standard, Standard One e Standard Two
- Microsoft SQL Server versões 2012, 2014, 2016, 2017, 2019 e 2022 para as edições Enterprise, Standard, Workgroup e Developer

> **Note**
>
> AWS DMS não oferece suporte ao SQL Server Express. A edição web só é compatível com replicação de carga máxima.

- MySQL, versões 5.5, 5.6, 5.7 e 8.0

> **Note**
>
> O suporte para o MySQL 8.0 como fonte está disponível nas AWS DMS versões 3.4.0 e superiores, exceto quando a carga útil da transação é compactada.

- MariaDB (compatível como uma fonte de dados compatível com MySQL) versões 10.0.24 a 10.0.28, 10.2, 10.3, 10.4, 10.5 e 10.6.

> **Note**
>
> O suporte para o MariaDB como fonte está disponível em AWS DMS todas as versões em que o MySQL é suportado.

- PostgreSQL versão 10.x, 11.x, 12.x, 13.x, 14.x, 15.x e 16.x.

> **Note**
>
> AWS DMS suporta apenas a versão 15.x do PostgreSQL nas versões 3.5.1 e superiores. AWS DMS suporta apenas a versão 16.x do PostgreSQL nas versões 3.5.3 e superiores.

- Amazon Aurora compatível com o MySQL (compatível como uma fonte de dados compatível com MySQL)
- Amazon Aurora compatível com PostgreSQL (compatível como uma fonte de dados compatível com PostgreSQL).
- Amazon S3
- Amazon DocumentDB (com compatibilidade com MongoDB) versões 3.6, 4.0 e 5.0.
- Amazon RDS para IBM Db2 LUW.

Para obter informações sobre como trabalhar com uma fonte específica, consulte Trabalhando com AWS DMS endpoints.

Para obter informações sobre endpoints de destino compatíveis, consulte Endpoints de destino para a migração de dados.

Bancos de dados de origem para o DMS Fleet Advisor

O DMS Fleet Advisor é compatível com os seguintes bancos de dados de origem.

- Microsoft SQL Server versão 2012 e superior até 2019
- MySQL versão 5.6 e superior até a 8
- Oracle versão 11g Release 2 e superior até a 12c, 19c e 21c
- PostgreSQL versão 9.6 e superior até a 13

Para obter informações sobre como trabalhar com origens específicas, consulte Criar usuários do banco de dados para o AWS DMS Fleet Advisor.

Para obter a lista dos bancos de dados que o DMS Fleet Advisor utiliza para gerar recomendações de destino, consulte Destinos do DMS Fleet Advisor.

Provedores de dados de origem para a DMS Schema Conversion

A DMS Schema Conversion é compatível com os seguintes provedores de dados como origens para projetos de migração.

- Microsoft SQL Server versão 2008 R2, 2012, 2014, 2016, 2017 e 2019.
- Oracle versão 10.2 e posterior, 11g e até 12.2, 18c e 19c, e Oracle Data Warehouse
- PostgreSQL versão 9.2 e superior
- MySQL versão 5.5 e superior

O provedor de dados de origem pode ser um mecanismo autogerenciado em execução on-premises ou em uma instância do Amazon Elastic Compute Cloud (Amazon EC2).

Para obter informações sobre como trabalhar com origens específicas, consulte Criar provedores de dados de origem na DMS Schema Conversion.

Para obter informações sobre os bancos de dados de destino compatíveis, consulte Provedores de dados de destino para a DMS Schema Conversion.

O AWS Schema Conversion Tool (AWS SCT) suporta mais bancos de dados de origem e destino do que o DMS Schema Conversion. Para obter informações sobre bancos de dados que oferecem AWS SCT suporte, consulte O que é AWS Schema Conversion Tool o.

Provedores de dados de origem para migrações de dados homogêneas do DMS

É possível utilizar os seguintes provedores de dados como origens para migrações de dados homogêneas.

- MySQL versão 5.7 e superior
- MariaDB versão 10.2 e superior
- PostgreSQL versões 10.4 a 14.x.
- MongoDB versão 4.x, 5.x, 6.0
- Amazon DocumentDB versão 3.6, 4.0, 5.0

O provedor de dados de origem pode ser um mecanismo autogerenciado em execução on-premises ou em uma instância do Amazon EC2. Além disso, é possível utilizar uma instância de banco de dados Amazon RDS como um provedor de dados de origem.

Para obter informações sobre como trabalhar com origens específicas, consulte Criação de provedores de dados de origem para migrações de dados homogêneas no AWS DMS.

Para obter informações sobre os bancos de dados de destino compatíveis, consulte Provedores de dados de destino para migrações de dados homogêneas do DMS.

Metas para AWS DMS

Você pode usar diferentes armazenamentos de dados de destino em diferentes AWS DMS recursos. As seções a seguir contêm as listas de armazenamentos de dados de destino suportados para cada AWS DMS recurso.

Tópicos

- Endpoints de destino para a migração de dados
- Bancos de dados de destino do DMS Fleet Advisor
- Provedores de dados de destino para a DMS Schema Conversion
- Provedores de dados de destino para migrações de dados homogêneas do DMS

Endpoints de destino para a migração de dados

É possível utilizar os seguintes datastores como endpoints de destino para a migração de dados utilizando o AWS DMS.

Bancos de dados on-premises e instância do Amazon EC2

- Oracle versões 10g, 11g, 12c, 18c e 19c para as edições Enterprise, Standard, Standard One e Standard Two.
- Microsoft SQL Server versões 2005, 2008, 2008R2, 2012, 2014, 2016, 2017, 2019 e 2022 para as edições Enterprise, Standard, Workgroup e Developer.

> **Note**
>
> AWS DMS não oferece suporte às edições SQL Server Web e Express.

- MySQL, versões 5.5, 5.6, 5.7 e 8.0
- MariaDB (compatível com um destino de dados compatível com MySQL) versões 10.0.24 a 10.0.28, 10.2, 10.3, 10.4, 10.5 e 10.6.

> **Note**
>
> O suporte para o MariaDB como destino está disponível em AWS DMS todas as versões em que o MySQL é suportado.

- PostgreSQL versão 9.4 e superior (para versões 9.x), 10.x, 11.x, 12.x, 13.x, 14.x, 15.x e 16.x.

> **Note**
>
> AWS DMS só oferece suporte ao PostgreSQL 15.x nas versões 3.5.1 e superiores. AWS DMS suporta apenas a versão 16.x do PostgreSQL nas versões 3.5.3 e superiores.

- SAP Adaptive Server Enterprise (ASE) versões 15, 15.5, 15.7, 16 e superior.
- Redis versões 6.x

Bancos de dados de instâncias do Amazon RDS, Amazon Redshift, Amazon Redshift Serverless, Amazon DynamoDB, Amazon S3, Amazon Service, Amazon for Redis, Amazon Kinesis Data Streams, OpenSearch Amazon DocumentDB ElastiCache , Amazon Neptune e Apache Kafka

- Oracle versões 11g (versões 11.2.0.3.v1 e superior), 12c, 18c e 19c para as edições Enterprise, Standard, Standard One e Standard Two.
- Microsoft SQL Server versões 2012, 2014, 2016, 2017, 2019 e 2022 para as edições Enterprise, Standard, Workgroup e Developer

> **Note**
>
> AWS DMS não oferece suporte às edições SQL Server Web e Express.

- MySQL, versões 5.5, 5.6, 5.7 e 8.0
- MariaDB (compatível com um destino de dados compatível com MySQL) versões 10.0.24 a 10.0.28, 10.2, 10.3, 10.4, 10.5 e 10.6.

> **Note**
>
> O suporte para o MariaDB como destino está disponível em AWS DMS todas as versões em que o MySQL é suportado.

- PostgreSQL versão 10.x, 11.x, 12.x, 13.x, 14.x, 15.x e 16.x.

> **Note**
>
> AWS DMS só oferece suporte ao PostgreSQL 15.x nas versões 3.5.1 e superiores. AWS DMS só oferece suporte ao PostgreSQL 16.x nas versões 3.5.3 e superiores.

- IBM Db2 LUW versões 11.1 e 11.5
- Amazon Aurora Edição Compatível com MySQL
- Amazon Aurora Edição Compatível com PostgreSQL
- Amazon Aurora Sem Servidor v2
- Amazon Redshift
- Amazon Redshift sem servidor
- Amazon S3

- Amazon DynamoDB
- OpenSearch Serviço Amazon
- Amazon ElastiCache para Redis
- Amazon Kinesis Data Streams
- Amazon DocumentDB (compatível com MongoDB)
- Amazon Neptune
- Apache Kafka: Amazon Managed Streaming for Apache Kafka (Amazon MSK) e Apache Kafka autogerenciado
- Babelfish (versão 3.2.0 e superior) para Aurora PostgreSQL (versões 15.3/14.8 e superior)

Para obter informações sobre como trabalhar com um alvo específico, consulte Trabalhando com AWS DMS endpoints.

Para obter informações sobre endpoints de origem compatíveis, consulte Endpoints de origem da migração de dados.

Bancos de dados de destino do DMS Fleet Advisor

O DMS Fleet Advisor gera recomendações de destinos utilizando a versão mais recente dos seguintes bancos de dados de destino.

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

Para obter informações sobre as recomendações de destino no DMS Fleet Advisor, consulteUtilizar o recurso Recomendações de destino do AWS DMS Fleet Advisor.

Para obter informações sobre bancos de dados de origem compatíveis, consulte Bancos de dados de origem para o DMS Fleet Advisor.

Provedores de dados de destino para a DMS Schema Conversion

A DMS Schema Conversion é compatível com os seguintes provedores de dados como destinos para projetos de migração.

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

Para obter informações sobre como trabalhar com destinos específicos, consulte Criar provedores de dados de destino no DMS Schema Conversion.

Para obter informações sobre bancos de dados de origem compatíveis, consulte Provedores de dados de origem para a DMS Schema Conversion.

Provedores de dados de destino para migrações de dados homogêneas do DMS

É possível utilizar os seguintes provedores de dados como destinos para migrações de dados homogêneas.

- Amazon Aurora MySQL versão 5.7 e posterior
- Amazon Aurora PostgreSQL versões 10.4 a 14.x
- Amazon Aurora Sem Servidor v2
- Amazon RDS para MySQL versão 5.7 e posterior
- Amazon RDS para MariaDB versão 10.2 e posterior
- Amazon RDS para PostgreSQL versões 10.4 a 14.x
- Amazon DocumentDB versão 4.0, 5.0 e cluster Elástico DocumentDB

Para obter informações sobre como trabalhar com destinos específicos, consulte Criação de provedores de dados de destino para migrações de dados homogêneas no AWS DMS.

Para obter informações sobre bancos de dados de origem compatíveis, consulte Provedores de dados de origem para migrações de dados homogêneas do DMS.

Construindo um nome de recurso da Amazon (ARN) para AWS DMS

Se você usa a AWS DMS API AWS CLI ou para automatizar sua migração de banco de dados, você trabalha com o Amazon Resource Name (ARNs). Cada recurso criado na Amazon Web Services é identificado por um ARN, que é um identificador exclusivo. Se você usa a AWS DMS API AWS CLI ou para configurar a migração do banco de dados, você fornece o ARN do recurso com o qual deseja trabalhar.

Um ARN para um AWS DMS recurso usa a seguinte sintaxe:

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

Nesta sintaxe, o seguinte se aplica:

- *`region`*é o ID do Região da AWS local em que o AWS DMS recurso foi criado, como`us-west-2`.

  A tabela a seguir mostra Região da AWS os nomes e os valores que você deve usar ao criar um ARN.

Região	Nome
Região Ásia-Pacífico (Tóquio)	ap-northeast-1
Região da Ásia-Pacífico (Seul)	ap-northeast-2
Região da Ásia-Pacífico (Mumbai)	ap-south-1
Região Ásia-Pacífico (Singapura)	ap-southeast-1
Região Ásia-Pacífico (Sydney)	ap-southeast-2
Região Canadá (Central)	ca-central-1
Região China (Pequim)	cn-north-1
Região China (Ningxia)	cn-northwest-1
Região Europa (Estocolmo)	eu-north-1

Região	Nome
Região Europa (Milão)	eu-south-1
Região UE (Frankfurt)	eu-central-1
Região Europa (Irlanda)	eu-west-1
Região da UE (Londres)	eu-west-2
Região Europa (Paris)	eu-west-3
Região América do Sul (São Paulo)	sa-east-1
Região Leste dos EUA (N. da Virgínia)	us-east-1
Região Leste dos EUA (Ohio)	us-east-2
Região Oeste dos EUA (Norte da Califórnia)	us-west-1
Região Oeste dos EUA (Oregon)	us-west-2

- *account number* é o número da sua conta com os traços omitidos. Para encontrar o número da sua conta, entre na sua AWS conta em http://aws.amazon.com, escolha Minha conta/console e escolha Minha conta.
- *resourcetype*é o tipo de AWS DMS recurso.

  A tabela a seguir mostra os tipos de recursos a serem usados ao criar um ARN para um AWS DMS recurso específico.

AWS DMS tipo de recurso	Formato ARN
Instância da replicação	arn:aws:dms: *region*: *account*:rep: *resourcename*
Endpoint	arn:aws:dms: *region*:*account*:endpoint: *resourcename*
Tarefa de replicação	arn:aws:dms: *region*:*account*:task:*resourcename*

AWS DMS tipo de recurso	Formato ARN
Grupo de sub-redes	`arn:aws:dms:` *`region`*`:`*`account`*`:subgrp:`*`resourcename`*

- *`resourcename`*é o nome do recurso atribuído ao AWS DMS recurso. Esta string é gerada arbitrariamente.

A tabela a seguir mostra exemplos de ARNs para AWS DMS recursos. Aqui, pressupomos uma conta da AWS de 123456789012, criada na região Leste dos EUA (Norte da Virgínia) e que tem um nome de recurso.

Tipo de recurso	Amostra de ARN
Instância da replicação	`arn:aws:dms:us-east-1:123456789012:rep:QLXQZ64MH7CXF4QCQMGRVYVXAI`
Endpoint	`arn:aws:dms:us-east-1:123456789012:endpoint:D3HMZ2IGUCGFF3NTAXUXGF6S5A`
Tarefa de replicação	`arn:aws:dms:us-east-1:123456789012:task:2PVREMWNPGYJCVU2IBPTOYTIV4`
Grupo de sub-redes	`arn:aws:dms:us-east-1:123456789012:subgrp:test-tag-grp`

Usando AWS DMS com outros AWS serviços

Você pode usar AWS DMS com vários outros AWS serviços:

- Utilize uma instância do Amazon EC2 ou uma instância de banco de dados Amazon RDS como destino para uma migração de dados.
- Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter seu esquema de origem e código SQL em um esquema de destino e código SQL equivalentes.
- Utilize o Amazon S3 como um local de armazenamento para os dados ou utilize-o como uma etapa intermediária ao migrar grandes quantidades de dados.

- Você pode usar AWS CloudFormation para configurar seus AWS recursos para gerenciamento ou implantação da infraestrutura. Por exemplo, você pode provisionar AWS DMS recursos como instâncias de replicação, tarefas, certificados e endpoints. Você cria um modelo que descreve todos os AWS recursos que você deseja e AWS CloudFormation provisiona e configura esses recursos para você.

AWS DMS suporte para AWS CloudFormation

Você pode provisionar AWS DMS recursos usando AWS CloudFormation o. AWS CloudFormation é um serviço que ajuda você a modelar e configurar seus AWS recursos para gerenciamento ou implantação de infraestrutura. Por exemplo, você pode provisionar AWS DMS recursos como instâncias de replicação, tarefas, certificados e endpoints. Você cria um modelo que descreve todos os AWS recursos que você deseja, AWS CloudFormation provisiona e configura esses recursos para você.

Como desenvolvedor ou administrador do sistema, é possível criar e gerenciar coleções desses recursos para utilizar em tarefas de migração repetitivas ou para implantar recursos em sua organização. Para obter mais informações sobre AWS CloudFormation, consulte AWS CloudFormation os conceitos no Guia AWS CloudFormation do usuário.

AWS DMS suporta a criação dos seguintes AWS DMS recursos usando AWS CloudFormation:

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

Conceitos básicos do AWS Database Migration Service

No tutorial a seguir, é possível descobrir como executar uma migração de banco de dados com o AWS Database Migration Service (AWS DMS).

Para executar uma migração de banco de dados, execute as seguintes etapas:

1. Configure a conta da AWS seguindo as etapas em Configurando para AWS Database Migration Service.
2. Crie os bancos de dados de amostra e um cliente do Amazon EC2 para preencher o banco de dados de origem e testar a replicação. Além disso, crie uma nuvem privada virtual (VPC) com base no serviço Amazon Virtual Private Cloud (Amazon VPC) para conter os recursos do tutorial. Para criar esses recursos, siga as etapas em Pré-requisitos para AWS Database Migration Service.
3. Preencha o banco de dados de origem utilizando um exemplo de script de criação de banco de dados.
4. Utilize o DMS Schema Conversion ou a AWS Schema Conversion Tool (AWS SCT) para converter o esquema do banco de dados de origem no banco de dados de destino. Para utilizar a DMS Schema Conversion, siga as etapas em Conceitos básicos da DMS Schema Conversion. Para converter o esquema com a AWS SCT, siga as etapas em Esquema de migração.
5. Crie uma instância de replicação para executar todos os processos da migração. Para fazer isso e as tarefas a seguir, siga as etapas em Replicação.
6. Especifique os endpoints dos bancos de dados de origem e de destino. Para obter informações sobre como criar endpoints, consulte Criar endpoints de origem e de destino.
7. Crie uma tarefa para definir as tabelas e os processos de replicação que deseja utilizar e inicie a replicação. Para obter informações sobre como criar tarefas de migração de banco de dados, consulte Criar uma tarefa.
8. Verifique se a replicação está funcionando executando consultas no banco de dados de destino.

Configurando para AWS Database Migration Service

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções on-line.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Signing in as the root user (Fazer login como usuário-raiz) no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

Para obter instruções, consulte Habilitar AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Create a permission set no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Add groups no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos para AWS Database Migration Service

Nesta seção, você pode aprender as tarefas de pré-requisito para AWS DMS, como configurar seus bancos de dados de origem e destino. Como parte dessas tarefas, configure também uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC para conter os recursos. Além disso, configure uma instância do Amazon EC2 utilizada para preencher o banco de dados de origem e verificar a replicação no banco de dados de destino.

> **Note**
>
> O preenchimento do banco de dados de origem demora até 45 minutos.

Para este tutorial, crie um banco de dados MariaDB como a origem e um banco de dados PostgreSQL como o destino. Este cenário utiliza mecanismos de bancos de dados comuns e de baixo custo para demonstrar a replicação. O uso de mecanismos de banco de dados diferentes demonstra AWS DMS recursos para migrar dados entre plataformas heterogêneas.

Os recursos deste tutorial utilizam a região Oeste dos EUA (Oregon). Se você quiser usar uma AWS região diferente, especifique a região escolhida em vez de onde aparece Oeste dos EUA (Oregon).

> **Note**
>
> Para simplicidade, os bancos de dados criados para este tutorial não usam criptografia ou outros recursos avançados de segurança. Utilize recursos de segurança para manter os bancos de dados de produção seguros. Para obter mais informações, consulte Segurança no Amazon RDS.

Para obter as etapas de pré-requisito, consulte os tópicos a seguir.

Tópicos

- Crie uma VPC
- Criação de grupos de parâmetros do Amazon RDS
- Criação do banco de dados de origem do Amazon RDS
- Criação do banco de dados Amazon RDS de destino
- Criação de um cliente do Amazon EC2
- Preenchimento do banco de dados de origem

Crie uma VPC

Nesta seção, você cria uma VPC para conter seus AWS recursos. Usar uma VPC é uma prática recomendada ao usar AWS recursos, para que seus bancos de dados, instâncias do Amazon EC2, grupos de segurança e assim por diante estejam logicamente organizados e seguros.

A utilização de uma VPC para os recursos do tutorial também garante excluir todos os recursos utilizados ao concluir o tutorial. Exclua todos os recursos que uma VPC contém para poder excluí-la.

Para criar uma VPC para uso com AWS DMS

1. Faça login AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, escolha Painel da VPC e Criar VPC.
3. Na página Criar VPC, escolha as seguintes opções:

   - Recursos a serem criados: VPC e mais.
   - Geração automática da etiqueta de nome: escolha Gerar automaticamente e insira **DMSVPC**.
   - Bloco IPv4: **10.0.1.0/24**
   - Bloco CIDR IPv6: Nenhum bloco CIDR IPv6
   - Locação: Padrão
   - Número de zonas de disponibilidade: 2
   - Número de sub-redes públicas: 2
   - Número de sub-redes privadas: 2
   - Gateways NAT ($): Nenhum
   - Endpoints da VPC: Nenhum

   Escolha Criar VPC.
4. No painel de navegação, escolha Suas VPCs. Anote o ID da VPC da DMSVPC.
5. No painel de navegação, escolha Grupos de segurança.
6. Escolha o grupo chamado padrão que tem um ID de VPC que corresponda ao ID anotado para DMSVPC.
7. Vá para a guia Regras de entrada e escolha Editar regras de entrada.
8. Escolha Adicionar regra. Adicione uma regra do tipo MySQL/Aurora e escolha Anywhere-IPv4 para Origem.
9. Escolha Adicionar regra novamente. Adicione uma regra do tipo PostgreSQL e escolha Anywhere-IPv4 para Origem.
10. Escolha Salvar regras.

Criação de grupos de parâmetros do Amazon RDS

Para especificar configurações para seus bancos de dados de origem e destino AWS DMS, use grupos de parâmetros do Amazon RDS. Para permitir a replicação inicial e contínua entre os bancos de dados, configure o seguinte:

- O log binário do seu banco de dados de origem, para que ele AWS DMS possa determinar quais atualizações incrementais ele precisa replicar.
- A função de replicação do seu banco de dados de destino, de forma que AWS DMS ignore as restrições de chave estrangeira durante a transferência inicial de dados. Com essa configuração, é AWS DMS possível migrar dados fora de ordem.

Para criar grupos de parâmetros para uso com AWS DMS

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Grupos de parâmetros.
3. Na página Grupos de parâmetros, escolha Criar grupo de parâmetros.
4. Na página Criar grupo de parâmetros, insira as seguintes configurações:

   - Família de grupos de parâmetros: mariadb10.6
   - Nome do grupo: **`dms-mariadb-parameters`**
   - Descrição: **`Group for specifying binary log settings for replication`**

   Escolha Criar.
5. Na página Grupos de parâmetros, escolha dms-mariadb-parameters e, na página dms-mariadb-parameters, escolha Editar.
6. Defina os parâmetros a seguir como um dos seguintes valores:

   - binlog_checksum: NENHUM
   - binlog_format: LINHA

   Escolha Salvar alterações.
7. Na página Grupos de parâmetros, escolha Criar grupo de parâmetros novamente.
8. Na página Criar grupo de parâmetros, insira as seguintes configurações:

- Família de grupos de parâmetros: postgres13
- Nome do grupo: **`dms-postgresql-parameters`**
- Descrição: **`Group for specifying role setting for replication`**

Escolha Criar.

9. Na página Grupos de parâmetros, escolha dms-postgresql-parameters.
10. Na página dms-postgresql-parameters, escolha Editar e defina o parâmetro session_replication_role como réplica. Observe que o parâmetro session_replication_role não está na primeira página de parâmetros. Utilize os controles de paginação ou o campo de pesquisa para localizar o parâmetro.
11. Escolha Salvar alterações.

Criação do banco de dados de origem do Amazon RDS

Utilize o procedimento a seguir para criar o banco de dados Amazon RDS de origem.

Como criar o banco de dados Amazon RDS para MariaDB de origem

1. Abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. Na página Painel, escolha Criar banco de dados na seção Banco de dados. Não escolha Criar banco de dados na seção Tentar a nova opção de implantação multi-AZ do Amazon RDS para MySQL e PostgreSQL na parte superior da página.
3. Na página Criar banco de dados, defina as seguintes opções:

   - Escolher um método de criação de banco de dados: escolha Criação padrão.
   - Opções do mecanismo: em Tipo de mecanismo, escolha MariaDB. Em Versão, deixe MariaDB 10.6.14 selecionado.
   - Modelos: escolha Dev/teste.
   - Configurações:
     - Identificador da instância do banco de dados: insira **`dms-mariadb`**.
     - Na seção Configurações de credenciais, insira o seguinte:
       - Nome do usuário principal: deixe como**`admin`**.
       - Deixe a opção Gerenciar credenciais mestras no AWS Secrets Manager desmarcada.

    - Gerar automaticamente uma senha: deixe desmarcada.
    - Senha principal: insira **changeit**.
    - Confirmar senha: insira **changeit** novamente.
- Configuração da instância:
    - Classe da instância do banco de dados: deixe a opção Classes padrão escolhida.
    - Em Classe da instância do banco de dados, escolha db.m5.large.
- Armazenamento:
    - Desmarque a caixa Ativar escalabilidade automática do armazenamento.
    - Deixe o restante das configurações como estão.
- Disponibilidade e durabilidade: deixe a opção Não criar uma instância em espera selecionada.
- Conectividade:
    - Recurso de computação, deixe Não conectar-se a um recurso de computação do EC2
    - Tipo de rede: deixe IPv4 selecionado.
    - Nuvem privada virtual: DMSVPC-vpc
    - Acesso público: Sim. Ative o acesso público para utilizar a AWS Schema Conversion Tool.
    - Zona de disponibilidade: us-west-2a
    - Deixe o restante das configurações como estão.
- Autenticação do banco de dados: deixe Autenticação por senha selecionada.
- Em Monitoramento, desmarque a caixa Ativar o Performance Insights. Expanda a seção Configuração adicional e desmarque a caixa Ativar monitoramento avançado.
- Expanda Configuração adicional.
    - Em Opções de banco de dados, insira **dms_sample** como o Nome do banco de dados inicial.
    - Em Grupo de parâmetros do banco de dados, escolha dms-mariadb-parameters.
    - Em Grupo de opções, deixe default:mariadb-10-6 selecionado.
    - Em Backup, faça o seguinte:
        - Deixe a opção Ativar backups automáticos selecionada. O banco de dados de origem deve ter backups automáticos ativados para compatibilidade com a replicação contínua.
        - Em Período de retenção de backup, escolha Um dia.
        - Em Janela de backup, deixe a opção Sem preferência selecionada.
        - Desmarque a caixa Copiar tags para snapshots.

- Deixe a opção Ativar replicação em outra AWS região desmarcada.
- Em Criptografia, desmarque a caixa Ativar criptografia.
- Deixe a seção Exportações de logs como está.
- Em Manutenção, desmarque a caixa Ativar o upgrade automático da versão secundária e deixe a configuração Janela de manutenção como Sem preferência.
- Deixe a opção Ativar a proteção contra exclusão desmarcada.

4. Selecione Criar banco de dados.

Criação do banco de dados Amazon RDS de destino

Repita o procedimento anterior para criar o banco de dados Amazon RDS de destino, com as seguintes alterações.

Como criar o banco de dados RDS para PostgreSQL de destino

1. Repita as etapas de 1 e 2 do procedimento anterior.
2. Na página Criar banco de dados, defina as mesmas opções, exceto estas:
   a. Em Opções do mecanismo, escolha PostgreSQL.
   b. Em Versão, escolha PostgreSQL 13.7-R1
   c. Em Identificador de instância de banco de dados, insira **`dms-postgresql`**.
   d. Em Nome de usuário principal, deixe **`postgres`** selecionado.
   e. Na página Grupos de parâmetros do banco de dados, escolha dms-postgresql-parameters.
   f. Desmarque Ativar backups automáticos.
3. Selecione Criar banco de dados.

Criação de um cliente do Amazon EC2

Nesta seção, você cria um cliente do Amazon EC2. Utilize esse cliente para preencher o banco de dados de origem com dados para replicação. Também é possível utilizar esse cliente para verificar a replicação executando consultas no banco de dados de destino.

A utilização de um cliente do Amazon EC2 para acessar os bancos de dados fornece as seguintes vantagens em relação ao acesso aos bancos de dados pela internet:

- É possível restringir o acesso aos bancos de dados a clientes que estejam na mesma VPC.
- Confirmamos que as ferramentas utilizadas neste tutorial funcionam e são fáceis de instalar no Amazon Linux 2023, que é recomendável utilizar neste tutorial.
- As operações de dados entre componentes em uma VPC geralmente têm um desempenho melhor do que por meio da internet.

Como criar e configurar um cliente do Amazon EC2 para preencher o banco de dados de origem

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No Painel, escolha Iniciar instância.
3. Na página Iniciar uma instância, insira os seguintes valores:

   a. Na seção Nome e tags, insira **`DMSClient`** para Nome.

   b. Na seção Imagens de aplicações e de sistemas operacionais (Imagem de máquina da Amazon) deixe as configurações como estão.

   c. Na seção Tipo de instância, escolha t2.xlarge.

   d. Na seção Par de chaves (login), escolha Criar um novo par de chaves.

   e. Na página Criar par de chaves insira o seguinte:

      - Nome do par de chaves: **`DMSKeyPair`**
      - Tipo de par de chaves: deixe como RSA.
      - Formato de arquivo de chave privada: escolha pem para OpenSSH no MacOS ou Linux ou ppk para PuTTY no Windows.

      Salve o arquivo de chave quando solicitado.

      > **Note**
      >
      > Também é possível utilizar um par de chaves existente do Amazon EC2 em vez de criar um novo.

   f. Na seção Configurações de rede, escolha Editar. Escolha as seguintes configurações:

      - VPC obrigatória: escolha a VPC com o ID registrado para a VPC DMSVPC-vpc.
      - Sub-rede: escolha a primeira sub-rede pública.

- Atribuir IP público automaticamente: escolha Habilitar.

Deixe o restante das configurações como estão e escolha Iniciar instância.

Preenchimento do banco de dados de origem

Nesta seção, você encontra endpoints para os bancos de dados de origem e de destino para uso posterior, e utiliza as seguintes ferramentas para preencher o banco de dados de origem:

- Git, para baixar o script que preenche o banco de dados de origem.
- Cliente MariaDB, para executar esse script.

Obtenção de endpoints

Encontre e anote os endpoints de suas instâncias de banco de dados RDS para MariaDB e RDS para PostgreSQL para uso posterior.

Como encontrar os endpoints da instância do banco de dados

1. Faça login AWS Management Console e abra o console do Amazon RDS em https://console.aws.amazon.com/rds/.
2. No painel de navegação, escolha Bancos de dados.
3. Escolha o banco de dados dms-mariadb e anote o valor do Endpoint do banco de dados.
4. Repita as etapas anteriores para o banco de dados dms-postgresql.

Preenchimento do banco de dados de origem

Em seguida, conecte-se à sua instância cliente, instale o software necessário, baixe AWS exemplos de scripts de banco de dados do Git e execute os scripts para preencher seu banco de dados de origem.

Como preencher o banco de dados de origem

1. Conecte-se à instância do cliente utilizando o nome do host e a chave pública salva nas etapas anteriores.

Para obter mais informações sobre como se conectar a uma instância do Amazon EC2, consulte Como acessar instâncias no Guia do usuário do Amazon EC2.

> **Note**
>
> Se estiver utilizando o PuTTY, habilite os keepalives de TCP na página de configurações da Conexão para que a conexão não atinja o tempo limite devido à inatividade.

2. Instale o Git, o MariaDB e o PostgreSQL. Confirme a instalação conforme necessário.

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

3. Execute o comando a seguir para baixar os scripts de criação de banco de dados do GitHub.

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

4. Mude para o diretório `aws-database-migration-samples/mysql/sampledb/v1/`.
5. Execute o seguinte comando . Forneça o endpoint para a instância do RDS de origem anotada anteriormente, por exemplo `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. Deixe o script de criação do banco de dados ser executado. O script demora até 45 minutos para criar o esquema e preencher os dados. É possível ignorar com segurança os erros e avisos exibidos pelo script.

Migração do esquema de origem para o banco de dados de destino utilizando a AWS SCT

Nesta seção, você utiliza a AWS Schema Conversion Tool para migrar o esquema de origem para o banco de dados de destino. Como alternativa, é possível utilizar a DMS Schema Conversion para converter os esquemas do banco de dados de origem. Para obter mais informações, consulte Conceitos básicos da DMS Schema Conversion.

Como migrar o esquema de origem para o banco de dados de destino com a AWS SCT

1. Instale a AWS Schema Conversion Tool. Para obter mais informações, consulte Instalação, verificação e atualização da AWS SCT no AWSGuia do usuário da Schema Conversion Tool.

   Ao baixar drivers JDBC para o MySQL e o PostgreSQL, anote o local em que salva os drivers, caso a ferramenta solicite a localização deles.
2. Abra o AWS Schema Conversion Tool. Escolha Arquivo e Novo projeto.
3. Na janela Novo projeto, defina os seguintes valores:

   - Defina o Nome do projeto como **`DMSProject`**.
   - Mantenha a Localização como está para armazenar o projeto da AWS SCT na pasta padrão.

   Escolha OK.
4. Escolha Adicionar origem para adicionar um banco de dados MySQL de origem ao projeto e escolha MySQL e Avançar.
5. Na página Adicionar origem, defina os seguintes valores:

   - Nome da conexão: **`source`**
   - Nome do servidor: insira o endpoint do banco de dados MySQL anotado anteriormente.
   - Porta do servidor: **`3306`**
   - Nome do usuário: **`admin`**
   - Senha: **`changeit`**
6. Escolha Adicionar destino para adicionar um banco de dados de destino do Amazon RDS para PostgreSQL ao projeto e escolha Amazon RDS para PostgreSQL. Escolha Próximo.
7. Na página Adicionar origem, defina os seguintes valores:

   - Nome da conexão: **`target`**
   - Nome do servidor: insira o endpoint do banco de dados PostgreSQL anotado anteriormente.
   - Porta do servidor: **`5432`**
   - Banco de dados: insira o nome do banco de dados PostgreSQL.
   - Nome do usuário: **`postgres`**
   - Senha: **`changeit`**

8. No painel esquerdo, escolha dms_sample em Esquemas. No painel direito, escolha o banco de dados Amazon RDS para PostgreSQL de destino. Escolha Criar mapeamento. É possível adicionar várias aplicações a um único projeto da AWS SCT. Para obter mais informações sobre regras de mapeamento, consulte Criação de regras de mapeamento.
9. Escolha Visualização principal.
10. No painel esquerdo, escolha dms_sample em Esquemas. Abra o menu de contexto (clique com o botão direito do mouse) e selecione Converter esquema. Confirme a ação.

    Depois que a ferramenta converte o esquema, o esquema dms_sample aparece no painel direito.
11. No painel direito, em Esquemas, abra o menu de contexto (clique com o botão direito do mouse) de dms_sample e escolha Aplicar ao banco de dados. Confirme a ação.

Verifique se a migração do esquema foi concluída. Siga as etapas a seguir.

Para verificar a migração do esquema

1. Conecte-se ao cliente do Amazon EC2.
2. Inicie o cliente do PSQL com o comando a seguir. Especifique o endpoint do banco de dados PostgreSQL e forneça a senha do banco de dados quando solicitado.

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

3. Consulte uma das tabelas (vazias) para verificar se a AWS SCT aplicou o esquema corretamente.

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

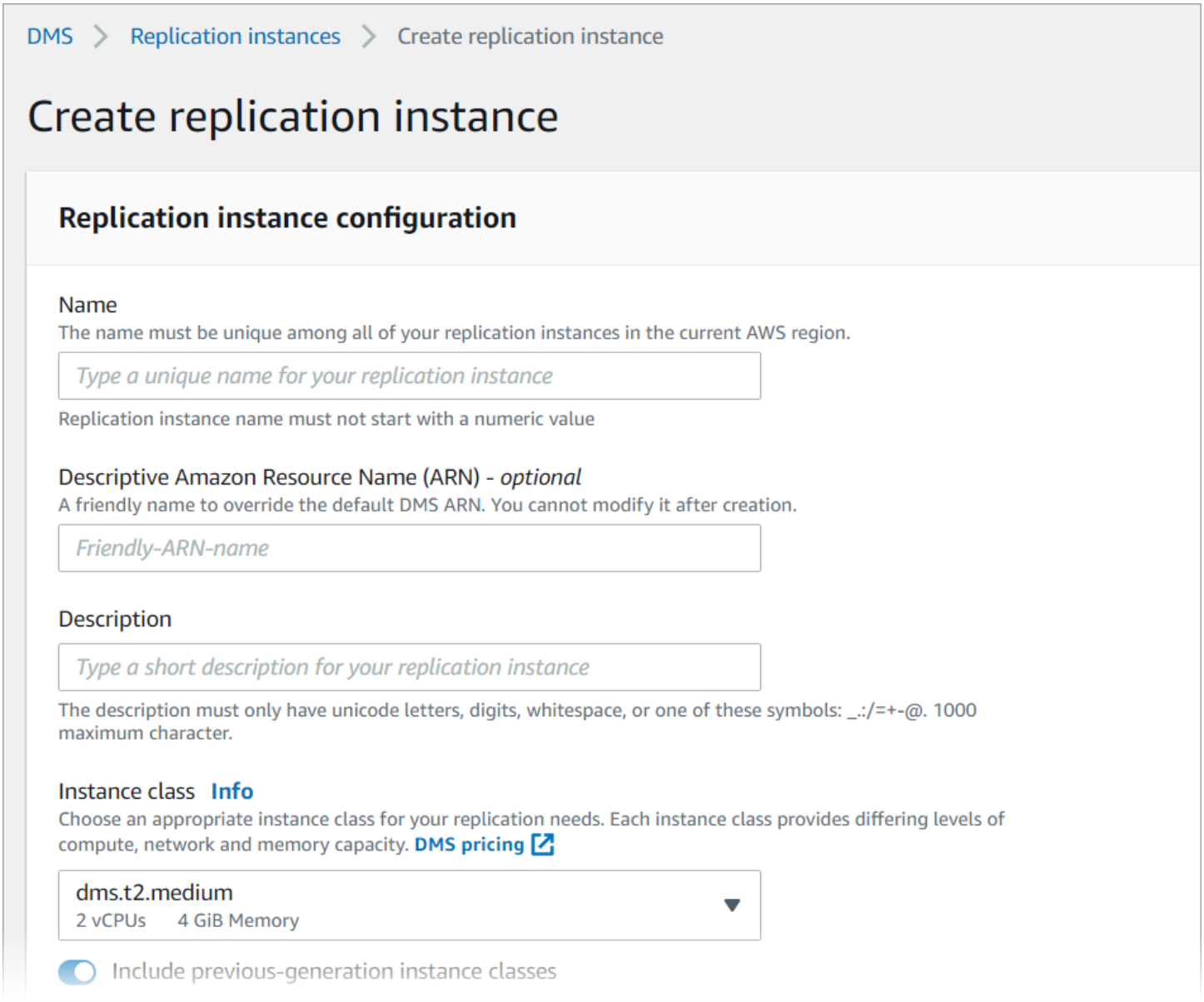
Configuração da replicação do AWS Database Migration Service

Neste tópico, você configura a replicação entre os bancos de dados de origem e de destino.

Etapa 1: Criar uma instância de replicação utilizando o console do AWS DMS

Para começar a trabalhar com o AWS DMS, crie uma instância de replicação.

Uma instância de replicação executa a migração real de dados entre os endpoints de origem e de destino. A instância precisa de armazenamento e capacidade de processamento suficientes para executar as tarefas que migram os dados do banco de dados de origem para o banco de dados de destino. O tamanho dessa instância de replicação depende da quantidade de dados a serem migrados e das tarefas que a instância precisa executar. Para obter mais informações sobre as instâncias de replicação, consulte Trabalhando com uma instância de AWS DMS replicação.



Como criar uma instância de replicação utilizando o console

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, selecione Instâncias de replicação e escolha Criar instância de replicação.
3. Na página Criar instância de replicação, especifique a configuração da instância de replicação:
   a. Em Nome, insira **`DMS-instance`**.
   b. Em Descrição, insira uma breve descrição para a instância de replicação (opcional).

c. Em Classe de instância, deixe dms.t3.medium escolhida.

   A instância precisa de armazenamento, de rede e de capacidade de processamento suficientes para a migração. Para obter mais informações sobre como escolher uma classe de instância, consulte Escolhendo a instância de replicação AWS DMS certa para sua migração.

d. Em Versão do mecanismo, aceite o padrão.

e. Em Multi AZ, escolha Workload de dev ou de teste (Single-AZ).

f. Em Armazenamento alocado (GiB), aceite o padrão de 50 GiB.

   No AWS DMS, o armazenamento é usado principalmente por arquivos de log e transações armazenadas em cache. Para transações armazenadas em cache, o armazenamento só é usado quando as transações devem ser gravadas em disco. Portanto, o AWS DMS não utiliza uma quantidade significativa de armazenamento.

g. Em Tipo de rede, escolha IPv4.

h. Em VPC, escolha DMSVPC.

i. Em Grupo de sub-redes de replicação, deixe o grupo de sub-redes de replicação escolhido atualmente.

j. Desmarque Publicamente acessível.

4. Escolha a guia Segurança avançada e configuração de rede para definir os valores das configurações de rede e de criptografia, se necessário:

   a. Em Zona de disponibilidade, escolha us-west-2a.

   b. Em Grupos de segurança da VPC, escolha o grupo de segurança Padrão, se ainda não estiver escolhido.

   c. Em AWS KMS key, deixe (Padrão) aws/dms escolhido.

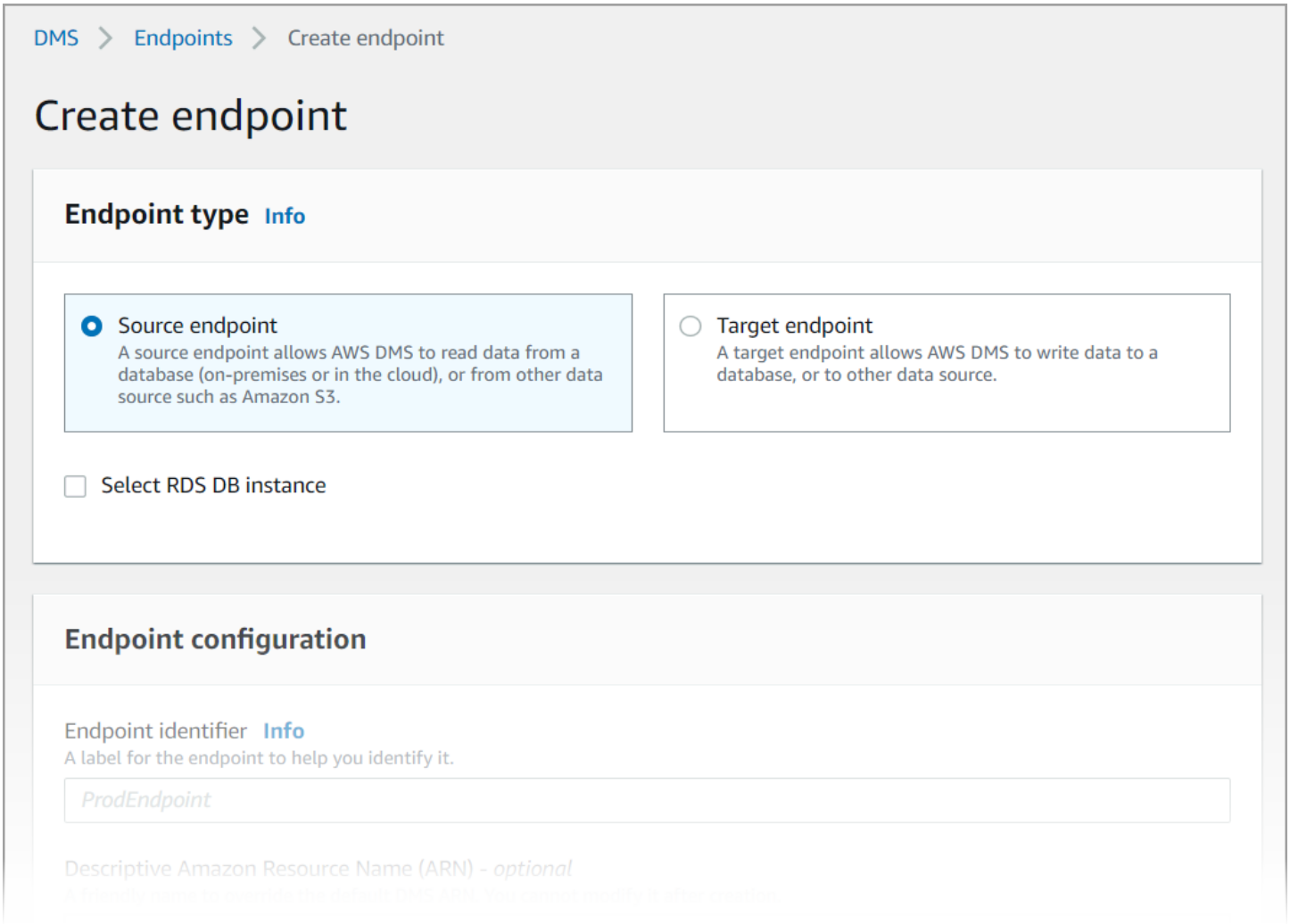
5. Deixe as configurações na guia Manutenção como estão. O padrão é uma janela de 30 minutos selecionada aleatoriamente em um bloco de tempo de 8 horas para cada região da AWS, ocorrendo em um dia da semana aleatório.

6. Escolha Criar.

O AWS DMS cria uma instância de replicação para executar a migração.

Etapa 2: Especificar endpoints de origem e de destino

Enquanto a instância de replicação estiver sendo criada, é possível especificar os endpoints dos datastore de destino para os bancos de dados Amazon RDS criados anteriormente. Crie cada endpoint separadamente.



Como especificar endpoints de origem e de banco de dados utilizando o console do AWS DMS

1. No console, escolha Endpoints no painel de navegação e escolha Criar endpoint.
2. Na página Criar endpoint, escolha o tipo de endpoint de Origem. Escolha caixa Selecionar instância de banco de dados RDS e escolha a instância dms-mariadb.
3. Na seção Configuração de endpoint, insira **`dms-mysql-source`** para o Identificador do endpoint.
4. Em Mecanismo de origem, deixe o MySQL escolhido.

5. Em Acesso ao banco de dados de endpoint, escolha Fornecer informações de acesso manualmente. Verifique se a Porta, o Modo SSL (Secure Socket Layer), o Nome do usuário e a Senha estão corretos.
6. Escolha a guia Testar a conexão do endpoint (opcional) Em VPC, escolha DMSVPC.
7. Em Instância de replicação, deixe dms-instance escolhida.
8. Escolha Executar teste.

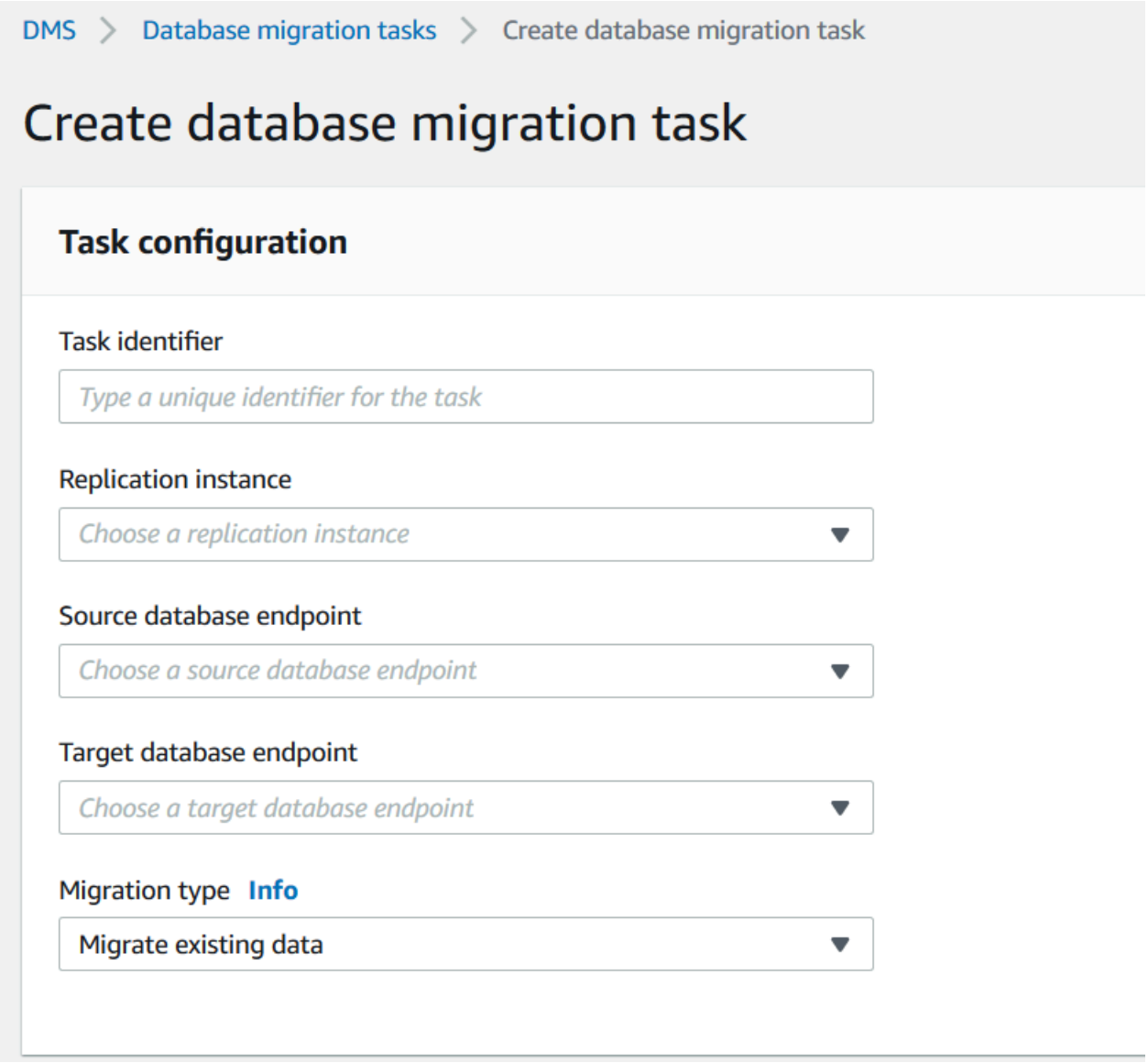
   Depois de escolher Executar teste, o AWS DMS cria o endpoint com os detalhes fornecidos e conecta-se a ele. Se a conexão falhar, edite a definição do endpoint e teste a conexão novamente. Também é possível excluir o endpoint manualmente.
9. Depois que o teste for bem-sucedido, escolha Criar endpoint.
10. Especifique um endpoint do banco de dados de destino utilizando o console do AWS DMS. Para isso, repita as etapas anteriores, com as seguintes configurações:

    - Tipo de endpoint: Endpoint de destino.
    - Instância do RDS: dms-postgresql
    - Identificador do endpoint: **`dms-postgresql-target`**
    - Mecanismo de destino: deixe **`PostgreSQL`** escolhido.

Depois que você concluir o fornecimento de todas as informações para os endpoints, o AWS DMS cria os endpoints de origem e de destino para uso durante a migração do banco de dados.

Etapa 3: Criar uma tarefa e migrar os dados

Nesta etapa, você cria uma tarefa para migrar os dados entre os bancos de dados criados.



Como criar uma tarefa de migração e iniciar a migração do banco de dados

1. No painel de navegação do console, escolha Tarefas de migração de banco de dados e escolha Criar tarefa. A página Criar tarefa de migração de banco de dados é aberta.
2. Na seção Configuração da tarefa, especifique as seguintes opções de tarefa:
   - Identificador da tarefa: insira **`dms-task`**.
   - Instância de replicação: escolha a instância de replicação (dms-instance-vpc-***<vpc id>***).
   - Endpoint do banco de dados de origem: escolha dms-mysql-source.
   - Endpoint do banco de dados de destino: escolha dms-postgresql-target.
   - Tipo de migração: escolha Migrar dados existentes e replicar as alterações em andamento.

3. Escolha a guia Configurações da tarefa. Defina as seguintes configurações:
   - Modo de preparação da tabela de destino: Não executar nenhuma ação
   - Interromper a tarefa após a conclusão da carga máxima: Não interromper
4. Escolha a guia Mapeamentos de tabela e expanda a guia Regras de seleção. Escolha Adicionar nova regra de seleção. Defina as seguintes configurações:
   - Esquema: Insira um esquema
   - Nome do esquema: **`dms_sample`**
5. Escolha a guia Configuração de startup da tarefa de migração e escolha Automaticamente na criação.
6. Escolha Criar tarefa.

O AWS DMS criará a tarefa de migração e a iniciará. A replicação inicial do banco de dados demora cerca de 10 minutos. Execute a próxima etapa do tutorial antes que o AWS DMS conclua a migração dos dados.

Etapa 4: Testar a replicação

Nesta seção, você insere os dados no banco de dados de origem durante e depois da replicação inicial e consulta os dados inseridos no banco de dados de destino.

Como testar a replicação

1. Verifique se a tarefa de migração de banco de dados mostra o status Em execução, mas se a replicação inicial do banco de dados, iniciada na etapa anterior, não está concluída.
2. Conecte-se ao cliente Amazon EC2 e inicie o cliente MySQL com o comando a seguir. Forneça o endpoint de banco de dados MySQL.

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

3. Execute o comando a seguir para inserir um registro no banco de dados de origem.

```
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. Saia do cliente MySQL.

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

5. Antes da conclusão da replicação, consulte o banco de dados de destino para obter o novo registro.

   Na instância do Amazon EC2, conecte-se ao banco de dados de destino utilizando o comando a seguir, fornecendo o endpoint do banco de dados de destino.

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

   Quando solicitado, forneça a senha (**changeit**).

6. Antes da conclusão da replicação, consulte o banco de dados de destino para obter o novo registro.

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

7. Enquanto a tarefa de migração estiver em execução, é possível monitorar o andamento da migração do banco de dados à medida que acontece:

   - No painel de navegação do console do DMS, escolha Tarefas de migração do banco de dados.
   - Escolha dms-task.
   - Escolha Estatísticas da tabela.

   Para obter mais informações sobre o monitoramento, consulte Monitoramento de tarefas do AWS DMS.

8. Após a conclusão da replicação, consulte o banco de dados de destino novamente para obter o novo registro. O AWS DMS migra o novo registro após a conclusão da replicação inicial.

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

9. Saia do cliente psql.

```
dms_sample=> quit
```

10. Repita a etapa 1 para conectar-se novamente ao banco de dados de origem.
11. Insira outro registro na tabela `person`.

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

12. Repita as etapas 3 e 4 para desconectar-se do banco de dados de destino.
13. Consulte o banco de dados de destino novamente para obter os dados replicados.

```
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)
```

Etapa 5: Limpar os recursos do AWS DMS

Depois de concluir o tutorial de conceitos básicos, é possível excluir os recursos criados. É possível utilizar o console da AWS para removê-los. Exclua as tarefas de migração antes de excluir a instância de replicação e os endpoints.

Como excluir uma tarefa de migração utilizando o console

1. No painel de navegação do console do AWS DMS, escolha Tarefas de migração do banco de dados.
2. Escolha dms-task.

3. Escolha Ações, Excluir.

Como excluir uma instância de replicação utilizando o console

1. No painel de navegação do console do AWS DMS, escolha Instâncias de replicação.
2. Escolha DMS-instance.
3. Escolha Ações, Excluir.

O AWS DMS exclui a instância de replicação e a remove da página Instâncias de replicação.

Como remover os endpoints utilizando o console

1. No painel de navegação do console do AWS DMS, escolha Endpoints.
2. Escolha dms-mysql-source.
3. Escolha Ações, Excluir.

Depois de excluir os recursos do AWS DMS, exclua também os seguintes recursos. Para obter ajuda com a exclusão de recursos em outros serviços, consulte a documentação de cada serviço.

- Os bancos de dados RDS.
- Os grupos de parâmetros do banco de dados RDS.
- Os grupos de sub-redes do RDS.
- Qualquer log do Amazon CloudWatch que tenha sido criado junto com os bancos de dados e instância de replicação.
- Os grupos de segurança criados para a Amazon VPC e o cliente do Amazon EC2. Remova a regra de entrada de Padrão para os grupos de segurança launch-wizard-1, o que é necessário para que seja possível excluí-los.
- O cliente Amazon EC2.
- A Amazon VPC.
- O par de chaves do Amazon EC2 do cliente do Amazon EC2.

Recursos adicionais para trabalhar com o AWS Database Migration Service.

Mais adiante neste guia, você saberá como utilizar o AWS DMS para migrar os dados para os bancos de dados e vice-versa de código aberto mais usados comercialmente.

Também é recomendável verificar os seguintes recursos ao preparar e executar um projeto de migração de banco de dados:

- Guia de migração passo a passo do AWS DMS: este guia fornece orientações passo a passo sobre o processo de migração de dados do AWS.
- Referência da API do AWS DMS: descreve em detalhes todas as operações da API do AWS Database Migration Service.
- AWS CLI do AWS DMS: esta referência fornece informações sobre como utilizar a AWS Command Line Interface (AWS CLI) com o AWS DMS.

Descoberta e avaliação de bancos de dados para migração como o AWS DMS Fleet Advisor

É possível utilizar o DMS Fleet Advisor para coletar metadados e métricas de desempenho de vários ambientes de banco de dados. Essas métricas coletadas fornecem insights da infraestrutura de dados. O DMS Fleet Advisor coleta metadados e métricas do banco de dados on-premises e de servidores de análise de um ou mais locais centrais sem a necessidade de instalá-los em todos os computadores. Atualmente, o DMS Fleet Advisor oferece suporte à descoberta e à coleta de métricas para servidores de banco de dados Microsoft SQL Server, MySQL, Oracle e PostgreSQL.

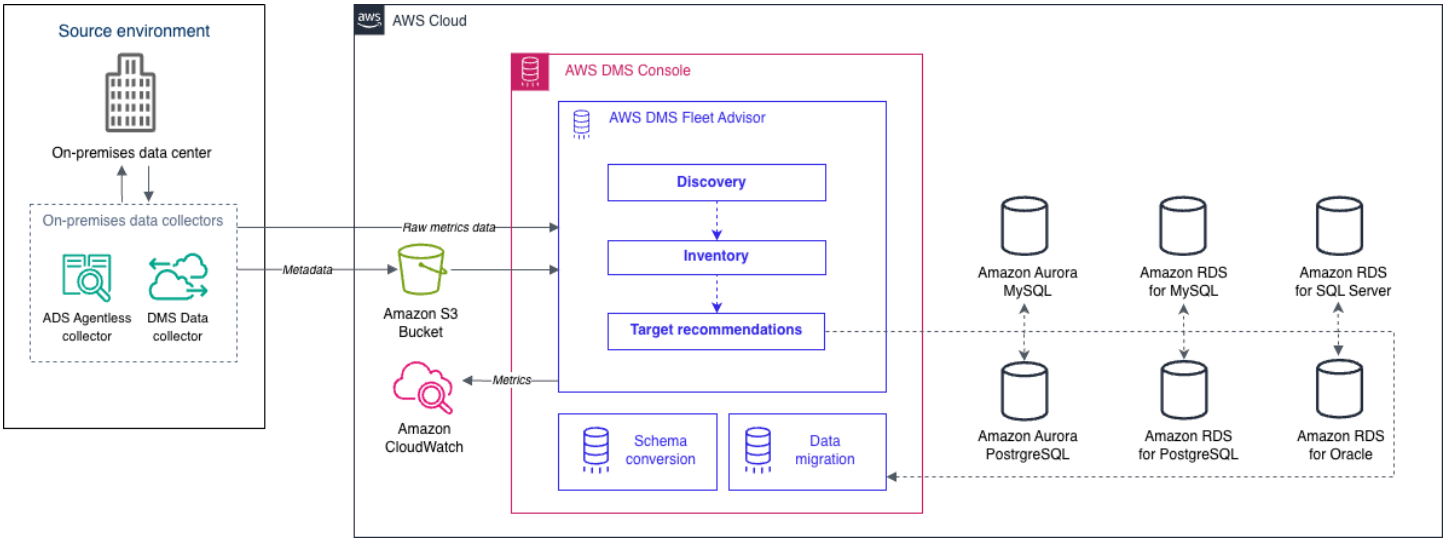
Com base nos dados descobertos na rede, é possível criar um inventário para definir a lista de servidores de banco de dados para coleta de dados adicional. Depois de AWS DMS coletar informações sobre servidores, bancos de dados e esquemas, é possível analisar a viabilidade das migrações de banco de dados pretendidas.

Para os bancos de dados no inventário que você planeja migrar para a Nuvem AWS, o DMS Fleet Advisor gera recomendações de tamanho certo para o destino. Para gerar recomendações de destino, o DMS Fleet Advisor considera as métricas do coletor de dados e as configurações preferidas. Depois que o DMS Fleet Advisor gera recomendações, é possível visualizar informações detalhadas de cada configuração do banco de dados de destino. Os engenheiros e administradores de banco de dados da organização podem utilizar as recomendações de destino do DMS Fleet Advisor para planejar a migração dos bancos de dados on-premises para a AWS. Você pode explorar as diferentes opções de migração disponíveis e exportar essas recomendações AWS Pricing Calculator para otimizar ainda mais o custo.

Para obter a lista dos bancos de dados de origem compatíveis, consulte Origens do DMS Fleet Advisor.

Para obter a lista dos bancos de dados que o DMS Fleet Advisor utiliza para gerar recomendações de destino, consulte Destinos do DMS Fleet Advisor. O DMS Fleet Advisor gera recomendações semelhantes a semelhantes, por exemplo, do Oracle de origem para o banco de dados Oracle de destino. O DMS Fleet Advisor também gera recomendações heterogêneas, como a migração do Oracle ou Microsoft SQL Server de origem para o banco de dados RDS for PostgreSQL ou Aurora PostgreSQL de destino.

O diagrama a seguir ilustra o processo de recomendações de destino do AWS DMS Fleet Advisor.



Utilize os seguintes tópicos para compreender melhor o AWS DMS Fleet Advisor.

Tópicos

- Com suporte Regiões da AWS
- Conceitos básicos do DMS Fleet Advisor
- Configurar o AWS DMS Fleet Advisor
- Descobrir bancos de dados para a migração utilizando coletores de dados
- Utilizar inventários para análise no AWS DMS Fleet Advisor
- Utilizar o recurso Recomendações de destino do AWS DMS Fleet Advisor
- Limitações do DMS Fleet Advisor

Com suporte Regiões da AWS

É possível utilizar o DMS Fleet Advisor nas seguintes Regiões da AWS.

Nome da região	Região
Leste dos EUA (Norte da Virgínia)	us-east-1
Leste dos EUA (Ohio)	us-east-2

Nome da região	Região
Oeste dos EUA (Norte da Califórnia)	us-west-1
Oeste dos EUA (Oregon)	us-west-2
Ásia-Pacífico (Hong Kong)	ap-east-1
Ásia-Pacífico (Tóquio)	ap-northeast-1
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia-Pacífico (Osaka)	ap-northeast-3
Ásia-Pacífico (Mumbai)	ap-south-1
Ásia-Pacífico (Cingapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2
Ásia-Pacífico (Jacarta)	ap-southeast-3
Canadá (Central)	ca-central-1
Europa (Frankfurt)	eu-central-1
Europa (Estocolmo)	eu-north-1
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (Paris)	eu-west-3
Europa (Milão)	eu-south-3
Canadá (Central)	ca-central-1
América do Sul (São Paulo)	sa-east-1

Nome da região	Região
Oriente Médio (Barém)	me-south-1
África (Cidade do Cabo)	af-south-1

Conceitos básicos do DMS Fleet Advisor

É possível utilizar o DMS Fleet Advisor para descobrir os bancos de dados on-premises de origem para a migração para a Nuvem AWS. É possível determinar o destino de migração correto na Nuvem AWS para cada um dos bancos de dados on-premises. Utilize o fluxo de trabalho a seguir para criar um inventário de bancos de dados de origem e gerar recomendações de destino.

1. Crie um bucket do Amazon S3 e políticas, perfis e usuários do IAM. Para obter mais informações, consulte Criar os recursos necessários.
2. Crie os usuários do banco de dados com as permissões mínimas necessárias para o coletor de dados do DMS. Para obter mais informações, consulte Criar usuários do banco de dados.
3. Crie e baixe um coletor de dados. Para obter mais informações, consulte Criar um coletor de dados.
4. Instale o coletor de dados no ambiente local. Configure o coletor de dados para garantir que ele possa enviar os dados coletados para o DMS Fleet Advisor. Para obter mais informações, consulte Instalar um coletor de dados.
5. Descubra o SO e os servidores de banco de dados no ambiente de dados. Para obter mais informações, consulte Descobrir sistemas operacionais e servidores de banco de dados.
6. Colete metadados do banco de dados e as métricas de utilização de recursos. Para obter mais informações, consulte Coletar dados.
7. Analise os bancos de dados e os esquemas de origem. O DMS Fleet Advisor executa a avaliação em grande escala dos bancos de dados para identificar esquemas semelhantes. Para obter mais informações, consulte Utilizar inventários para análise no AWS DMS Fleet Advisor.
8. Gere, visualize e salve uma cópia local das recomendações de destino para os bancos de dados de origem. Para obter mais informações, consulte Recomendações de destino.

Depois de determinar o destino da migração para cada banco de dados de origem, é possível usar a DMS Schema Conversion para converter os esquemas de banco de dados em uma nova

plataforma. Utilize o AWS DMS para migrar os dados. Para obter mais informações, consulte Converter esquemas de banco de dados utilizando a DMS Schema Conversion e O que é o AWS Database Migration Service?

Este vídeo apresenta a interface de usuário da DMS Schema Conversion e ajuda você a se familiarizar com os principais componentes desse serviço.

Configurar o AWS DMS Fleet Advisor

Para configurar o AWS DMS Fleet Advisor, execute as seguintes tarefas de pré-requisito.

Tópicos

- Criar os recursos da AWS necessários para o AWS DMS Fleet Advisor
- Criar usuários do banco de dados para o AWS DMS Fleet Advisor

Criar os recursos da AWS necessários para o AWS DMS Fleet Advisor

O DMS Fleet Advisor precisa de um conjunto de recursos da AWS na sua conta para encaminhar e importar as informações do inventário e atualizar o status do coletor de dados do DMS.

Antes de coletar dados e criar inventários de bancos de dados e esquemas pela primeira vez, execute os seguintes pré-requisitos.

Para configurar o bucket do Amazon S3 e os recursos do IAM, realize uma das seguintes ações:

- Configure os recursos do Amazon S3 e do IAM usando o AWS CloudFormation. (recomendado).
- Configurar os recursos do Amazon S3 e do IAM no AWS Management Console

Configure os recursos do Amazon S3 e do IAM usando o AWS CloudFormation.

Uma CloudFormation pilha é um conjunto de recursos da AWS que você pode gerenciar como uma unidade. Para simplificar a criação dos recursos necessários para o DMS Fleet Advisor, você pode usar os arquivos AWS CloudFormation de modelo para criar CloudFormation pilhas. Para ter mais informações, consulte Criar uma pilha no console do AWS CloudFormation no Guia do usuário do AWS CloudFormation.

> **Note**
>
> Esta seção só se aplica ao uso do coletor autônomo do DMS Fleet Advisor. Para obter informações sobre o uso de um único coletor on-premises para coletar informações sobre bancos de dados e servidores, consulte Application Discovery Service Agentless Collector no Guia do usuário do AWS Application Discovery Service.

Recursos do Amazon S3 e IAM criados por CloudFormation

Quando você usa os CloudFormation modelos, eles criam pilhas que incluem os seguintes recursos em seuConta da AWS:

- Um bucket do Amazon S3 denominado `dms-fleetadvisor-data-accountId-region`
- Um usuário do IAM denominado `FleetAdvisorCollectorUser-region`
- Um perfil de serviço do IAM denominado `FleetAdvisorS3Role-region`
- Uma política de acesso denominada `FleetAdvisorS3Role-region-Policy`
- Uma política de acesso denominada `FleetAdvisorCollectorUser-region-Policy`
- Um perfil vinculado ao serviço (SLR) do IAM denominado `AWSServiceRoleForDMSFleetAdvisor`

Siga as etapas listadas abaixo para configurar seus recursos com CloudFormation.

- Etapa 1: baixar os arquivos CloudFormation de modelo
- Etapa 2: Configurar o Amazon S3 e o IAM usando CloudFormation

Etapa 1: baixar os arquivos CloudFormation de modelo

Um CloudFormation modelo é uma declaração dos AWS recursos que compõem uma pilha. O modelo é armazenado como um arquivo JSON.

Para baixar os arquivos CloudFormation de modelo

1. Abra o menu de contexto (clique com o botão direito do mouse) para um dos seguintes links e escolha Salvar link como.

- Se você planeja usar o DMS Fleet Advisor, escolha dms-fleetadvisor-iam-slr-s3.zip. Se você já criou a SLR para o DMS Fleet Advisor, escolha 3.zip dms-fleetadvisor-iam-s
- Se você planeja usar o Coletor Sem Agente do AWS Application Discovery Service (ADS) e ainda não criou a SLR para ele, escolha -slr-s3.zip. dms-fleetadvisor-ads-iam Se você já criou a SLR para o DMS Fleet Advisor com ADS, escolha dms-fleetadvisor-ads-iam -s3.zip.

2. Salve o arquivo no computador.

Etapa 2: Configurar o Amazon S3 e o IAM usando CloudFormation

Quando você usa o CloudFormation modelo para o IAM, ele cria os recursos do Amazon S3 e do IAM listados anteriormente.

Para configurar o Amazon S3 e o IAM usando CloudFormation

1. Abra o CloudFormation console em https://console.aws.amazon.com/cloudformation.
2. Inicie o assistente de criação de pilha escolhendo Criar pilha e Com novos recursos na lista suspensa.
3. Na página Create a stack (Criar uma pilha), faça o seguinte:
   a. Em Prepare template (Preparar modelo), selecione Template is ready (O modelo está pronto).
   b. Para Template source (Origem do template), escolha Upload a template file (Fazer upload de um arquivo de template).
   c. Em Escolher arquivo, navegue até -s3.json, -s3.json e, em seguida, escolha dms-fleetadvisor-iam-slr-s3.json. dms-fleetadvisor-iam , dms-fleetadvisor-ads-iam-slr-s3.zip ou dms-fleetadvisor-ads-iam-s3.zip.
   d. Escolha Próximo.
4. Na página Specify stack details (Especificar detalhes da pilha), faça o seguinte:
   a. Em Nome da pilha, insira **`dms-fleetadvisor-iam-slr-s3`**,**`dms-fleetadvisor-iam-s3`**, **`dms-fleetadvisor-ads-iam-slr-s3`** ou **`dms-fleetadvisor-ads-iam-s3`**.
   b. Escolha Avançar.
5. Na página Configurar opções de pilha, selecione Avançar.
6. Na página Revisão dms-fleetadvisor-iam-slr -s3, Revisão dms-fleetadvisor-iam-s 3, Revisão dms-fleetadvisor-ads-iam -slr-s3 ou Revisão dms-fleetadvisor-ads-iam -s3, faça o seguinte:

a. Marque a caixa de seleção Confirmo que o AWS CloudFormation pode criar recursos do IAM com nomes personalizados.

b. Selecione Enviar.

CloudFormation cria o bucket do S3 e as funções e o usuário do IAM que o DMS Fleet Advisor exige. No painel esquerdo, quando dms-fleetadvisor-iam-slr-s3, 3, dms-fleetadvisor-iam-sdms-fleetadvisor-ads-iam-slr-s3 ou dms-fleetadvisor-ads-iam-s3 mostrar CREATE_COMPLETE, vá para a próxima etapa.

7. No painel esquerdo, escolha dms-fleetadvisor-iam-slr-s3, 3, dms-fleetadvisor-iam-sdms-fleetadvisor-ads-iam-slr-s3 ou -s3. dms-fleetadvisor-ads-iam No painel à direita, faça o seguinte:

   a. Selecione Informações da pilha. ***Sua pilha tem um ID no formato arn:aws:cloudformation: region: account-no:stack/ -s3/ identifier, arn:aws:cloudformation: region: account-no:stack/ 3/ identifier, arn:aws:cloudformation: region: account-no:stack/ -slr-s3/ dms-fleetadvisor-iam-slr identifier ou arn:aws:cloudformation: region: account-account-account-account não: stack/ -s3/ identificador dms-fleetadvisor-iam-s dms-fleetadvisor-ads-iam dms-fleetadvisor-ads-iam*** .

   b. Escolha atributos. Você deve ver o seguinte:

   - Um bucket do Amazon S3 denominado `dms-fleetadvisor-data-`*`accountId`*`-`*`region`*
   - Um perfil de serviço denominado `FleetAdvisorS3Role-`*`region`*
   - Um usuário do IAM denominado `FleetAdvisorCollectorUser-`*`region`*
   - Um SLR do IAM denominado `AWSServiceRoleForDMSFleetAdvisor` (se você tiver baixado `dms-fleet-advisor-iam-slr-s3.zip` oudms-fleet-advisor-ads-iam-slr-s3.zip)
   - Uma política de acesso denominada `FleetAdvisorS3Role-`*`region`*`-Policy`
   - Uma política de acesso denominada `FleetAdvisorCollectorUser-`*`region`*`-Policy`

Configurar os recursos do Amazon S3 e do IAM no AWS Management Console

Criar um bucket do Amazon S3

Crie um bucket do Amazon S3 no qual os metadados do inventário podem ser armazenados. É recomendável pré-configurar esse bucket do S3 antes de utilizar o DMS Fleet Advisor. O AWS DMS armazena os metadados do inventário do DMS Fleet Advisor nesse bucket do S3.

Para obter mais informações sobre como criar um bucket do S3, consulte Criar seu primeiro bucket do S3, no Guia do usuário do Amazon S3.

> **Note**
>
> O DMS Fleet Advisor só oferece suporte a buckets criptografados SSE-S3.

Como criar um bucket do Amazon S3 para armazenar informações do ambiente de dados local

1. Faça login no AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Escolha Criar bucket.
3. Na página Criar bucket, insira um nome globalmente exclusivo que inclua seu nome de login para o bucket, como fa-bucket-***seulogin***.
4. Selecione a Região da AWS em que você utiliza o DMS Fleet Advisor.
5. Mantenha as configurações restantes e escolha Criar bucket.

Criar recursos do IAM

Nesta seção, você cria recursos do IAM para o coletor de dados, o usuário do IAM e o DMS Fleet Advisor.

Tópicos

- Criar recursos do IAM para o coletor de dados
- Criar o perfil vinculado a serviço do DMS Fleet Advisor

Criar recursos do IAM para o coletor de dados

Para garantir que o coletor de dados funcione corretamente e faça upload dos metadados coletados no bucket do Amazon S3, crie as seguintes políticas. Crie um usuário do IAM com as permissões mínimas a seguir. Para obter mais informações sobre o coletor de dados do DMS, consulte Descobrir bancos de dados para a migração utilizando coletores de dados.

Como criar uma política do IAM para o DMS Fleet Advisor e o coletor de dados acessarem o Amazon S3

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas.
3. Escolha Create policy.
4. Na página Criar política, escolha a guia JSON.
5. Cole o seguinte JSON no editor, substituindo o código de exemplo. Substitua *fa_bucket* pelo nome do bucket do Amazon S3 criado na seção anterior.

```
{
    "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. Selecione Next: Tags (Próximo: tags) e Next: Review (Próximo: revisar).
7. Insira **`FleetAdvisorS3Policy`** em Nome* e escolha Criar política.

Como criar uma política do IAM para o coletor de dados do DMS para acessar o DMS Fleet Advisor

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas.
3. Escolha Create policy.
4. Na página Criar política, escolha a guia JSON.
5. Cole o seguinte código JSON no editor, substituindo o código de exemplo.

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

6. Selecione Next: Tags (Próximo: tags) e Next: Review (Próximo: revisar).
7. Insira **`DMSCollectorPolicy`** em Nome* e escolha Criar política.

Como criar um usuário do IAM com permissões mínimas para utilizar o coletor de dados do DMS

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Usuários.
3. Escolha Adicionar usuários.
4. Na página Adicionar usuário, insira **`FleetAdvisorCollectorUser`** em Nome do usuário*. Escolha Chave de acesso: acesso programático em Selecionar tipo de acesso da AWS. Escolha Próximo: permissões.
5. Na seção Definir permissões, escolha Anexar políticas existentes diretamente.

6. Use o controle de pesquisa para encontrar e escolher as políticas DMS CollectorPolicy e FleetAdvisorS3Policy que você criou antes. Escolha Próximo: etiquetas.
7. Na página Tags (Etiquetas), escolha Next: Review (Avançar: revisar).
8. Na página Review (Revisar), selecione Create user (Criar usuário). Na próxima página, escolha Baixar .csv para salvar as novas credenciais do usuário. Utilize essas credenciais com o DMS Fleet Advisor para obter as permissões de acesso mínimas necessárias.

Como criar um perfil do IAM para o DMS Fleet Advisor e o coletor de dados acessarem o Amazon S3

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles.
3. Escolha Criar Perfil.
4. Na página Selecionar entidade confiável, em Tipo de entidade confiável, escolha Serviço da AWS. Em Casos de uso para outros serviços da AWS, escolha DMS.
5. Marque a caixa de seleção DMS e escolha Próximo.
6. Na página Adicionar permissões, escolha FleetAdvisorS3Policy. Escolha Próximo.
7. Na página Nomear, revisar e criar, insira **`FleetAdvisorS3Role`** em Nome do perfil e escolha Criar função.
8. Na página Perfis, insira **`FleetAdvisorS3Role`** em Nome do perfil. Escolha FleetAdvisorS3Role.
9. Na página FleetAdvisorS3Role, escolha a guia Relações de confiança. Escolha Editar política de confiança.
10. Na página Editar política de confiança, cole o seguinte JSON no editor, substituindo o texto existente.

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "Service": [
         "dms.amazonaws.com",
```

```
          "dms-fleet-advisor.amazonaws.com"
        ]
      },
     "Action": "sts:AssumeRole"
    }
  ]
}
```

A política anterior concede a permissão `sts:AssumeRole` aos serviços que o AWS DMS utiliza para importar dados coletados no bucket do Amazon S3.

11. Escolha Atualizar política.

Criar o perfil vinculado a serviço do DMS Fleet Advisor

O DMS Fleet Advisor usa uma função vinculada ao serviço para gerenciar as CloudWatch métricas da Amazon em seu. Conta da AWS O DMS Fleet Advisor usa essa função vinculada ao serviço para publicar as métricas de desempenho do banco de dados coletadas CloudWatch em seu nome.

Como criar o perfil vinculado a serviço do DMS Fleet Advisor

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis. Depois, escolha Create role (Criar função).
3. Em Tipo de entidade confiável, escolha Serviços da AWS.
4. Para Casos de uso de outros serviços da AWS, escolha DMS – Fleet Advisor.
5. Marque a caixa de seleção DMS – Fleet Advisor e escolha Próximo.
6. Na página Adicionar permissões, escolha Próximo.
7. Na página Nomear, revisar e criar, escolha Criar função.

Como alternativa, é possível criar esse perfil vinculado a serviço na API da AWS API ou na CLI da AWS. Para ter mais informações, consulte Criar um perfil vinculado a serviço para o AWS DMS Fleet Advisor.

Depois de criar o perfil vinculado ao serviço para o DMS Fleet Advisor, é possível ver as métricas de desempenho dos bancos de dados de origem nas recomendações de destino. Além disso, você pode ver essas métricas e em sua CloudWatch conta. Para ter mais informações, consulte Recomendações de destino.

Como criar uma política do IAM necessária para o perfil vinculado a serviço do DMS Fleet Advisor

As permissões mínimas exigidas para criar um perfil vinculado ao serviço estão especificadas na política `DMSFleetAdvisorCreateServiceLinkedRolePolicy`. Crie essa política do IAM para sua conta se não conseguir criar um perfil vinculado ao serviço.

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas.
3. Escolha Create policy.
4. Na página Criar política, escolha a guia JSON.
5. Cole o seguinte código JSON no editor, substituindo o código de exemplo.

```
{
    "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. Selecione Next: Tags (Próximo: tags) e Next: Review (Próximo: revisar).
7. Insira **`DMSFleetAdvisorCreateServiceLinkedRolePolicy`** em Nome* e escolha Criar política.

Agora é possível utilizar essa política para criar o perfil vinculado a serviço do DMS Fleet Advisor.

Criar usuários do banco de dados para o AWS DMS Fleet Advisor

Esta seção descreve como criar os usuários do banco de dados com as permissões mínimas necessárias para o coletor de dados do DMS.

Esta seção contém os seguintes tópicos:

- Utilizar um usuário do banco de dados com o AWS DMS Fleet Advisor
- Criar um usuário de banco de dados com o MySQL
- Criar uma usuário do banco de dados com o Oracle
- Criar um usuário de banco de dados com o PostgreSQL
- Criar um usuário de banco de dados com o Microsoft SQL Server
- Excluir usuários do banco de dados

Utilizar um usuário do banco de dados com o AWS DMS Fleet Advisor

É possível utilizar um usuário do banco de dados que não seja `root` com o coletor de dados do DMS. Especifique o nome de usuário e a senha depois de adicionar o banco de dados ao inventário, mas antes de executar o coletor de dados. Para obter mais informações sobre como adicionar bancos de dados ao inventário, consulte Gerenciar objetos monitorados.

Após a conclusão da utilização do coletor de dados do DMS, é possível excluir os usuários do banco de dados criados. Para ter mais informações, consulte Excluir usuários do banco de dados.

> **Important**
>
> Nos exemplos a seguir, substitua *`{your_user_name}`* pelo nome do usuário do banco de dados que você criou para o banco de dados. Substitua *`{your_password}`* por uma senha segura.

Criar um usuário de banco de dados com o MySQL

Para criar um usuário de banco de dados em um banco de dados de origem do MySQL, utilize o script a seguir. Mantenha uma versão da instrução `GRANT` que depende da versão do banco de dados MySQL.

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

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

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

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

GRANT SELECT ON performance_schema.* TO {your_user_name};

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

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

Criar uma usuário do banco de dados com o Oracle

Para criar um usuário de banco de dados em um banco de dados de origem Oracle, utilize o script a seguir.

Para executar esse script SQL, conecte-se ao banco de dados Oracle utilizando privilégios de SYSDBA. Depois de executar esse script SQL, conecte-se ao banco de dados utilizando as

credenciais de usuário criadas com esse script. Além disso, utilize as credenciais desse usuário para executar o coletor de dados do DMS.

O script a seguir adiciona o prefixo C## ao nome do usuário para bancos de dados de contêineres multilocatários (CDB) Oracle.

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

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

Criar um usuário de banco de dados com o PostgreSQL

Para criar um usuário de banco de dados em um banco de dados de origem do PostgreSQL, utilize o script a seguir.

```
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}";
```

Criar um usuário de banco de dados com o Microsoft SQL Server

Para criar um usuário de banco de dados em um banco de dados de origem do Microsoft SQL Server, utilize o script a seguir.

```
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}]
```

Excluir usuários do banco de dados

Depois de concluir todas as tarefas de coleta de dados, é possível excluir os usuários do banco de dados criados para o coletor de dados do DMS. É possível utilizar os scripts a seguir para excluir os usuários com permissões mínimas dos bancos de dados.

Para excluir o usuário do banco de dados MySQL, execute o script a seguir.

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

Para excluir o usuário do banco de dados Oracle, execute o script a seguir.

```
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;
```

Para excluir o usuário do banco de dados PostgreSQL, execute o script a seguir.

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

Para excluir o usuário do banco de dados SQL Server, execute o script a seguir.

```
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
```

Descobrir bancos de dados para a migração utilizando coletores de dados

Para descobrir sua infraestrutura de dados de origem, você pode usar o AWS Application Discovery Service Agentless Collector ou coletores de dados do AWS DMS. O ADS Agentless Collector é uma aplicação local que coleta informações sobre o ambiente on-premises por meio de métodos sem agente, incluindo informações do perfil do servidor (por exemplo, sistema operacional, número de CPUs, quantidade de RAM), metadados do banco de dados e métricas de utilização. Você pode instalar o Agentless Collector como uma máquina virtual (VM) em seu ambiente do VMware

vCenter Server usando um arquivo de virtualização aberto (OVA). Um coletor de AWS DMS dados é um aplicativo do Windows que você instala em seu ambiente local. Essa aplicação se conecta ao ambiente de dados e coleta metadados e métricas de desempenho do banco de dados e de servidores analíticos on-premises. Depois que os metadados do banco de dados e as métricas de desempenho são coletados por meio do ADS Agentless Collector ou de um coletor de dados do DMS, o DMS Fleet Advisor cria um inventário de servidores, bancos de dados e esquemas que você pode migrar para a Nuvem AWS.

O coletor de dados DMS é um aplicativo do Windows que usa bibliotecas, conectores e provedores de dados .NET para se conectar aos bancos de dados de origem para descoberta e coleta de dados.

O coletor de dados do DMS é executado no Windows. No entanto, o coletor de dados do DMS pode coletar dados de todos os fornecedores de banco de dados compatíveis, independentemente do servidor de SO em que são executados.

O coletor de dados do DMS utiliza um protocolo RTPS protegido com criptografia TLS para estabelecer uma conexão segura com o DMS Fleet Advisor. Portanto, os dados são criptografados durante o trânsito por padrão.

O AWS DMS possui o número máximo de coletores de dados que é possível criar para a Conta da AWS. Consulte a seção a seguir para obter informações sobre as cotas de serviço do Cotas para o AWS Database Migration Service do AWS DMS.

Tópicos

Permissões para um coletor de dados do DMS

Os usuários do banco de dados criados para o coletor de dados do DMS devem ter permissões de leitura. No entanto, em alguns casos, o usuário do banco de dados exige a permissão `EXECUTE`.

Para ter mais informações, consulte Criar usuários do banco de dados para o AWS DMS Fleet Advisor.

O coletor de dados do DMS requer permissões adicionais para executar os scripts de descoberta.

- Para a descoberta de SO, o coletor de dados do DMS precisa de credenciais para que o servidor de domínio execute solicitações utilizando o protocolo LDAP.
- Para a descoberta de banco de dados no Linux, o coletor de dados do DMS precisa de credenciais com concessões `sudo` SSH. Além disso, configure os servidores Linux para permitir a execução de scripts SSH remotos.
- Para descobrir bancos de dados no Windows, o coletor de dados do DMS precisa de credenciais com concessões para executar consultas do Windows Management Instrumentation (WMI) e WMI Query Language (WQL) e ler o registro. Além disso, você deve configurar seus servidores Windows para permitir a execução remota de WMI, WQL e PowerShell scripts.

Criar um coletor de dados para o AWS DMS Fleet Advisor

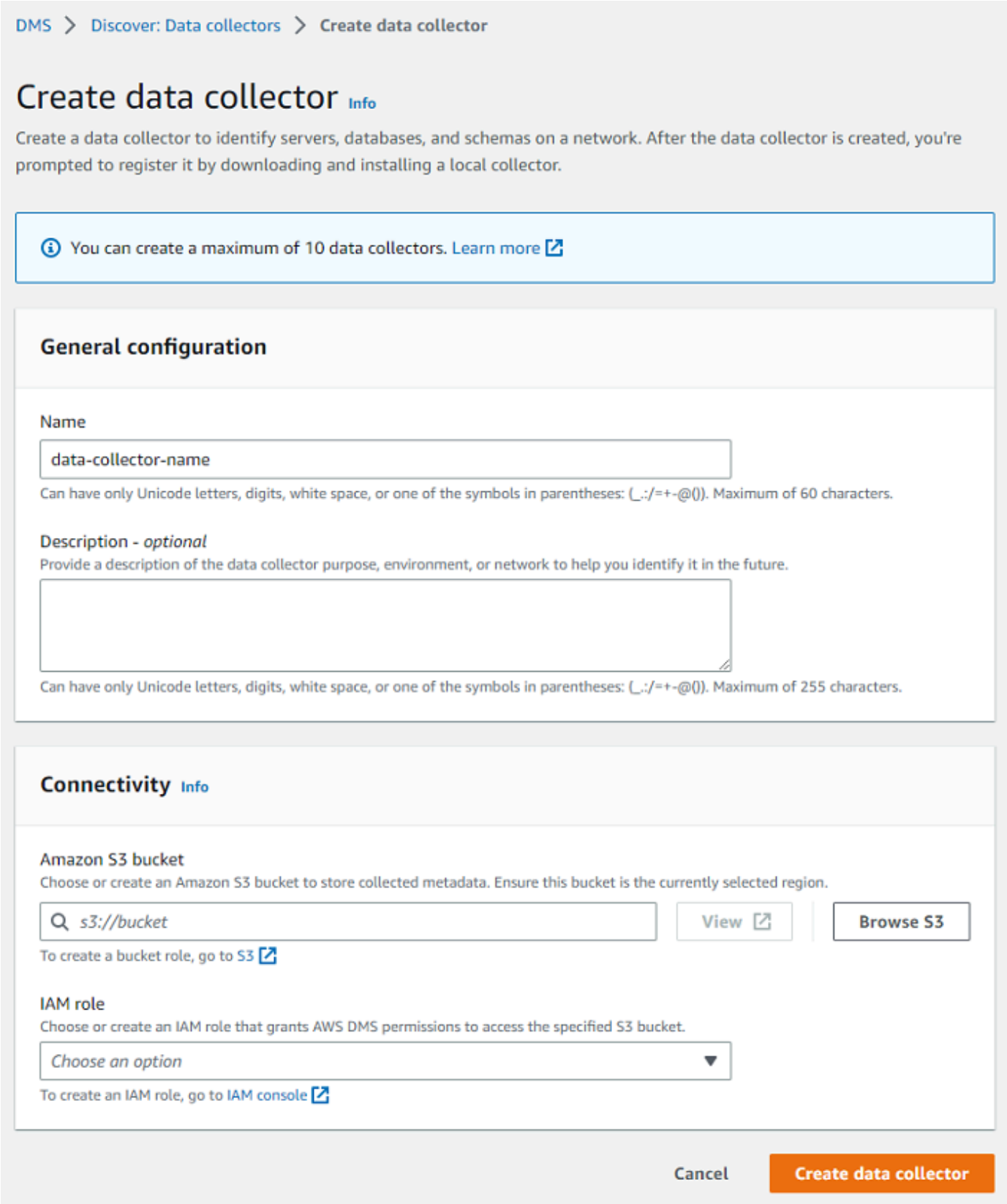
Saiba como criar e baixar um coletor de dados do DMS.

Antes de criar um coletor de dados, utilize o console do IAM para criar um perfil vinculado a serviço para o DMS Fleet Advisor. Essa função permite que os diretores publiquem pontos de dados métricos na Amazon CloudWatch. O DMS Fleet Advisor utiliza esse perfil para exibir gráficos com métricas de banco de dados. Para ter mais informações, consulte Criar um perfil vinculado a serviço para o AWS DMS Fleet Advisor.

Como criar e baixar um coletor de dados

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Selecione a região em que você utiliza o DMS Fleet Advisor.
2. No painel de navegação, escolha Coletores de dados em Descobrir. A página Coletores de dados é aberta.
3. Escolha Criar coletor de dados. A página Criar coletor de dados é aberta.



4. Em Nome na seção Configuração geral, insira o nome do coletor de dados.

5. Na seção Conectividade, escolha Procurar S3. Escolha o bucket do Amazon S3 pré-configurado na lista exibida.

   O AWS DMS armazena os metadados do inventário do DMS Fleet Advisor nesse bucket do S3. Verifique se o bucket do Amazon S3 está na mesma Região da AWS em que o AWS DMS Fleet Advisor está em execução.

   > **Note**
   >
   > O DMS Fleet Advisor só oferece suporte a buckets criptografados SSE-S3.

6. Na lista de perfis do IAM, escolha o perfil do IAM pré-configurado na lista exibida. Esse perfil concede ao AWS DMS permissões para acessar o bucket especificado do Amazon S3.
7. Escolha Criar coletor de dados. A página Coletores de dados é aberta e o coletor de dados criado aparece na lista.

   Ao criar seu primeiro coletor de dados, o AWS DMS configura um ambiente no bucket do Amazon S3 que formata dados e armazena atributos para utilização com o DMS Fleet Advisor.
8. Escolha Baixar coletor local no banner de informações para baixar o coletor de dados recém-criado. Uma mensagem informa que o download está em andamento. Após a conclusão do download, é possível acessar o arquivo `AWS_DMS_Collector_Installer_`*`version_number`*`.msi`.

Agora é possível instalar o coletor de dados do DMS em seu cliente. Para ter mais informações, consulte Instalar e configurar um coletor de dados.

Instalar e configurar um coletor de dados

Saiba como instalar o coletor de dados do DMS, como especificar credenciais de encaminhamento de dados e como adicionar um servidor LDAP ao projeto.

A tabela a seguir descreve os requisitos de hardware e de software para instalar um coletor de dados do DMS.

Mínimo	Recomendado
Processador: 2 núcleos com pontuação de benchmark de CPU superior a 8.000	Processador: 4 núcleos com pontuação de benchmark de CPU superior a 11.000

Mínimo	Recomendado
RAM: 8 GB	RAM: 16 GB
Tamanho do disco rígido: 256 GB	Tamanho do disco rígido: 512 GB
Sistema operacional: Microsoft Windows Server 2012 ou superior	Sistema operacional: Microsoft Windows Server 2016 ou superior

Como instalar um coletor de dados em um cliente na rede

1. Execute o instalador .MSI. A página Assistente de configuração do coletor do Fleet Advisor do AWS DMS é aberta.
2. Escolha Próximo. O Contrato de licença do usuário final é aberto.
3. Leia e aceite o Contrato de licença do usuário final.
4. Escolha Próximo. A página Pasta de destino é aberta.
5. Escolha Próximo para instalar o coletor de dados no diretório padrão.

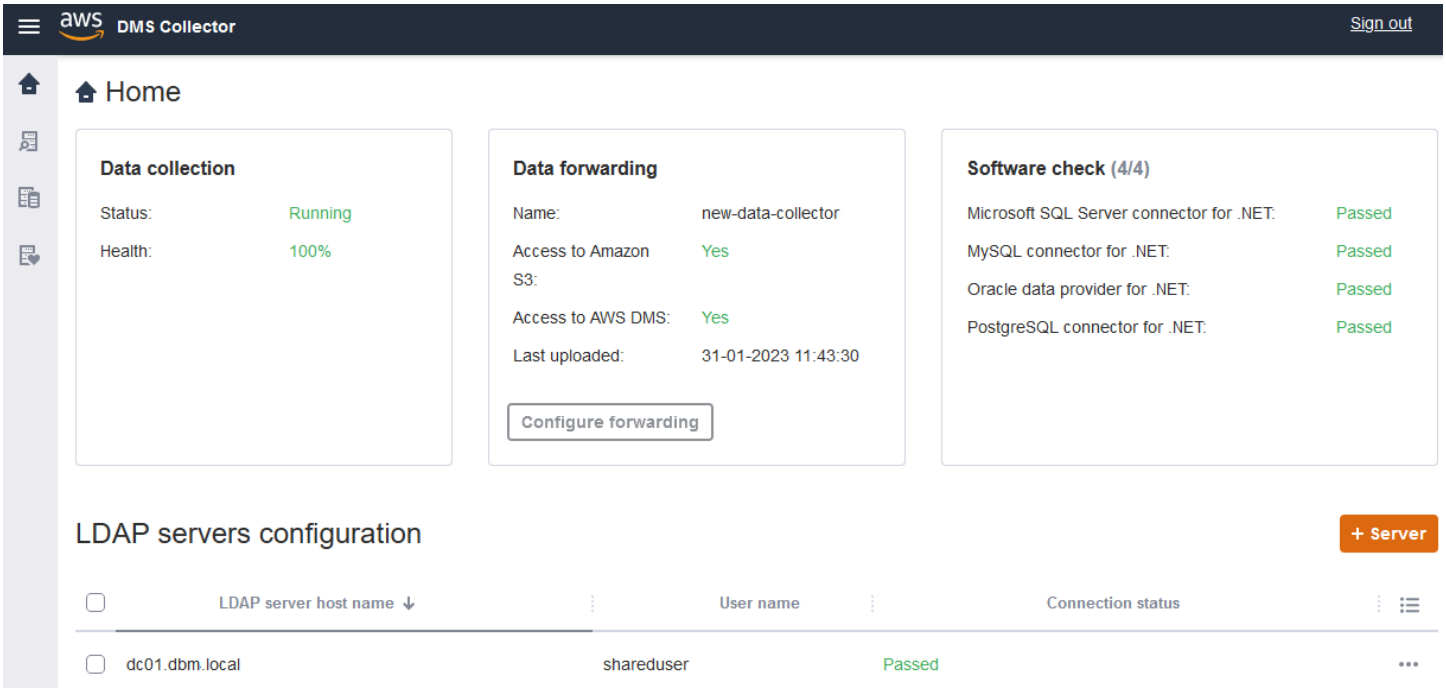
   Ou escolha Alterar para inserir outro diretório de instalação. Em seguida, escolha Próximo.
6. Na página Atalho da área de trabalho, selecione a caixa para instalar um ícone na área de trabalho.
7. Escolha Instalar. O coletor de dados é instalado no diretório escolhido.
8. Na página Assistente de configuração do coletor do DMS concluído, escolha Iniciar coletor do AWS DMS e Concluir.

O coletor de dados do DMS utiliza bibliotecas, conectores e provedores de dados .NET para conexão aos seus bancos de dados de origem. O instalador do coletor de dados do DMS instala automaticamente esse software necessário para todos os bancos de dados compatíveis no servidor.

Depois da instalação do coletor de dados, é possível executá-lo em um navegador inserindo **`http://localhost:11000/`** como o endereço. Opcionalmente, no menu Iniciar do Microsoft Windows, escolha Coletor do AWS DMS na lista de programas. Ao executar o coletor de dados do DMS pela primeira vez, você deverá configurar as credenciais. Crie o nome de usuário e a senha para fazer login no coletor de dados.

Na página inicial do coletor de dados do DMS, é possível encontrar informações para preparar e executar a coleta de metadados, incluindo as seguintes condições de status:

- Status e integridade da coleta de dados.
- Acessibilidade ao bucket do Amazon S3 e ao AWS DMS para que o coletor de dados possa encaminhar dados ao AWS DMS.
- Conectividade com os drivers de banco de dados instalados.
- Credenciais de um servidor LDAP para executar a descoberta inicial.



O coletor de dados do DMS utiliza um diretório LDAP para coletar informações sobre as máquinas e os servidores de banco de dados na rede. O Lightweight Directory Access Protocol (LDAP) é um protocolo de aplicação de padrão aberto. Ele é utilizado para acessar e manter serviços distribuídos de informações de diretórios em uma rede IP. É possível adicionar um servidor LDAP existente ao projeto do coletor de dados que você utiliza para descobrir informações sobre a infraestrutura dos sistemas. Para fazer isso, escolha a opção +Servidor e especifique um nome de domínio totalmente qualificado (FQDN) e as credenciais do controlador de domínio. Depois de adicionar o servidor, valide a verificação da conexão. Para começar a usar o processo de descoberta, consulte Descobrir sistemas operacionais e servidores de banco de dados para monitoramento.

Configurar credenciais para encaminhamento de dados

Depois de instalar o coletor de dados, verifique se essa aplicação pode enviar os dados coletados para o AWS DMS Fleet Advisor.

Como configurar credenciais para encaminhamento de dados no AWS DMS Fleet Advisor

1. Na página inicial do coletor de dados do DMS, na seção Encaminhamento de dados, escolha Configurar encaminhamento. A caixa de diálogo Configurar credenciais para encaminhamento de dados é aberta.
2. Escolha a Região da AWS em que deseja utilizar o DMS Fleet Advisor.
3. Insira o ID da chave de acesso da AWS e a Chave de acesso secreta da AWS obtida anteriormente ao criar os recursos do IAM. Para ter mais informações, consulte Criar recursos do IAM.
4. Escolha Procurar coletores de dados.

   Se você ainda não criou um coletor de dados na região especificada, crie um coletor de dados antes de continuar. Para ter mais informações, consulte Criar um coletor de dados.
5. Na janela Escolher coletor de dados, selecione um coletor de dados na lista e selecione Escolher.
6. Na caixa de diálogo Configurar credenciais para encaminhamento de dados, escolha Salvar.

Na página inicial do Coletor do DMS, no cartão de Encaminhamento de dados, verifique se os status de Acesso ao Amazon S3 e Acesso ao AWS DMS estão definidos como Sim.

Se você vir que o status de Acesso ao Amazon S3 ou de Acesso ao AWS DMS está definido como Não, verifique se criou recursos do IAM para acessar o Amazon S3 e o DMS Fleet Advisor. Depois de criar esses recursos do IAM com todas as permissões necessárias, configure o encaminhamento de dados novamente. Para ter mais informações, consulte Criar recursos do IAM.

Descobrir sistemas operacionais e servidores de banco de dados para monitoramento

É possível utilizar o coletor de dados do DMS para encontrar e listar todos os servidores disponíveis na rede. É recomendável descobrir todos os servidores de banco de dados disponíveis na rede, mas não é obrigatório. Opcionalmente, é possível adicionar ou fazer upload manualmente da lista de

servidores para posterior coleta de dados. Para obter mais informações sobre como adicionar uma lista de servidores manualmente, consulte Gerenciar objetos monitorados.

É recomendável descobrir todos os servidores de SO antes de descobrir os bancos de dados nesses servidores. Para descobrir servidores do sistema operacional, você precisa de permissão para executar scripts e comandos remotos PowerShell, do Secure Shell (SSH) e do Windows Management Instrumentation (WMI), bem como acessar o registro do Windows. Para descobrir servidores de banco de dados na rede e coletar metadados deles, é necessário ter permissões de administrador somente para leitura para uma conexão remota com o banco de dados. Verifique se adicionou um servidor LDAP antes de continuar com a descoberta. Para ter mais informações, consulte Configurar credenciais para encaminhamento de dados.

Para começar a usar o coletor de dados do DMS, conclua as seguintes tarefas:

- Descubra todos os servidores de SO na rede.
- Adicione servidores de SO específicos como objetos a serem monitorados.
- Verifique as conexões dos servidores de SO monitorados.
- Descubra bancos de dados Microsoft SQL Server, MySQL, Oracle e PostgreSQL em execução nos servidores de SO.
- Adicione servidores de banco de dados à coleta de dados.
- Verifique as conexões com os bancos de dados monitorados.

Como descobrir servidores de SO na rede que podem ser monitorados

1. No painel de navegação do coletor de dados do DMS, escolha Descoberta. Para exibir o painel de navegação, escolha o ícone do menu no canto superior esquerdo da página inicial do coletor de dados do DMS.

   A página Descoberta é aberta.
2. Verifique se a guia Servidores de SO está selecionada e escolha Executar descoberta. A caixa de diálogo Parâmetros da descoberta é aberta.
3. Insira os servidores LDAP que deseja utilizar para verificar a rede.
4. Escolha Executar descoberta. A página exibe uma lista de todos os servidores de SO descobertos na rede, independentemente de estarem executando um banco de dados.

   É recomendável executar a descoberta em todos os servidores de SO antes de executar a descoberta de bancos de dados nesses servidores. As credenciais possibilitam a descoberta

primeiro para os servidores host e depois para os bancos de dados que residem neles. Você quer descobrir os servidores de SO antes de executar a descoberta de bancos de dados nesses servidores. Lembre-se de que as credenciais que utiliza para que um servidor LDAP encontre servidores de SO na rede podem ser diferentes das credenciais necessárias para descobrir bancos de dados em um determinado servidor de SO. Portanto, é recomendável adicionar servidores de SO aos objetos monitorados, verificar as credenciais e corrigi-las, se necessário, e verificar a conectividade antes de continuar.

Na lista de servidores de SO descobertos na rede, agora é possível selecionar os servidores que deseja adicionar aos objetos monitorados.

Como selecionar servidores de SO como objetos a serem monitorados

1. Na página Descoberta, escolha a guia Servidores de SO.
2. Na lista de servidores de SO descobertos mostrada, marque a caixa de seleção ao lado de cada servidor a ser monitorado.
3. Escolha Adicionar aos objetos monitorados.

É possível ver a lista de servidores de SO a serem monitorados e verificar as conexões na página Monitorar objetos.

Como verificar as conexões dos servidores de SO selecionados a serem monitorados

1. No painel de navegação do coletor de dados do DMS, escolha Objetos monitorados.
2. Na página Objetos monitorados, escolha a guia Servidores de SO. Uma lista dos servidores de SO descobertos a serem monitorados é aberta.
3. Marque a caixa de seleção na parte superior da coluna para escolher todos os servidores de SO listados.
4. Escolha Ações e Verificar conexão. Para cada objeto de servidor, visualize os resultados na coluna Status das conexões.
5. Selecione os servidores com um status de conexão diferente de Sucesso. Escolha Ações e Editar. A caixa de diálogo Editar servidor é aberta.
6. Verifique se as informações estão corretas ou edite, se necessário. Quando terminar, escolha Save (Salvar). A caixa de diálogo Substituir credenciais é aberta.
7. Escolha Substituir. O coletor de dados do DMS verifica e atualiza o status de cada conexão como Sucesso.

Agora é possível descobrir os bancos de dados que residem nos servidores que você selecionou para monitorar.

Descubra bancos de dados em execução nos servidores

1. No painel de navegação do coletor de dados do DMS, escolha Descoberta.
2. Escolha a guia Servidores de banco de dados e escolha Executar descoberta. A caixa de diálogo Parâmetros de descoberta é aberta.
3. Na caixa de diálogo Parâmetros de descoberta, em Descoberta por, escolha Objetos monitorados. Em Servidores, escolha os servidores de SO nos quais você deseja executar a descoberta de bancos de dados.
4. Escolha Executar descoberta. A página exibe uma lista de todos os bancos de dados que residem nos servidores de SO que você escolheu monitorar.

Visualize informações como o endereço do banco de dados, o nome do servidor e o mecanismo do banco de dados para ajudar a selecionar os bancos de dados a serem monitorados.

Como selecionar os bancos de dados a serem monitorados

1. Na página Descoberta, escolha a guia Servidores de banco de dados.
2. Na lista de bancos de dados descobertos mostrada, marque a caixa de seleção ao lado de todos os bancos de dados a serem monitorados.
3. Escolha Adicionar aos objetos monitorados.

Agora é possível verificar as conexões com os bancos de dados que você escolheu monitorar.

Como verificar as conexões com os bancos de dados monitorados

1. No painel de navegação do coletor de dados do DMS, escolha Objetos monitorados.
2. Na página Objetos monitorados, escolha a guia Servidores de banco de dados. Uma lista dos servidores de banco de dados descobertos a serem monitorados é aberta.
3. Marque a caixa de seleção na parte superior da coluna para escolher todos os servidores de banco de dados listados.
4. Escolha Ações e Verificar conexão. Para cada banco de dados, visualize os resultados na coluna Status das conexões.

5. Selecione as conexões com status indefinido (em branco) ou com status de Falha. Escolha Ações e Editar. A caixa de diálogo Editar objetos monitorados é aberta.
6. Insira as credenciais de Login e Senha e escolha Salvar. A caixa de diálogo Alterar credenciais é aberta.
7. Escolha Substituir. O coletor de dados do DMS verifica e atualiza o status de cada conexão como Sucesso.

Depois de descobrir os servidores de SO e os bancos de dados a serem monitorados, também é possível executar ações para gerenciar os objetos monitorados.

Gerenciar objetos monitorados

É possível selecionar objetos para monitorar ao executar o processo de descoberta de servidor, conforme descrito em Descobrir sistemas operacionais e servidores de banco de dados. Além disso, é possível gerenciar objetos manualmente, como servidores de SO e servidores de banco de dados. É possível executar as seguintes ações para gerenciar objetos monitorados:

- Adicionar novos objetos para monitorar
- Remover objetos existentes
- Editar objetos existentes
- Exportar e importar uma lista de objetos para monitorar
- Verificar as conexões com os objetos
- Iniciar a coleta de dados

Por exemplo, é possível adicionar manualmente um objeto para monitorar.

Como adicionar um objeto para monitorar manualmente

1. Na página Objetos monitorados, escolha +Servidor. A caixa de diálogo Adicionar objeto monitorado é aberta.
2. Adicione as informações sobre o servidor e escolha Salvar.

Também é possível utilizar um arquivo `.csv` para importar uma lista grande de objetos a serem monitorados. Utilize o seguinte formato de arquivo `.csv` para importar uma lista de objetos no coletor de dados do DMS.

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

Como importar um arquivo .csv com uma lista de objetos a serem monitorados

1. Escolha Importar. A página Importar objetos monitorados é aberta.
2. Navegue até o arquivo `.csv` a ser importado e escolha Próximo.

É possível visualizar todos os objetos e selecionar aqueles nos quais deseja começar a coletar metadados.

Associar um servidor de SO a um banco de dados adicionado manualmente

O DMS Fleet Advisor não pode coletar métricas de desempenho dos bancos de dados MySQL e PostgreSQL. Para coletar as métricas necessárias para recomendações de destino, o DMS Fleet Advisor utiliza métricas do SO em que os bancos de dados são executados.

Ao adicionar manualmente os bancos de dados MySQL e PostgreSQL à lista de objetos monitorados, o coletor de dados do DMS não pode identificar os servidores de SO em que esses bancos de dados são executados. Por causa desse problema, associe os bancos de dados MySQL e PostgreSQL aos servidores de SO.

Não é necessário associar manualmente os servidores de SO aos bancos de dados que o DMS Fleet Advisor descobriu automaticamente.

Para associar um servidor de SO ao banco de dados

1. No painel de navegação do coletor de dados do DMS, escolha Objetos monitorados.
2. Na página Objetos monitorados, escolha a guia Servidores de banco de dados. Uma lista de servidores de banco de dados é aberta.
3. Marque a caixa de seleção ao lado do servidor de banco de dados MySQL ou PostgreSQL adicionado manualmente.
4. Escolha Ações e Editar. A caixa de diálogo Editar banco de dados é aberta.
5. Se o coletor de dados do DMS já tiver descoberto o servidor de SO em que esse banco de dados é executado, escolha Detectar automaticamente. O coletor de dados do DMS executa um script SQL para identificar automaticamente o servidor de SO em que o banco de dados é executado. Em seguida, o coletor de dados do DMS associa esse servidor de SO ao banco de dados. Ignore a próxima etapa e salve a configuração do banco de dados que você editou.

   Se o coletor de dados do DMS não puder identificar automaticamente o servidor de SO do banco de dados, utilize as credenciais corretas e forneça permissões de acesso ao banco de dados. Com opção, é possível adicionar o servidor de SO manualmente.
6. Para adicionar o servidor de SO manualmente, escolha +Adicionar servidor de SO. A caixa de diálogo Adicionar servidor de SO host é aberta.

   Adicione informações sobre o servidor de SO e escolha Salvar.
7. Na caixa de diálogo Editar banco de dados, escolha Verificar conexão para garantir que o coletor de dados do DMS possa se conectar ao servidor de SO.
8. Depois de verificar a conexão, escolha Salvar.

Se você alterar o servidor de SO associado ao banco de dados de origem, o DMS Fleet Advisor utilizará as métricas atualizadas para gerar recomendações. No entanto, os CloudWatch gráficos da Amazon exibem os dados antigos do seu servidor de banco de dados. Para obter mais informações sobre CloudWatch gráficos, consulteDetalhes da recomendação.

Utilizar SSL com o AWS DMS Fleet Advisor

Para proteger os dados, o AWS DMS Fleet Advisor pode utilizar SSL para acessar os bancos de dados.

Bancos de dados compatíveis

O AWS DMS Fleet Advisor é compatível com a utilização de SSL para acessar os seguintes bancos de dados:

- Microsoft SQL Server
- MySQL
- PostgreSQL

Configurar o SSL

Para utilizar o SSL para acessar o banco de dados, configure o servidor de banco de dados para ser compatível com o SSL. Para obter mais informações, consulte a seguinte documentação do banco de dados:

- SQL Server: Ativar conexões criptografadas com o mecanismo de banco de dados
- MySQL: Configurar o MySQL para utilizar conexões criptografadas
- PostgreSQL: Conexões TCP/IP seguras com SSL

Para utilizar SSL para conectar-se ao banco de dados, selecione Certificado de servidor confiável e Utilizar SSL ao adicionar um servidor manualmente. Para um banco de dados MySQL, é possível utilizar um certificado personalizado. Para utilizar um certificado personalizado, marque a caixa de seleção Verificar CA. Para obter mais informações sobre como adicionar um servidor, consulte Gerenciar objetos monitorados.

Verificar o certificado da autoridade de certificação (CA) do servidor para o SQL Server

Para validar o certificado da autoridade de certificação (CA) de servidor do SQL Server, desmarque o Certificado de servidor confiável ao adicionar o servidor. Se o servidor utilizar uma CA conhecida, e a CA estiver instalada por padrão no SO, a verificação deverá funcionar normalmente. Se o DMS Fleet Advisor não puder se conectar ao servidor de banco de dados, instale o certificado CA utilizado pelo servidor de banco de dados. Para obter mais informações, consulte Configurar cliente.

Coletar dados para o AWS DMS Fleet Advisor

Para começar a coletar dados, selecione os objetos na página Objetos monitorados e escolha Executar coleta de dados. O coletor de dados do DMS pode coletar até 100 bancos de dados ao mesmo tempo. Além disso, o coletor de dados do DMS pode utilizar até oito threads paralelos para se conectar aos bancos de dados no ambiente. Nesses oito threads, o coletor de dados do DMS pode utilizar até cinco threads paralelos para se conectar a uma única instância de banco de dados.

> **Important**
>
> Antes de começar a coletar dados, consulte a seção Verificação de software na página inicial do coletor de dados do DMS. Verifique se todos os mecanismos de banco de dados que deseja monitorar possuem o status Aprovado. Se alguns mecanismos de banco de dados tiverem o status Com falha e você tiver servidores de banco de dados com mecanismos correspondentes na lista de objetos monitorados, corrija o problema antes de continuar. É possível encontrar dicas ao lado do status Com falha listado na seção Verificação de software.

O coletor de dados do DMS pode funcionar em dois modos: execução única ou monitoramento contínuo. Depois de iniciar a coleta de dados, a caixa de diálogo Executar coleta de dados é aberta. Escolha uma das seguintes opções:

Metadados e capacidade do banco de dados

O coletor de dados do DMS coleta informações do banco de dados ou dos servidores de SO. Ele inclui esquemas, versões, edições, CPU, memória e capacidade de disco. O coletor de dados do DMS também coleta e fornece métricas como IOPS, taxa de transferência de E/S e conexões ativas do servidor de banco de dados. É possível calcular as recomendações de destino no DMS Fleet Advisor com base nessas informações. Se o banco de dados de origem estiver superprovisionado ou subprovisionado, as recomendações de destino também serão superprovisionadas ou subprovisionadas.

Esta é a opção padrão.

Metadados, capacidade do banco de dados e utilização de recursos

Além das informações de metadados e capacidade do banco de dados, o coletor de dados do DMS coleta métricas reais de CPU, memória e capacidade de disco dos bancos de dados

ou servidores de SO. O coletor de dados do DMS também coleta e fornece métricas como IOPS, taxa de transferência de E/S e conexões ativas do servidor de banco de dados. As recomendações de destino fornecidas serão mais precisas porque se baseiam nas workloads reais do banco de dados.

Se você escolher essa opção, definirá o período da coleta de dados. É possível coletar dados durante os Próximos 7 dias ou definir o Intervalo personalizado de 1 a 60 dias.

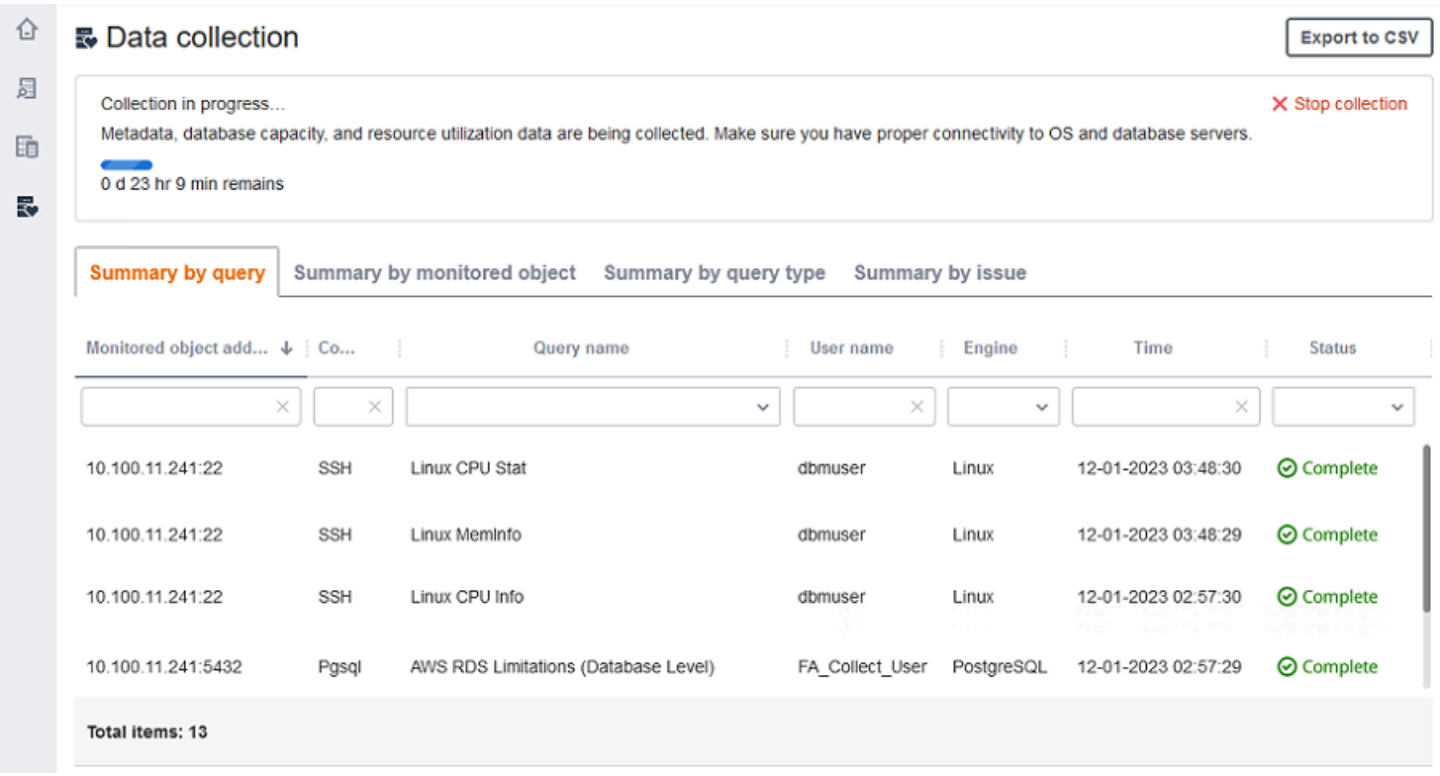
Após o início da coleta de dados, você será redirecionado para a página Coleta de dados, em que poderá ver como as consultas de coleta são executadas e monitorar o progresso em tempo real. Aqui, é possível visualizar a integridade geral da coleta na página inicial do coletor de dados do DMS. Se a integridade geral da coleta de dados for inferior a 100%, talvez seja necessário corrigir os problemas relacionados à coleta.

Se você executar o coletor de dados do DMS no modo Metadados e capacidade do banco de dados, poderá ver o número de consultas concluídas na página Coleta de dados.

Se você executar o coletor de dados do DMS no modo Metadados, capacidade do banco de dados e utilização de recursos, poderá ver o tempo restante antes que o coletor de dados do DMS conclua o monitoramento.

Na página Coleta de dados, é possível ver o status da coleta de cada objeto. Se algo não funcionar de forma adequada, uma mensagem será exibida mostrando quantos problemas ocorreram. Para ajudar a determinar uma correção para um problema, é possível verificar os detalhes. As guias a seguir listam os possíveis problemas:

- Resumo por consulta: mostra o status de testes, como o teste Ping. É possível filtrar os resultados na coluna Status. A coluna Status fornece uma mensagem indicando quantas falhas ocorreram durante a coleta de dados.
- Resumo por objeto monitorado: mostra o status geral por objeto.
- Resumo por tipo de consulta: mostra o status do tipo de consulta do coletor, como chamadas de SQL, Secure Shell (SSH) ou Windows Management Instrumentation (WMI).
- Resumo por problema: mostra todos os problemas exclusivos que ocorreram, com os nomes dos problemas e o número de vezes em que cada problema ocorre.



Para exportar os resultados da coleta, escolha Exportar para CSV.

Depois de identificar os problemas e resolvê-los, escolha Iniciar coleta e execute novamente o processo de coleta de dados. Depois de executar a coleta de dados, o coletor de dados utiliza conexões seguras para fazer upload dos dados coletados em um inventário do DMS Fleet Advisor. O DMS Fleet Advisor armazena informações no bucket do Amazon S3. Para obter informações sobre como configurar credenciais para encaminhamento de dados, consulte Configurar credenciais para encaminhamento de dados.

Coletar métricas de capacidade e utilização de recursos com o AWS DMS Fleet Advisor

É possível coletar metadados e métricas de desempenho em dois modos: execução única ou monitoramento contínuo. Dependendo da opção selecionada, o coletor de dados do DMS rastreia diferentes métricas no ambiente de dados. Durante uma única execução, o coletor de dados do DMS só rastreia métricas de metadados do banco de dados e dos servidores de SO. Durante o monitoramento contínuo, o coletor de dados do DMS rastreia a utilização real dos recursos.

AWS DMSreúne os seguintes metadados e métricas durante uma única execução do seu coletor de dados do DMS.

- Memória disponível nos servidores de SO
- Armazenamento disponível nos servidores de SO
- Versão e edição do banco de dados
- Número de CPUs nos servidores de SO
- Número de esquemas
- O número máximo de procedimentos armazenados.
- Número de tabelas
- Número de acionadores
- Número de visualizações
- Estrutura do esquema

O DMS Fleet Advisor utiliza essas métricas para criar um inventário de bancos de dados e de servidores de SO. Além disso, o DMS Fleet Advisor usa esses metadados e métricas para analisar os esquemas do banco de dados de origem.

O DMS Fleet Advisor pode gerar recomendações de metas usando as métricas coletadas durante uma única execução do coletor de dados. No entanto, nesse caso, para seus bancos de dados de origem superprovisionados, a recomendação de destino também é superprovisionada. Assim, você incorre em custos adicionais com a manutenção de seus recursos noNuvem AWS. Para bancos de dados de origem subprovisionados, a recomendação de destino também é subprovisionada, o que pode levar a problemas de desempenho. Recomendamos coletar os dados usando o monitoramento contínuo, escolhendo os metadados, a capacidade do banco de dados e o modo de utilização de recursos para o coletor de dados do DMS.

O AWS DMS reúne as seguintes métricas durante o monitoramento contínuo. É possível executar o coletor de dados do DMS por um período de 1 a 60 dias.

- Throughput de E/S nos servidores de banco de dados
- Operações de entrada e saída por segundo (IOPS) nos servidores de banco de dados
- Número de CPUs que os servidores de SO utilizam
- Utilização de memória nos servidores de SO
- Número de conexões ativas do banco de dados e do servidor do sistema operacional

O DMS Fleet Advisor utiliza essas métricas para gerar recomendações precisas de destino, para que os bancos de dados de destino atendam às necessidades de desempenho. Isso pode evitar custos adicionais incorridos na manutenção de seus recursos noNuvem AWS.

Como AWS DMS Fleet Advisor coleta a capacidade e as métricas de utilização de recursos?

O DMS Fleet Advisor coleta métricas de desempenho a cada minuto.

Para o Oracle e o SQL Server, o DMS Fleet Advisor executa consultas SQL para capturar valores de cada métrica do banco de dados.

Para o MySQL e o PostgreSQL, o DMS Fleet Advisor coleta métricas de desempenho do servidor de SO em que o banco de dados é executado. No Windows, o DMS Fleet Advisor executa scripts WMI Query Language (WQL) e recebe dados WMI. No Linux, o DMS Fleet Advisor executa comandos que capturam as métricas do servidor de SO.

> **Important**
>
> A execução de scripts SQL remotos pode afetar o desempenho dos bancos de dados de produção. No entanto, as consultas de coleta de dados não contêm nenhuma lógica de cálculo. Portanto, o processo da coleta de dados não utiliza mais de 1% dos recursos do banco de dados.

É possível visualizar todas as consultas que o coletor de dados executa para coletar métricas. Para isso, abra o arquivo `DMSCollector.Collections.json`. É possível encontrar esse arquivo na pasta `etc` localizada na mesma pasta em que você instalou o coletor de dados. O caminho padrão é `C:\ProgramData\Amazon\AWS DMS Collector\etc\DMSCollector.Collections.json`.

O coletor de dados do DMS utiliza o sistema de arquivos local como armazenamento temporário para todos os dados coletados. O coletor de dados do DMS armazena os dados coletados no formato JSON. É possível utilizar o coletor local em modo off-line e conferir manualmente ou verificar os arquivos coletados antes de configurar o encaminhamento de dados. É possível ver todos os arquivos coletados na pasta `out` localizada na mesma pasta em que você instalou o coletor de dados do DMS. O caminho padrão é `C:\ProgramData\Amazon\AWS DMS Collector\out`.

> **Important**
>
> Se você executar seu coletor de dados DMS em um modo off-line e armazenar os dados coletados em seu servidor por mais de 14 dias, não poderá usar CloudWatch a Amazon para exibir essas métricas. No entanto, o DMS Fleet Advisor ainda utiliza esses dados para gerar recomendações. Para obter mais informações sobre CloudWatch gráficos, consulteDetalhes da recomendação.

Também é possível conferir ou verificar os arquivos de dados coletados no modo on-line. O coletor de dados do DMS encaminha todos os dados para o bucket do Amazon S3 especificado nas configurações do coletor de dados do DMS.

É possível utilizar o coletor de dados do DMS para coletar dados de bancos de dados on-premises. Além disso, é possível coletar dados dos bancos de dados Amazon RDS e Aurora. No entanto, não é possível executar com êxito todas as consultas do coletor de dados do DMS na nuvem, devido às diferenças entre o Amazon RDS ou o Aurora e as instâncias de banco de dados on-premises. Como o coletor de dados do DMS reúne métricas de utilização dos bancos de dados MySQL e PostgreSQL do SO host, essa abordagem não funcionará com o Amazon RDS e o Aurora.

Solução de problemas para o coletor de dados do DMS

Na lista a seguir, é possível encontrar ações a serem executadas ao encontrar problemas específicos ao coletar dados com o coletor de dados.

Tópicos

- Problemas de coleta de dados relacionados às conexões de rede e servidor
- Problemas de coleta de dados relacionados ao Windows Management Instrumentation
- Problemas de coleta de dados relacionados ao compositor de páginas da web do Windows
- Problemas de coleta de dados relacionados ao SSL

Problemas de coleta de dados relacionados às conexões de rede e servidor

NET: ocorreu uma exceção durante uma solicitação de ping.

Confira o nome do computador para verificar se ele está em um estado em que não pode ser resolvido para um endereço IP.

Por exemplo, verifique se o computador está desligado, desconectado da rede ou desativado.

NET: tempo limite

Ative a regra de firewall de entrada “Compartilhamento de arquivos e de impressora (Echo Request - ICMPv4-In)”. Por exemplo: .

* Inbound ICMPv4

REDE: DestinationHostUnreachable

Verifique o endereço IP do computador. Especificamente, verifique se ele está na mesma sub-rede do computador que executa o coletor de dados do DMS e se ele responde às solicitações do Address Resolution Protocol (ARP).

Se o computador estiver em uma sub-rede diferente, o endereço IP do gateway não poderá ser resolvido para o endereço de controle de acesso à mídia (MAC).

Além disso, verifique se o computador está desligado, desconectado da rede ou desativado.

Problemas de coleta de dados relacionados ao Windows Management Instrumentation

WMI: o servidor RPC está indisponível. (Exceção do HRESULT: 0x800706BA)

Ative a regra de firewall de entrada "Windows Management Instrumentation (DCOM–In)". Por exemplo: .

* Inbound TCP/IP at local port 135.

Além disso, ative a regra de firewall de entrada "Windows Management Instrumentation (WMI-In)". Por exemplo: .

* Inbound TCP/IP at local port 49152 – 65535 para Windows Server 2008 e versões superiores.

* Inbound TCP/IP at local port 1025 – 5000 para Windows Server 2003 e versões inferiores.

WMI: acesso negado. (Exceção do HRESULT: 0x80070005 (E_ACCESSDENIED))

Faça o seguinte:

- Adicione o usuário do coletor de dados do DMS ao grupo do Windows, usuários ou administradores COM distribuídos.
- Inicie o serviço Windows Management Instrumentation e defina seu tipo de inicialização como Automático.
- Verifique se o nome de usuário do coletor de dados do DMS está no formato \.

WMI: acesso negado

Adicione a permissão Ativar remoto ao usuário do coletor de dados do DMS no namespace do WMI raiz.

Utilize as Configurações avançadas e verifique se as permissões se aplicam a “Este namespace e subnamespaces”.

WMI: a chamada foi cancelada pelo filtro de mensagens. (Exceção do HRESULT: 0x80010002...)

Reinicie o serviço Windows Management Instrumentation.

Problemas de coleta de dados relacionados ao compositor de páginas da web do Windows

WPC: o caminho da rede não foi encontrado

Ative a regra de firewall de entrada “Compartilhamento de arquivos e de impressora (PME-In)". Por exemplo: .

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

Além disso, inicie o serviço Registro remoto e defina seu tipo de inicialização como Automático.

WPC: o acesso é negado

Adicione o usuário coletor de dados do DMS ao grupo de usuários ou administradores do monitor de desempenho.

WPC: a categoria não existe

Execute `loader /r` para reconstruir o cache do contador de desempenho e reinicie o computador.

> **Note**
>
> Para obter informações sobre a solução de problemas ao migrar dados utilizando o AWS Database Migration Service (AWS DMS), consulte Suporte para solução de problemas e diagnóstico.

Problemas de coleta de dados relacionados ao SSL

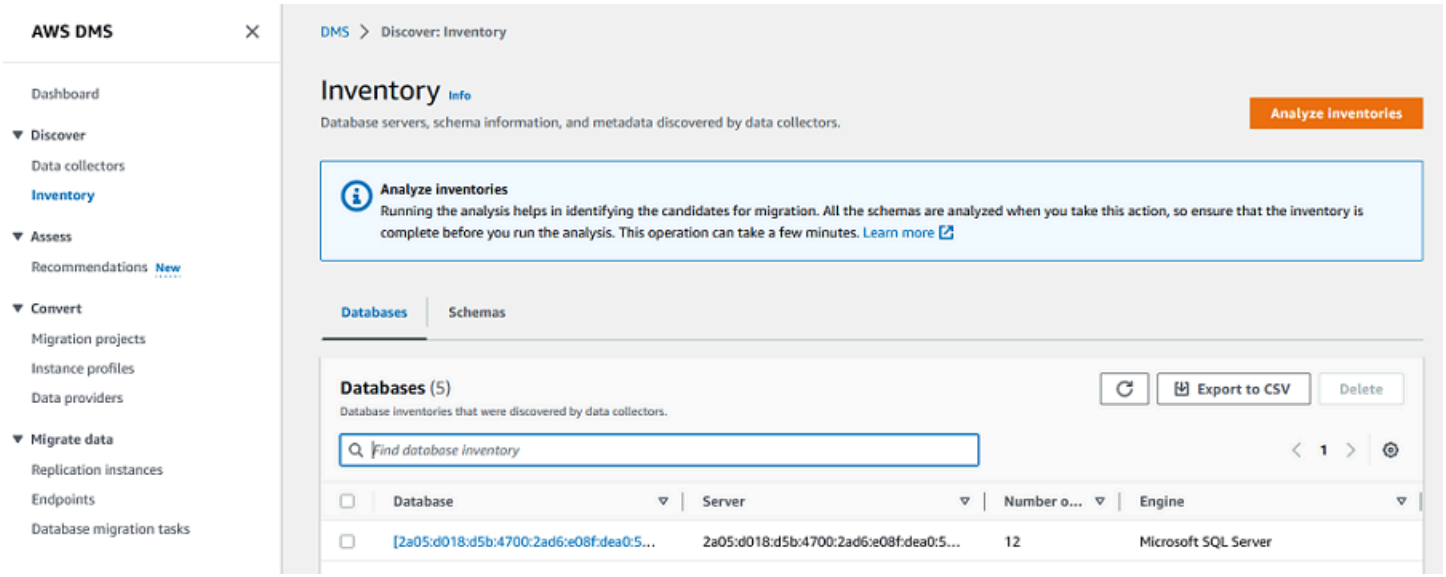
Erros de SSL

O banco de dados requer uma conexão SSL segura, e você não ativou as opções Verificar CA e Utilizar SSL para a conexão. Ative essas opções e verifique se o SO local tem a autoridade de certificação instalada que o banco de dados utiliza. Para ter mais informações, consulte Configurar o SSL.

Utilizar inventários para análise no AWS DMS Fleet Advisor

Para verificar a viabilidade de possíveis migrações de banco de dados, é possível trabalhar com os inventários dos bancos de dados e esquemas descobertos. É possível utilizar as informações desses inventários para compreender quais bancos de dados e esquemas são bons candidatos à migração.

É possível acessar os inventários de bancos de dados e esquemas no console. Para fazer isso, escolha Inventário no console.



O DMS Fleet Advisor analisa os esquemas de banco de dados para determinar a semelhança de diferentes esquemas. Essa análise não compara o código real dos objetos. O DMS Fleet Advisor compara somente os nomes dos objetos do esquema, como perfis e procedimentos, para identificar objetos semelhantes em diferentes esquemas dos bancos de dados.

Tópicos

Utilizar um inventário de bancos de dados para análise

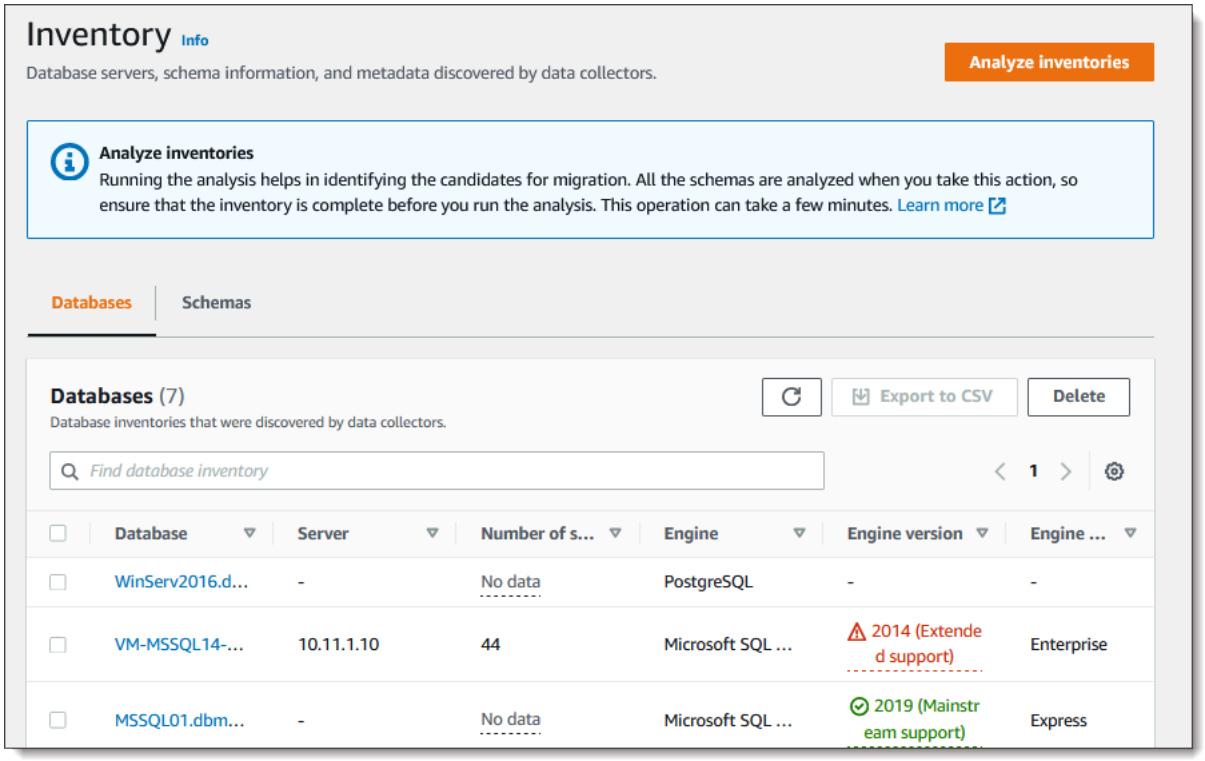
Para ver uma lista de todos os bancos de dados em todos os servidores descobertos na rede nos quais os dados foram coletados, utilize o procedimento a seguir.

Como visualizar uma lista de bancos de dados nos servidores da rede nos quais os dados foram coletados

1. Escolha Inventário no console.

   A página Inventário é aberta.
2. Escolha a guia Databases (Bancos de dados).

   Uma lista dos bancos de dados descobertos é exibida.



3. Escolha Analisar inventários para determinar as propriedades dos esquemas, como semelhança e complexidade. A quantidade de tempo que o processo leva depende do número de objetos a serem analisados, mas não levará mais de uma hora. Os resultados da análise são encontrados na guia Esquemas localizada na página Inventário.

   O DMS Fleet Advisor analisa esquemas em todos os bancos de dados descobertos para definir a interseção de seus objetos. O resultado da análise é expresso em porcentagem. O DMS Fleet Advisor considera os esquemas com interseções de mais de 50% como duplicatas. O esquema original é identificado como o esquema no qual foram encontradas duplicatas. Isso ajuda a identificar os esquemas originais a serem convertidos ou migrados primeiro.

   Todo o inventário é analisado em conjunto para identificar esquemas duplicados.

Utilizar um inventário de esquemas para análise

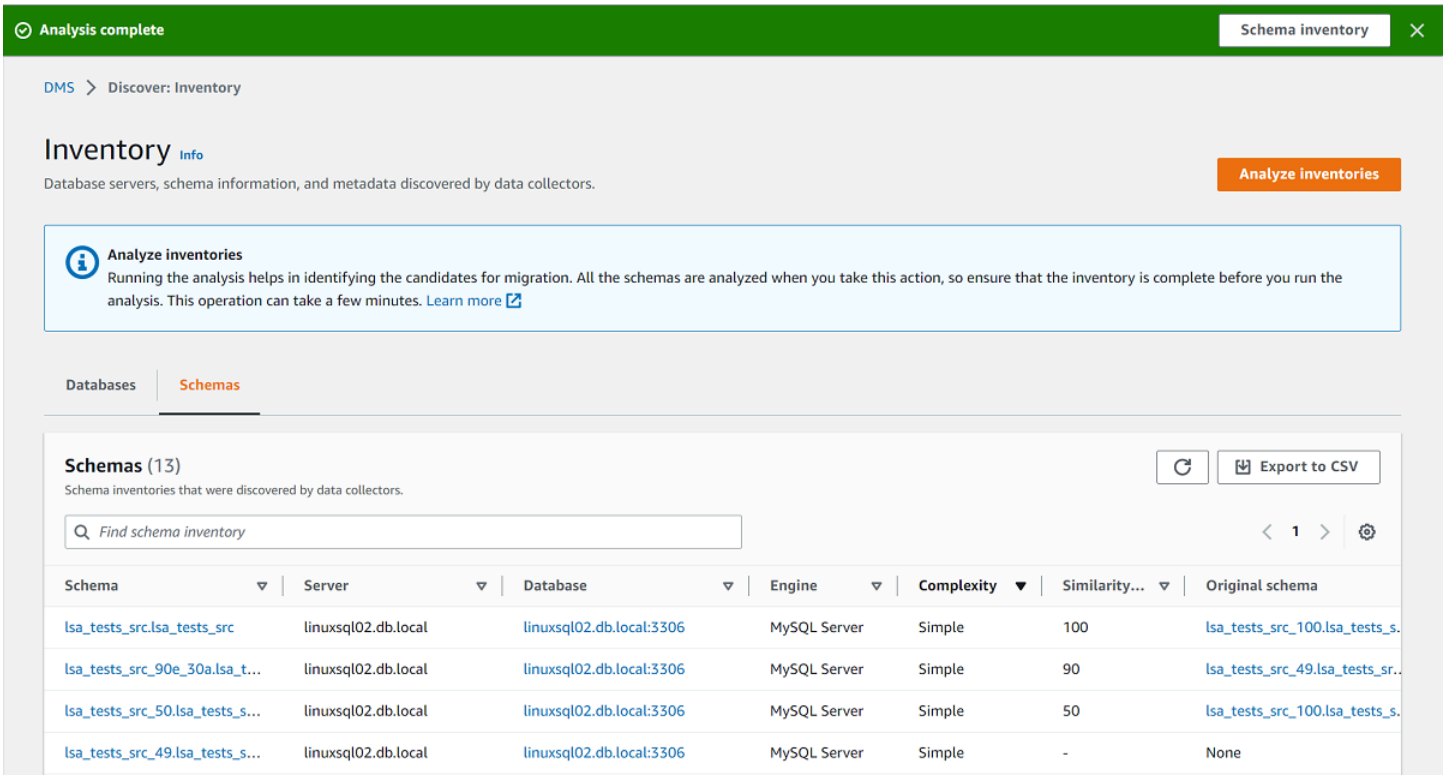
É possível visualizar uma lista dos esquemas de bancos de dados descobertos nos servidores da rede nos quais os dados foram coletados. Execute o procedimento a seguir.

Como visualizar uma lista de esquemas nos servidores da rede nos quais os dados foram coletados

1. Escolha Inventário no console. A página Inventário é aberta.

2. Escolha a guia Esquemas. Uma lista de esquemas é exibida.
3. Selecione um esquema na lista para visualizar as informações sobre ele, incluindo servidor, banco de dados, tamanho e complexidade.

   Para cada esquema, é possível visualizar um resumo dos objetos que fornece informações sobre tipos de objetos, número de objetos, tamanho do objeto e linhas de código.
4. (Opcional) Escolha Analisar inventários para identificar esquemas duplicados. O DMS Fleet Advisor analisa esquemas de bancos de dados para definir a interseção dos objetos.
5. É possível exportar as informações do inventário para um arquivo `.csv` para análise posterior.



Para identificar esquemas a serem migrados e determinar o destino da migração, você pode usar AWS Schema Conversion Tool (AWS SCT) ou DMS Schema Conversion. Para obter mais informações, consulte Utilizar um novo assistente de projeto no AWS SCT.

Depois de identificar os esquemas a serem migrados, é possível converter os esquemas utilizando o AWS SCT ou a DMS Schema Conversion. Para obter mais informações sobre a DMS Schema Conversion, consulte Converter esquemas de banco de dados utilizando a DMS Schema Conversion.

Utilizar o recurso Recomendações de destino do AWS DMS Fleet Advisor

Para explorar e escolher um destino de migração ideal, é possível gerar recomendações de destino para os bancos de dados on-premises de origem no DMS Fleet Advisor. Uma recomendação inclui um ou mais mecanismos de destino possíveis da AWS que é possível escolher para a migração do banco de dados on-premises de origem. A partir desses possíveis mecanismos de destino, o DMS Fleet Advisor sugere um único mecanismo de destino como o destino de migração do tamanho certo e indica esse alvo como recomendado pelo DMS. Para determinar esse destino de migração de tamanho certo, o DMS Fleet Advisor utiliza os metadados e métricas de inventário coletados pelo coletor de dados.

É possível utilizar as recomendações antes do início de uma migração para descobrir as opções de migração, economizar custos e reduzir riscos. É possível exportar as recomendações como um arquivo de valores separados por vírgula (CSV) e compartilhá-lo com as partes interessadas principais para facilitar a tomada de decisões. Você pode exportar recomendações para o AWS Pricing Calculator para otimizar ainda mais os custos de manutenção. Para obter mais informações, consulte https://calculator.aws/#/.

Não é possível modificar as recomendações de destino no DMS Fleet Advisor. Portanto, não é possível utilizar o DMS Fleet Advisor para análises hipotéticas. A análise hipotética é o processo de alterar os parâmetros de destino para ver como essas alterações afetam a estimativa de preços da recomendação. É possível executar uma análise hipotética no AWS Pricing Calculator utilizando os parâmetros de destino recomendados como ponto de início no AWS Pricing Calculator. Para obter mais informações, consulte https://calculator.aws/#/.

É recomendável considerar que a recomendação do DMS Fleet Advisor é um ponto de início no planejamento de uma migração. É possível decidir alterar os parâmetros de instância recomendados para otimizar o custo ou o desempenho das workloads do banco de dados.

Tópicos

- Exportar as recomendações de destino com o AWS DMS Fleet Advisor
- Descobrindo e analisando as limitações de migração com o AWS DMS Fleet Advisor
- Solução de problemas de recomendações de destino

Instâncias de destino recomendadas

Para recomendações de destino, o DMS Fleet Advisor considera as seguintes instâncias de banco de dados do Amazon RDS de uso geral, otimizadas para memória e desempenho com capacidade de intermitência.

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

Para obter mais informações sobre as classes de instâncias de banco de dados do Amazon RDS, consulte Classes de instâncias de banco de dados no Guia do usuário do Amazon RDS.

Como o DMS Fleet Advisor determina as especificações da instância de destino para a recomendação?

O DMS Fleet Advisor pode gerar recomendações com base na capacidade ou na utilização do banco de dados.

- Se você optar por gerar a recomendação com base na capacidade do banco de dados, o DMS Fleet Advisor mapeará a capacidade do banco de dados existente para as especificações da classe de instância mais próxima.
- Se você optar por gerar a recomendação com base na utilização de recursos, o DMS Fleet Advisor determinará o valor do 95º percentil para métricas, como CPU, memória, throughput de E/S e IOPS. O 95º percentil significa que 95% dos dados coletados são menores que esse valor. O DMS Fleet Advisor mapeia esses valores para as especificações da classe de instância mais próxima.

Para determinar o tamanho do banco de dados de destino, o DMS Fleet Advisor coleta informações sobre o tamanho do banco de dados de origem. O DMS Fleet Advisor recomenda utilizar o mesmo tamanho para o armazenamento de destino. Se o armazenamento do banco de dados de origem estiver superprovisionado, o tamanho recomendado do armazenamento de destino também será superprovisionado.

Se você quiser migrar dados utilizando o AWS DMS, talvez seja necessário aumentar o provisionamento de IOPS para a instância de banco de dados de destino. Quando o DMS Fleet Advisor gera recomendações de destino, o serviço considera somente as métricas do banco de dados de origem. O DMS Fleet Advisor não considera IOPS adicionais que podem ser necessárias para executar tarefas de migração de dados. Para ter mais informações, consulte As tarefas de migração são executadas lentamente.

Para estimar os custos de IOPS, o DMS Fleet Advisor usa um one-to-one mapeamento do uso de sua fonte de IOPS como linha de base. O DMS Fleet Advisor considera o pico de carga como o valor de linha de base e a utilização de 100% para preços de IOPS.

Para bancos de dados de origem PostgreSQL e MySQL, o DMS Fleet Advisor pode incluir instâncias de bancos de dados Aurora e Amazon RDS nas recomendações de destino. Se uma configuração do Aurora for mapeada para os requisitos de origem, o DMS Fleet Advisor marcará essa opção como recomendada.

Gerar recomendações de origem com o AWS DMS Fleet Advisor

Depois de concluir a coleta de dados e o inventário do banco de dados e da frota de análise, é possível gerar recomendações de destino no DMS Fleet Advisor. Para fazer isso, escolha bancos de dados de origem e defina as configurações que o recurso Recomendações de destino do DMS Fleet Advisor utiliza para determinar o tamanho das instâncias de destino. Além disso, o recurso Recomendações de destino do DMS Fleet Advisor utiliza as métricas de capacidade e de utilização coletadas nos bancos de dados de origem.

Como gerar recomendações de destino

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Verifique se você escolheu a Região da AWS na qual você utiliza o DMS Fleet Advisor.
2. No painel de navegação, escolha Recomendações em Avaliar e escolha Gerar recomendações.

3. No painel Selecionar bancos de dados de origem, marque as caixas de seleção dos nomes dos bancos de dados que você deseja migrar para a Nuvem AWS.

   Em Pesquisar bancos de dados de origem, insira o nome do banco de dados para filtrar o inventário.

   O DMS Fleet Advisor pode gerar recomendações para até 100 bancos de dados ao mesmo tempo.

4. Em Disponibilidade e durabilidade, escolha a opção de implantação preferida.

   Para calcular as recomendações de destino para os bancos de dados de produção, escolha Produção (Multi-AZ). O DMS Fleet Advisor inclui duas instâncias de banco de dados em diferentes zonas de disponibilidade na recomendação de destino. Essa opção de implantação Multi-AZ fornece alta disponibilidade, redundância de dados e suporte a failover.

   Se o Aurora for o mecanismo de destino recomendado e a Disponibilidade e Durabilidade for uma implantação Multi-AZ, a recomendação de destino inclui uma instância de banco de dados de leitura e gravação.

   Para calcular as recomendações de destino para bancos de dados utilizados para desenvolvimento ou teste, escolha Dev/Teste (Single-AZ). O DMS Fleet Advisor inclui uma única instância de banco de dados na recomendação de destino. Essa opção de implantação Single-AZ reduz os custos de manutenção.

5. Para o Dimensionamento da instância de destino, escolha a opção preferida que o DMS Fleet Advisor utiliza para calcular as recomendações de destino.

   Para calcular as recomendações de destino com base no banco de dados de origem ou na configuração do servidor do sistema operacional, escolha Capacidade total. O DMS Fleet Advisor utiliza essas métricas, como capacidade total de CPU, de memória e de disco dos bancos de dados de origem ou servidores do sistema operacional, para gerar recomendações de destino. O DMS Fleet Advisor mapeia as métricas de capacidade do banco de dados para as especificações da classe de instância de banco de dados do Amazon RDS mais próxima.

   Para calcular as recomendações de destino com base na utilização real do banco de dados de origem ou do servidor do sistema operacional, escolha Utilização de recursos. O DMS Fleet Advisor utiliza métricas de utilização da CPU, da memória e da capacidade de disco dos bancos de dados de origem ou de servidores do sistema operacional para gerar recomendações de destino. De acordo coma as métricas de utilização, o DMS Fleet Advisor calcula o 95º percentil

de cada métrica. 95º percentil significa que 95% dos dados desse período são inferiores a esse valor. O DMS Fleet Advisor mapeia esses valores para as especificações da classe de instância mais próxima do banco de dados Amazon RDS.

É recomendável usar a opção Utilização de recursos para obter recomendações mais precisas. Para isso, verifique se você coletou as métricas de capacidade total e de utilização de recursos.

6. Escolha Gerar.

O DMS Fleet Advisor gera as recomendações de destino para os bancos de dados selecionados. Para recomendações geradas com sucesso, o DMS Fleet Advisor define o status como Calculado. Além disso, o DMS Fleet Advisor utiliza o AWS Pricing Calculator para determinar o custo mensal estimado para a instância de banco de dados de destino recomendada. Agora, é possível explorar detalhadamente as recomendações geradas. Para ter mais informações, consulte Detalhes da recomendação.

Para estimar o custo mensal total do inventário de dados, marque as caixas de seleção dos bancos de dados que você planeja mover para a nuvem. O DMS Fleet Advisor exibe o custo mensal total estimado e o resumo dos bancos de dados de destino na Nuvem AWS. O DMS Fleet Advisor utiliza a API de consulta do AWS Price List para fornecer detalhes de preços apenas em nível informativo. As taxas reais dependem de vários fatores, incluindo a utilização real dos Serviços da AWS. Para obter mais informações sobre os preços do AWS service (Serviço da AWS), consulte Preços de serviços em nuvem.

Explorar os detalhes das recomendações de destino com o AWS DMS Fleet Advisor

Depois que o DMS Fleet Advisor gerar as recomendações de destino, é possível visualizar os parâmetros-chave do destino da migração recomendada na tabela Recomendações. Esses parâmetros-chave incluem o mecanismo, a classe da instância, o número de CPUs virtuais, a memória, o armazenamento e o tipo de armazenamento do destino. Além desses parâmetros, o DMS Fleet Advisor exibe o custo mensal estimado desse destino de migração recomendada.

Cada recomendação pode incluir um ou mais mecanismos de destino possíveis da AWS. Se a recomendação incluir vários mecanismos de destino, o AWS DMS marcará um deles como recomendado. Além disso, o AWS DMS exibe os parâmetros e o custo mensal estimado para essa opção recomendada na tabela Recomendações.

Para comparar as recomendações de destino com a utilização e a capacidade do banco de dados de origem, explore as recomendações em detalhes. Além disso, é possível visualizar as limitações da migração de uma recomendação selecionada. Essas limitações incluem recursos de banco de dados não compatíveis, itens de ação e outras considerações de migração.

Como explorar a recomendação em detalhes

1. Gere recomendações de destino com o DMS Fleet Advisor. Para ter mais informações, consulte Gerar recomendações de destino.
2. Escolha o nome da recomendação na tabela Recomendações. A página de recomendação é aberta.
3. Se a recomendação incluir mais de uma opção de destino, em Recomendações de destino, escolha a opção de destino.
4. Expanda a seção Utilização e capacidade da origem. O DMS Fleet Advisor exibe gráficos de utilização de recursos para as seguintes métricas.

   - Número de DPUs
   - Memória
   - Throughput de E/S
   - Operações de entrada e saída por segundo (IOPS)
   - Armazenamento
   - Número de conexões ativas do servidor de banco de dados

   Utilize esses gráficos para comparar as métricas do banco de dados de origem do coletor de dados do DMS com as métricas do mecanismo de destino selecionado.

   Se você não conseguir ver os gráficos depois de expandir a seção de utilização e capacidade da fonte, certifique-se de conceder ao usuário do IAM permissões para visualizar os CloudWatch painéis da Amazon. Para obter mais informações, consulte Usando CloudWatch painéis da Amazon no Guia do CloudWatch usuário da Amazon.
5. Escolha o link com o nome do mecanismo de destino selecionado. A página Detalhes do destino é aberta.
6. Para exportar as recomendações de destino para CSV, escolha a opção Exportar para CSV no menu suspenso Ações.

7. Para exportar as recomendações de destino paraAWS Pricing Calculator, escolha a AWS Pricing Calculator opção Otimizar custo com no menu suspenso Ações.
8. Na seção Configuração, compare os valores dos parâmetros do banco de dados de origem com os parâmetros do mecanismo de destino. Para o mecanismo de destino, o DMS Fleet Advisor exibe os custos mensais estimados para os seus recursos de nuvem. O DMS Fleet Advisor utiliza a API de consulta do AWS Price List para fornecer detalhes de preços apenas em nível informativo. As taxas reais dependem de vários fatores, incluindo a utilização real dos Serviços da AWS. Para obter mais informações sobre os preços do AWS service (Serviço da AWS), consulte Preços de serviços em nuvem https://aws.amazon.com/pricing/.
9. Na seção Limitações da migração, visualize as limitações da migração. É recomendável considerar essas limitações ao migrar o banco de dados de origem para a Nuvem AWS.

Exportar as recomendações de destino com o AWS DMS Fleet Advisor

Depois de gerar as recomendações de destino, é possível salvar uma cópia da lista de recomendações como um arquivo de valores separados por vírgula (CSV).

Como gerar recomendações de destino

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Verifique se você escolheu a Região da AWS na qual você utiliza o DMS Fleet Advisor.
2. No painel de navegação, escolha Recomendações em Avaliar e selecione as recomendações a serem incluídas no arquivo CSV.
3. Escolha Exportar para CSV, insira o nome do arquivo e escolha a pasta em seu PC na qual salvar esse arquivo.
4. Abra o arquivo CSV.

O arquivo CSV com as recomendações contém as seguintes informações.

- CreatedDate— A data em que o DMS Fleet Advisor criou a recomendação do motor alvo.
- DatabaseId— O identificador do banco de dados de origem para o qual o DMS Fleet Advisor criou essa recomendação.
- DeploymentOption— A opção de implantação para a instância de banco de dados Amazon RDS recomendada.

- EngineEdition— A edição alvo recomendada do mecanismo Amazon RDS.
- EngineName— O nome do motor alvo.
- InstanceMemory— A quantidade de memória na instância de banco de dados Amazon RDS recomendada.
- InstanceSizingType— O tamanho da sua instância de destino.
- InstanceType— O tipo de instância alvo recomendado do Amazon RDS.
- InstanceVcpu— O número de CPUs virtuais na instância de banco de dados Amazon RDS recomendada.
- Preferencial: um sinalizador booliano que indica que essa opção de destino é recomendada.
- Status: o status da recomendação do mecanismo de destino.
- StorageIops— O número de operações de E/S concluídas a cada segundo (IOPS) na instância de banco de dados Amazon RDS recomendada.
- StorageSize— O tamanho de armazenamento da instância de banco de dados Amazon RDS recomendada.
- StorageType— O tipo de armazenamento da instância de banco de dados Amazon RDS recomendada.
- WorkloadType— A opção de implantação para seu mecanismo de destino, como implantação Multi-AZ ou Single-AZ.

Descobrindo e analisando as limitações de migração com o AWS DMS Fleet Advisor

É possível utilizar o coletor de dados do DMS para descobrir recursos de banco de dados com os quais o mecanismo de destino não é compatível. Para escolher o destino certo da migração, considere essas limitações.

O coletor de dados do DMS descobre recursos específicos do banco de dados de origem. Em seguida, o DMS Fleet Advisor analisa os recursos de origem do ponto de vista da migração para o destino especificado e fornece informações adicionais sobre a limitação e inclui ações recomendadas para abordar ou evitar essa limitação. Além disso, o DMS Fleet Advisor calcula o impacto dessas limitações.

A lista de limitações está disponível na página de detalhes do mecanismo Target. Navegue até essa página a partir da página Recomendações no menu de navegação à esquerda. Na lista de alvos,

escolha o mecanismo de destino a ser examinado. A lista de limitações está na parte inferior da página.

A tabela a seguir inclui recursos do banco de dados MySQL com os quais o Amazon RDS para MySQL não é compatível.

Limitação	Descrição	Impacto
Plug-ins de autenticação	O Amazon RDS não é compatível com os plug-ins de autenticação do MySQL.	Baixo
Registro de erros no log do sistema	O Amazon RDS não é compatível com a gravação do log de erros no log do sistema.	Baixo
Identificadores de transações globais	É possível utilizar identific adores de transações globais com todos os RDS para as versões 5.7 do MySQL e os RDS para as versões 8.0.26 e superior do MySQL e as versões 8.0 e superior do MySQL.	Baixo
Group Replication	O Amazon RDS não é compatível com o plug-in MySQL Group Replication.	Baixo
Criptografia de espaço para tabela do InnoDB	O Amazon RDS não é compatível com a criptogra fia de espaço para tabela do InnoDB.	Baixo
Palavra reservada InnoDB	InnoDB é uma palavra reservada do Amazon RDS para MySQL. Não é possível	Baixo

Limitação	Descrição	Impacto
	utilizar esse nome para um banco de dados MySQL.	
Plug-in Keyring	O Amazon RDS não é compatível com o plug-in Keyring do MySQL.	Baixo
Plug-in de validação de senha	O Amazon RDS não é compatível com o plug-in `validate_password` MySQL.	Baixo
Variáveis de sistema persistid as	O Amazon RDS não é compatível com variáveis de sistema persistidas do MySQL.	Baixo
Acesso restrito	O Amazon RDS restringe o acesso a alguns procedime ntos e tabelas do sistema que exigem privilégios avançados . Além disso, o Amazon RDS não permite o acesso direto do host a uma instância de banco de dados utilizando Telnet, Secure Shell (SSH) ou Windows Remote Desktop Connection.	Baixo
Plug-in de regravação de consulta de regravador	O Amazon RDS não é compatível com o plug-in de regravação de consultas do regravador do MySQL.	Baixo

Limitação	Descrição	Impacto
Replicação semissíncrona	O Amazon RDS não é compatível com a replicação semissíncrona do MySQL.	Baixo
Espaços para tabela transport áveis	O Amazon RDS não é compatível com o recurso de espaços para tabela transport áveis do MySQL.	Baixo
Plug-in X	O Amazon RDS não é compatível com o Plug-in X do MySQL.	Baixo

A tabela a seguir inclui recursos do Amazon RDS para Oracle que são incompatíveis com o Oracle.

Limitação	Descrição	Impacto
Active Data Guard	Não é possível utilizar o Active Data Guard com bancos de dados de contêineres multilocatários (CDB) do Oracle.	Médio
Automatic Storage Management	O Amazon RDS não é compatível com o Oracle Automatic Storage Management (Oracle ASM).	Médio
Fluxos de atividades do banco de dados	O Amazon RDS não é compatível com o Oracle Database Activity Streams para a arquitetura de locatário único.	Alta

Limitação	Descrição	Impacto
Limites de tamanho de arquivo	O tamanho máximo de um único arquivo em instâncias de banco de dados do RDS para Oracle é 16 TiB.	Médio
FTP e SFTP	O Amazon RDS não é compatível com FTP e SFTP.	Médio
Tabelas particionadas híbridas	O Amazon RDS não é compatível com tabelas particionadas híbridas do Oracle.	Alta
Oracle Data Guard	O Amazon RDS não utiliza o Oracle Data Guard para a arquitetura de locatário único.	Alta
Oracle Database Vault	Amazon RDS não é compatível com o Oracle Database Vault.	Alta
O Oracle DBA privilegia o Vault	O Amazon RDS tem limitaçõe s para privilégios do Oracle DBA. Para obter mais informações, consulte Limitações de privilégios do Oracle DBA.	Alta
Oracle Enterprise Manager	O Amazon RDS não utiliza o Oracle Enterprise Manager para a arquitetura de locatário único.	Alta

Limitação	Descrição	Impacto
Oracle Enterprise Manager Agent	O Amazon RDS não utiliza o Oracle Enterprise Manager Agent para a arquitetura de locatário único.	Médio
Oracle Enterprise Manager Cloud Control Management Repository	Não é possível utilizar uma instância de banco de dados Amazon RDS para Oracle para o Oracle Enterpris e Manager Cloud Control Management Repository.	Alta
Oracle Flashback Database	O Amazon RDS Oracle não é compatível com os seguintes recursos do Oracle Flashback Database.	Alta
Oracle Label Security	O Amazon RDS não é compatível com o Oracle Label Security para a arquitetu ra de locatário único. É possível utilizar o Oracle Label Security somente com bancos de dados de contêineres multilocatários (Oracle CDB).	Alta
Oracle Messaging Gateway	O Amazon RDS não é compatível com o Oracle Messaging Gateway.	Alta
Oracle Real Application Clusters	O Amazon RDS não é compatível com o Oracle Real Application Clusters (Oracle RAC).	Alta

Limitação	Descrição	Impacto
Oracle Real Application Testing	O Amazon RDS não é compatível com o Oracle Real Application Testing.	Alta
Bancos de dados Oracle Snapshot Standby	O Amazon RDS não é compatível com os bancos de dados Oracle Snapshot Standby.	Alta
Sinônimos públicos	O Amazon RDS não é compatível com sinônimos públicos para esquemas fornecidos pelo Oracle.	Médio
Schemas para recursos sem suporte	O Amazon RDS não é compatível com esquemas para recursos e componentes do Oracle que exigem privilégios do sistema.	Alta
Auditoria unificada pura	O Amazon RDS não é compatível com a auditoria unificada pura. É possível utilizar a auditoria unificada no modo misto.	Médio
Workspace Manager	O Amazon RDS não é compatível com o esquema do Oracle Database Workspace Manager`WMSYS`.	Alta

A tabela a seguir inclui recursos do banco de dados PostgreSQL com os quais o Amazon RDS para PostgreSQL não é compatível.

Limitação	Descrição	Impacto
Conexões simultâneas	O número máximo de conexões simultâneas à instância do RDS para PostgreSQL é limitado pelo parâmetro `max_connections` .	Baixo
Versões mais recentes	O Amazon RDS não aplica atualizações de versões principais automaticamente. Para executar uma atualização de versão principal, modifique manualmente a instância do banco de dados. Para obter mais informações, consulte Escolher uma atualização de versão principal para o PostgreSQL.	Baixo
Conexões reservadas	O Amazon RDS reserva até três conexões para manutenção do sistema. Se você especificar um valor para o parâmetro de conexões de usuário, adicione três ao número de conexões que você prevê utilizar.	Baixo
Extensões compatíveis	O RDS para PostgreSQL é compatível com um número limitado de extensões para o mecanismo de banco de dados PostgreSQL. É possível encontrar uma lista de extensões compatíve	Baixo

Limitação	Descrição	Impacto
	is no grupo de parâmetros padrão de banco de dados para a versão do PostgreSQL. Também é possível ver a lista de extensões atuais que utilizam o `psql` exibindo o parâmetro `rds.extensions` .	
Divisão ou isolamento de espaços para tabela	Não é possível utilizar espaços para tabela para divisão ou isolamento de E/S. No RDS para PostgreSQL, todo o armazenamento está em um único volume lógico.	Baixo

A tabela a seguir inclui recursos de banco de dados do SQL Server com os quais o Amazon RDS para SQL Server não é compatível.

Limitação	Descrição	Impacto
Fazer backup no armazenamento de Blobs do Microsoft Azure	O RDS para SQL Server não é compatível com o backup no Microsoft Azure Blob Storage.	Médio
Extensão do grupo de buffer	O RDS para SQL Server não é compatível com a extensão do grupo de buffers.	Alta
Gerenciar políticas de senha	O RDS para SQL Server não é compatível com políticas de senha personalizada.	Médio
Serviços de qualidade de dados	O RDS para SQL Server não é compatível com o SQL	Alta

Limitação	Descrição	Impacto
	Server Data Quality Services (DQS).	
Envio de logs de banco de dados	O RDS para SQL Server não é compatível com o envio de logs de banco de dados.	Alta
Nomes dos bancos de dados	Os nomes dos bancos de dados têm as seguintes limitações: não podem começar com rdsadmin; não podem começar nem terminar com um espaço ou uma tab; não podem conter nenhum dos caracteres que criam uma nova linha; não podem conter uma aspa simples (').	Médio
Snapshots do banco de dados	O RDS para SQL Server não é compatível com instantân eos de banco de dados. É possível utilizar somente snapshots de instâncias de banco de dados no Amazon RDS.	Médio
Procedimentos armazenados estendidos	O RDS para SQL Server não é compatível com procedime ntos armazenados estendidos, incluindo o `xp_cmdshell` .	Alta
Tabelas de arquivos	O RDS para SQL Server não é compatível com tabelas de arquivo.	Médio

Limitação	Descrição	Impacto
Suporte a FILESTREAM	O RDS para SQL Server não é compatível com FILESTREAM.	Médio
Servidores vinculados	O RDS para SQL Server tem compatibilidade limitada com servidores vinculados.	Alta
Machine Learning e R Services	O RDS para SQL Server não é compatível com o Machine Learning e R Services porque você precisa de acesso ao SO para instalar esses serviços.	Alta
Planos de manutenção	O RDS para SQL Server é compatível com planos de manutenção.	Alta
Coletor de dados de performance	O RDS para SQL Server não é compatível com o coletor de dados de desempenho.	Alta
Gerenciamento baseado em políticas	O RDS para SQL Server não é compatível com o gerenciamento baseado em políticas.	Médio
PolyBase	O RDS para SQL Server não oferece suporte PolyBase.	Alta
Replicação	O RDS para SQL Server não é compatível com a replicação.	Médio
Regulador de recursos	O RDS para SQL Server não é compatível com o Resource Governor.	Alta

Limitação	Descrição	Impacto
Triggers no nível do servidor	O RDS para SQL Server não é compatível com acionadores em nível de servidor.	Médio
Endpoints do Service Broker	O RDS para SQL Server não é compatível com endpoints de agentes de serviço.	Alta
SSAS	Considere as limitações que se aplicam à execução do SQL Server Analysis Services (SSAS) no RDS para SQL Server. Para obter mais informações, consulte Limitações.	Baixo
SSIS	Considere as limitações que se aplicam à execução do SQL Server Integration Services (SSIS) no RDS para SQL Server. Para obter mais informações, consulte Limitações.	Baixo
SSRS	Considere as limitações que se aplicam à execução do SQL Server Reporting Services (SSRS) no RDS para SQL Server. Para obter mais informações, consulte Limitações.	Baixo

Limitação	Descrição	Impacto
Tamanho de armazenamento de instâncias de banco de dados SQL Server	O tamanho máximo de armazenamento de instâncias de armazenamento do SQL Server General Purpose (SSD) e de instâncias de armazenamento de IOPS provisionadas é 16 TiB. O tamanho de armazenamento máximo para instâncias de armazenamento magnético do SQL Server Magnetic é 1 TiB.	Alta
Stretch Database	O RDS para SQL Server não é compatível com o recurso SQL Server Stretch Database.	Médio
Endpoints T-SQL	O RDS para SQL Server não é compatível com todas as operações que utilizam `CREATE ENDPOINT`.	Alta
Propriedade de banco de dados TRUSTWORTHY	O RDS para SQL Server não é compatível com a propriedade do banco de dados `TRUSTWORTHY` porque ela exige o perfil `sysadmin`.	Médio

A tabela a seguir inclui uma lista de problemas de recomendação. O DMS Fleet Advisor analisa os recursos do banco de dados de origem e destino e fornece essas limitações de migração. A limitação do impacto do Blocker significa que o DMS Fleet Advisor não pode gerar recomendações de destino para o banco de dados de origem.

Limitação	Descrição	Impacto
A instância apropriada não foi encontrada	AWS DMSnão consigo encontrar uma instância de destino que possa funcionar como um destino de migração do tamanho certo para uma combinação das métricas do seu banco de dados de origem.	Bloqueador
A instância apropriada não foi encontrada pelo IOPS	O banco de dados de origem usa várias IOPS, o que excede o número máximo de IOPS para as possíveis instâncias de banco de dados de destino.	Bloqueador
A instância apropriada não foi encontrada pela RAM	O banco de dados de origem usa vários GB de RAM, o que excede o tamanho máximo de RAM para as possíveis instâncias de banco de dados de destino.	Bloqueador
A instância apropriada não foi encontrada pelo tamanho do armazenamento	O banco de dados de origem usa vários TB de armazenam ento, o que excede o tamanho máximo de armazenamento para as possíveis instâncias de banco de dados de destino.	Bloqueador
A instância apropriada não foi encontrada por edição	O banco de dados de origem tem uma edição, que não é suportada pelo Amazon RDS.	Bloqueador

Limitação	Descrição	Impacto
A instância apropriada não foi encontrada pelos núcleos da CPU	O banco de dados de origem tem vários núcleos de CPU, o que excede o número máximo de núcleos de CPU para as possíveis instâncias de banco de dados de destino.	Bloqueador
A instância apropriada não foi encontrada por versão	Seu banco de dados de origem tem uma versão que AWS DMS não reconhece.	Bloqueador
O parâmetro da CPU é indefinido	O coletor de dados do DMS não coletou informações sobre a CPU que seu banco de dados de origem usa. Verifique se você coletou as métricas necessárias e configurou as credenciais para encaminhamento de dados em seu coletor de dados. Consulte Configurar credenciais para encaminhamento de dados.	Bloqueador

Limitação	Descrição	Impacto
O parâmetro de memória é indefinido	O coletor de dados do DMS não coletou informações sobre a memória que seu banco de dados de origem usa. Verifique se você coletou as métricas necessárias e configurou as credenciais para encaminhamento de dados em seu coletor de dados. Consulte Configurar credenciais para encaminhamento de dados.	Bloqueador
O parâmetro de tamanho de armazenamento é indefinido	O coletor de dados do DMS não coletou informações sobre o tamanho de armazenamento que seu banco de dados de origem usa. Verifique se você coletou as métricas necessárias e configurou as credenciais para encaminhamento de dados em seu coletor de dados. Consulte Configurar credenciais para encaminhamento de dados.	Bloqueador

Limitação	Descrição	Impacto
O parâmetro IOPS de armazenamento é indefinido	O coletor de dados do DMS não coletou as métricas de IOPS de armazenamento para seus usos do banco de dados de origem. Verifique se você coletou as métricas necessári as e configurou as credencia is para encaminhamento de dados em seu coletor de dados.	Bloqueador
Dados insuficientes	O coletor de dados do DMS não coletou dados suficientes para gerar uma recomenda ção de destino. Certifiqu e-se de ter configurado as credenciais para encaminha mento de dados em seu coletor de dados. Consulte Configurar credenciais para encaminhamento de dados.	Bloqueador
A edição do banco de dados é indefinida	O coletor de dados do DMS não coletou informações sobre a edição do banco de dados de origem. Verifique se você coletou as métricas necessári as e configurou as credencia is para encaminhamento de dados em seu coletor de dados. Consulte Configurar credenciais para encaminha mento de dados.	Bloqueador

Limitação	Descrição	Impacto
Erro desconhecido	O DMS Fleet Advisor não pode gerar recomendações de destino para seu banco de dados de origem.	Bloqueador
A versão do banco de dados é indefinida	O DMS Fleet Advisor não coletou informações sobre a versão do seu banco de dados de origem. O DMS Fleet Advisor recomenda que você use a versão mais recente do banco de dados para seu banco de dados de origem. Se você escolher essa recomendação, deverá atualizar sua versão do banco de dados. Analise as recomendações de destino geradas para seu banco de dados de origem e certifique-se de que essas recomendações atendam aos seus requisitos.	Alta

Limitação	Descrição	Impacto
Aumente o número de conexões de banco de dados nas configurações do RDS	Seu banco de dados de origem requer um certo número de conexões. Por padrão, o número de conexões disponíveis para instâncias de banco de dados do Amazon RDS é diferente. Certifique-se de alterar esse valor padrão ao criar sua instância de banco de dados do RDS. Para fazer isso, atualize o valor do `max_connections` parâmetro em Parameter Groups.	Médio
A edição Target é compatível	A recomendação de destino para seu banco de dados de origem usa uma edição de banco de dados diferente. Sua edição de banco de dados de origem oferece suporte aos mesmos recursos da edição de destino recomendada. No entanto, escolher essa nova edição do banco de dados pode aumentar suas despesas.	Médio

Limitação	Descrição	Impacto
O parâmetro de taxa de transferência de armazenamento é indefinido	O coletor de dados do DMS não coletou as métricas de taxa de transferência de armazenamento para o uso do banco de dados de origem. Analise as recomendações de destino geradas para seu banco de dados de origem e certifique-se de que essas recomendações atendam aos seus requisitos.	Médio
O parâmetro do número de conexão do banco de dados é indefinido	O coletor de dados do DMS não coletou informações sobre o número de conexões que seu banco de dados de origem usa. Analise as recomendações de destino geradas para seu banco de dados de origem e certifique-se de que essas recomendações atendam aos seus requisitos. Como alternativa, solicite um aumento de cota.	Médio

Limitação	Descrição	Impacto
Versão de downgrade do banco de dados	Seu banco de dados de origem é executado em uma versão superior à do banco de dados Amazon RDS. Para fazer o downgrade da versão do seu banco de dados, certifique-se de não usar os recursos que não estão implementados na versão inferior. Como alternativa, use o Amazon EC2 como destino de migração.	Médio
A edição Target é diferente	A recomendação de destino para seu banco de dados de origem usa uma edição de banco de dados diferente. Sua edição do banco de dados de origem é compatível com a edição de destino recomenda da. No entanto, a edição recomendada do banco de dados de destino não oferece suporte a alguns recursos da edição do banco de dados de origem. A escolha dessa nova edição do banco de dados pode aumentar suas despesas.	Médio

Limitação	Descrição	Impacto
Atualize a partir de uma versão não suportada	Seu banco de dados de origem chegou ao fim do estágio de suporte. Para usar a versão mais recente do mecanismo de banco de dados como destino, atualize seu banco de dados antes da migração. Como alternativa, use o Amazon EC2 como destino de migração. Dependendo do mecanismo do banco de dados, use um dos links a seguir para saber mais: Atualizando o MySQL Atualize o SQL Server Atualize o OracleDB Atualize o PostgreSQL	Médio

Solução de problemas de recomendações de destino

Na lista a seguir, é possível encontrar as ações a serem tomadas ao encontrar problemas com o recurso Recomendações de destino do DMS Fleet Advisor.

Tópicos

- Não é possível ver as estimativas de preços das recomendações de destino
- Não é possível ver os gráficos de utilização do recurso
- Não é possível ver o status da coleção de métricas

Não é possível ver as estimativas de preços das recomendações de destino

Se você vir Sem dados para o Custo mensal estimado de uma recomendação com status de Sucesso, conceda ao usuário do IAM as permissões para acessar a API do serviço AWS Price List. Para fazer isso, crie a política que inclui a permissão `pricing:GetProducts` e adicione-a ao usuário do IAM conforme descrito em Criar recursos do IAM.

O DMS Fleet Advisor não calcula o custo mensal estimado para recomendações com o status Com falha.

Não é possível ver os gráficos de utilização do recurso

Se você ver a mensagem Falha ao carregar métricas depois de expandir a seção de utilização e capacidade da fonte, certifique-se de conceder ao usuário do IAM permissões para visualizar os CloudWatch painéis da Amazon. Para isso, adicione a política necessária ao usuário do IAM, conforme descrito em Criar recursos do IAM.

Como alternativa, é possível criar uma política personalizada que inclua as permissões `cloudwatch:GetDashboard`, `cloudwatch:ListDashboards`, `cloudwatch:PutDashboard` e `cloudwatch:DeleteDashboards`. Para obter mais informações, consulte Usando CloudWatch painéis da Amazon no Guia do CloudWatch usuário da Amazon.

Não é possível ver o status da coleção de métricas

Se você vir Não há dados disponíveis para Coleção de métricas ao escolher Gerar recomendações, verifique se os dados foram coletados. Para ter mais informações, consulte Coletar dados para o AWS DMS Fleet Advisor.

Se você tiver esse problema depois de coletar os dados, certifique-se de conceder ao usuário do IAM a `cloudwatch:Get*` permissão para acessar a Amazon CloudWatch. O DMS Fleet Advisor usa uma função vinculada ao serviço para publicar as métricas de desempenho do banco de dados coletadas CloudWatch em seu nome. Certifique-se de criar uma função vinculada ao serviço para usar com o DMS Fleet Advisor. Para ter mais informações, consulte Criar recursos do IAM.

Limitações do DMS Fleet Advisor

As limitações ao usar o DMS Fleet Advisor incluem o seguinte:

- O DMS Fleet Advisor gera one-to-one recomendações. Para cada banco de dados de origem, o DMS Fleet Advisor determina um único mecanismo de destino. O DMS Fleet Advisor não trata

servidores multilocatários e não fornece recomendações para executar vários bancos de dados em uma única instância de banco de dados de destino.

- O DMS Fleet Advisor não fornece recomendações sobre as atualizações disponíveis da versão do banco de dados.
- O DMS Fleet Advisor pode gerar recomendações para até 100 bancos de dados ao mesmo tempo.
- Se você instalar o coletor de dados DMS, que é um aplicativo do Windows, certifique-se de instalar também o.NET Framework 4.8 e PowerShell 6.0 e superior. Para obter os requisitos de hardware, consulte Instalar um coletor de dados.
- O coletor de dados do DMS exige permissões para executar solicitações utilizando o protocolo LDAP no servidor de domínio.
- O coletor de dados do DMS requer o script sudo SSH em execução no Linux.
- O coletor de dados do DMS requer permissões para executar scripts remotos PowerShell, Windows Management Instrumentation (WMI), WMI Query Language (WQL) e scripts de registro no Windows.
- Para MySQL e PostgreSQL, o DMS Fleet Advisor não pode coletar métricas de desempenho do banco de dados. Em vez disso, o DMS Fleet Advisor coleta as métricas do servidor do sistema operacional. Portanto, não é possível gerar recomendações com base em métricas de utilização para bancos de dados MySQL e PostgreSQL executados no Amazon RDS e no Aurora.

Converter esquemas de banco de dados utilizando a DMS Schema Conversion

A conversão do esquema DMS em AWS Database Migration Service (AWS DMS) torna as migrações de banco de dados entre diferentes tipos de bancos de dados mais previsíveis. É possível utilizar a DMS Schema Conversion para avaliar a complexidade da migração para o provedor de dados de origem e para converter os esquemas de banco de dados e objetos de código. É possível aplicar o código convertido ao banco de dados de destino.

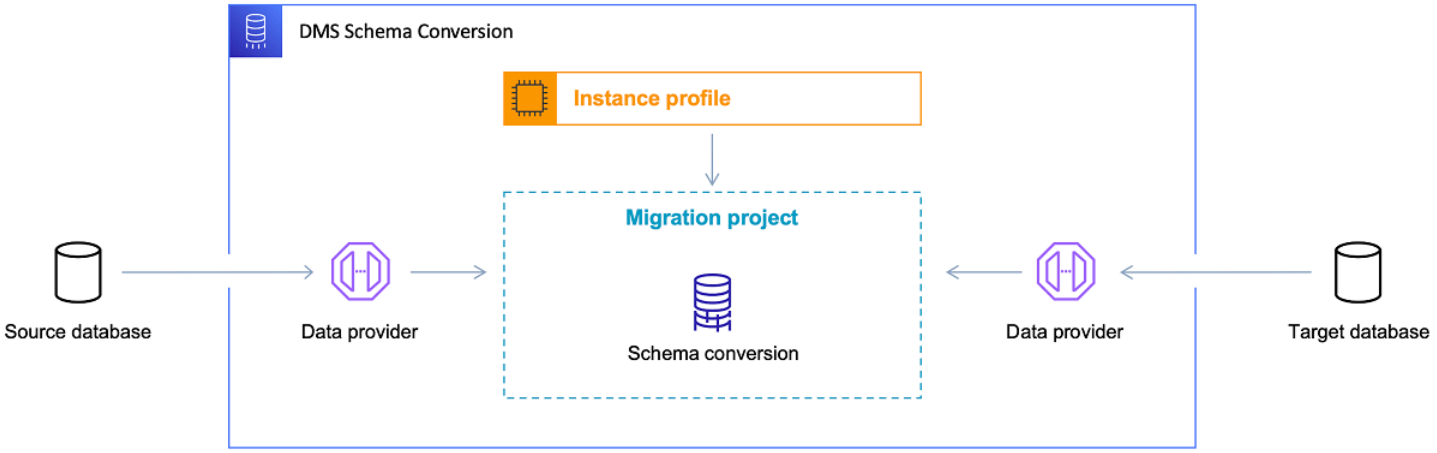
A DMS Schema Conversion converte automaticamente os esquemas do banco de dados de origem e a maioria dos objetos de código do banco de dados em um formato compatível com o banco de dados de destino. Essa conversão inclui tabelas, visualizações, procedimentos armazenados, perfis, tipos de dados, sinônimos e assim por diante. Todos os objetos que a DMS Schema Conversion não pode converter automaticamente são claramente marcados. Para concluir a migração, é possível converter esses objetos manualmente.

Em alto nível, a DMS Schema Conversion opera com os três componentes a seguir: perfis de instância, provedores de dados e projetos de migração. Um perfil de instância especifica as configurações da rede e da segurança. Um provedor de dados armazena as credenciais de conexão do banco de dados. Um projeto de migração contém provedores de dados, um perfil de instância e regras de migração. AWS DMS usa provedores de dados e um perfil de instância para criar um processo que converte esquemas de banco de dados e objetos de código.

Para obter a lista dos bancos de dados de origem compatíveis, consulte Origens para a DMS Schema Conversion.

Para obter a lista dos bancos de dados de destino compatíveis, consulte Destinos da DMS Schema Conversion.

O diagrama a seguir ilustra o processo da DMS Schema Conversion.



Utilize os tópicos a seguir para compreender melhor como utilizar a DMS Schema Conversion.

Tópicos

Suportado Regiões da AWS

Você pode criar um projeto de migração do DMS Schema Conversion da seguinte forma. Regiões da AWS Em outras regiões, é possível utilizar o AWS Schema Conversion Tool. Para obter mais informações sobre AWS SCT, consulte o Guia do usuário da AWS Schema Conversion Tool.

Nome da região	Região
Leste dos EUA (Norte da Virgínia)	us-east-1
Leste dos EUA (Ohio)	us-east-2
Oeste dos EUA (Oregon)	us-west-2
Ásia-Pacífico (Tóquio)	ap-northeast-1
Ásia-Pacífico (Singapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2
Europa (Frankfurt)	eu-central-1
Europa (Estocolmo)	eu-north-1
Europa (Irlanda)	eu-west-1

Recursos de conversão de esquema

A DMS Schema Conversion fornece os seguintes recursos:

- O DMS Schema Conversion gerencia automaticamente os Nuvem AWS recursos necessários para seu projeto de migração de banco de dados. Esses recursos incluem perfis de instância, provedores de dados e AWS Secrets Manager segredos. Eles também incluem funções AWS Identity and Access Management (IAM), buckets do Amazon S3 e projetos de migração.
- É possível utilizar a DMS Schema Conversion para se conectar ao banco de dados de origem, ler os metadados e criar relatórios de avaliação de migração de banco de dados. É possível salvar o relatório em um bucket do Amazon S3. Com esses relatórios, você obtém um resumo das tarefas de conversão de esquemas, e os detalhes dos itens que não podem ser convertidos automaticamente pela DMS Schema Conversion no banco de dados de destino. Os relatórios de avaliação de migração de banco de dados ajudam a avaliar quanto do projeto de migração a DMS Schema Conversion pode automatizar. Além disso, esses relatórios ajudam a estimar a quantidade de esforço manual necessária para concluir a conversão. Para ter mais informações, consulte Criar relatórios de avaliação de migração de banco de dados com a DMS Schema Conversion.

- Depois de se conectar aos provedores de dados de origem e de destino, a DMS Schema Conversion pode converter os esquemas de bancos de dados de origem existentes no mecanismo de banco de dados de destino. É possível escolher qualquer item do esquema do banco de dados de origem para converter. Depois de converter o código do banco de dados na DMS Schema Conversion, é possível revisar o código de origem e o código convertido. Além disso, é possível salvar o código SQL convertido em um bucket do Amazon S3.
- Antes de converter os esquemas do banco de dados de origem, é possível configurar regras de transformação. É possível utilizar as regras de transformação para alterar o tipo de dados de colunas, mover objetos de um esquema para outro e alterar os nomes de objetos. É possível aplicar as regras de transformação a bancos de dados, esquemas, tabelas e colunas. Para ter mais informações, consulte Configurar regras de transformação.
- É possível alterar as configurações de conversão para melhorar o desempenho do código convertido. Essas configurações são específicas para cada par de conversão e dependem dos recursos do banco de dados de origem que você utiliza no código. Para ter mais informações, consulte Especificar as configurações de conversão de esquemas.
- Em alguns casos, a DMS Schema Conversion não pode converter recursos de banco de dados para recursos equivalentes no Amazon RDS. Nesses casos, a DMS Schema Conversion cria um pacote de extensões no banco de dados de destino para emular os recursos que não foram convertidos. Para ter mais informações, consulte Utilizar pacotes de extensão.
- É possível aplicar o código convertido e o esquema do pacote de extensões ao banco de dados de destino. Para ter mais informações, consulte Aplicar o código convertido.
- O DMS Schema Conversion oferece suporte a todos os recursos da versão mais recente AWS SCT . Para obter mais informações, consulte As notas de versão mais recentes do AWS SCT.
- Você pode editar o código SQL convertido antes que o DMS o migre para o banco de dados de destino. Para ter mais informações, consulte Como editar e gravar seu código SQL convertido .

Limitações da conversão de esquemas

A conversão do esquema DMS é uma versão web do (). AWS Schema Conversion Tool AWS SCT A DMS Schema Conversion é compatível com menos plataformas de banco de dados e fornece funcionalidade mais limitada em comparação com a aplicação de desktop do AWS SCT . Para converter esquemas de data warehouse, estruturas de big data, código SQL de aplicações e processos ETL, utilize o AWS SCT. Para obter mais informações sobre AWS SCT, consulte o Guia do usuário da AWS Schema Conversion Tool.

As seguintes limitações se aplicam ao utilizar a DMS Schema Conversion para conversão de esquema de banco de dados:

- Não é possível salvar um projeto de migração e utilizá-lo em modo off-line.
- Você não pode editar o código SQL para a fonte em um projeto de migração para a conversão do esquema DMS. Para editar o código SQL do banco de dados de origem, utilize o editor SQL normal. Escolha Atualizar do banco de dados para adicionar o código atualizado ao projeto de migração.
- As regras de migração na DMS Schema Conversion não são compatíveis com a alteração do agrupamento de colunas. Além disso, não é possível utilizar as regras de migração para mover objetos para um novo esquema.
- Não é possível aplicar filtros às árvores dos bancos de dados de origem e de destino para exibir somente os objetos do banco de dados que atendem à cláusula de filtro.
- O pacote de extensão DMS Schema Conversion não inclui AWS Lambda funções que emulam envio de e-mail, agendamento de trabalhos e outros recursos em seu código convertido.
- O DMS Schema Conversion não usa chaves KMS gerenciadas pelo cliente para acessar nenhum recurso do cliente. AWS Por exemplo, a DMS Schema Conversion não é compatível com a utilização de uma chave do KMS gerenciada pelo cliente para acessar dados do cliente no Amazon S3.

Conceitos básicos da DMS Schema Conversion

Para começar a usar a DMS Schema Conversion, utilize o tutorial a seguir. Nele, você pode aprender a configurar a DMS Schema Conversion, criar um projeto de migração e conectar-se aos provedores de dados. É possível aprender a avaliar a complexidade da migração e converter o banco de dados de origem em um formato compatível com o banco de dados de destino. Além disso, é possível aprender a aplicar o código convertido ao banco de dados de destino.

O tutorial a seguir aborda as tarefas de pré-requisito e demonstra a conversão de um banco de dados Amazon RDS para SQL Server em Amazon RDS para MySQL. É possível utilizar qualquer um dos provedores de dados de origem e de destino compatíveis. Para ter mais informações, consulte Provedores de dados de origem para a DMS Schema Conversion.

Para obter mais informações sobre a conversão do esquema DMS, leia as instruções de step-by-step migração das migrações do Oracle para o PostgreSQL e do SQL Server para o MySQL.

Este vídeo apresenta a interface de usuário da DMS Schema Conversion e ajuda você a se familiarizar com os principais componentes desse serviço.

Tópicos

Pré-requisitos para trabalhar com a DMS Schema Conversion

Para configurar a DMS Schema Conversion, conclua as tarefas a seguir. É possível configurar um perfil de instância, adicionar provedores de dados e criar um projeto de migração.

Tópicos

Criar uma VPC com base em uma Amazon VPC

Nesta etapa, você cria uma nuvem privada virtual (VPC) no seu. Conta da AWS Essa VPC é baseada no serviço Amazon Virtual Private Cloud (Amazon VPC) e contém seus recursos. AWS

Como criar uma VPC para a DMS Schema Conversion

1. Faça login AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. Escolha Criar VPC.
3. Na página Criar VPC, insira as seguintes configurações:

- Recursos a serem criados: VPC e mais
- Geração automática da etiqueta de nome: escolha Gerar automaticamente e insira um nome globalmente exclusivo. Por exemplo, digite **`sc-vpc`**.
- IPv4 CIDR block (Bloco CIDR IPv4): **`10.0.1.0/24`**
- Gateways NAT: em uma AZ
- Endpoints da VPC: Nenhum

4. Mantenha o restante das configurações como padrão e escolha Criar VPC.
5. Escolha Sub-redes e anote os IDs das sub-redes pública e privada.

   Para conectar-se aos bancos de dados do Amazon RDS, crie um grupo de sub-redes que inclua sub-redes públicas.

   Para conectar-se aos bancos de dados on-premises, crie um grupo de sub-redes que inclua sub-redes privadas. Para ter mais informações, consulte Tarefa 1: Criar um perfil de instância.
6. Escolha Gateways NAT. Escolha o Gateway NAT e anote o Endereço IP elástico.

   Configure sua rede para garantir que ela AWS DMS possa acessar seu banco de dados local de origem a partir do endereço IP público desse gateway NAT. Para ter mais informações, consulte Utilizar uma conexão de internet com uma VPC.

Utilize essa VPC ao criar o perfil de instância e os bancos de dados de destino no Amazon RDS.

Criar um bucket do Amazon S3

Para armazenar as informações do projeto de migração, crie um bucket do Amazon S3. A DMS Schema Conversion utiliza esse bucket do Amazon S3 para salvar itens, como relatórios de avaliação, código SQL convertido, informações sobre objetos do esquema de banco de dados e assim por diante.

Como criar um bucket do Amazon S3 para a DMS Schema Conversion

1. Faça login no AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Selecione Criar bucket.
3. Na página Criar bucket, selecione um nome globalmente exclusivo para o bucket do S3. Por exemplo, digite **`sc-s3-bucket`**.

4. Em Região da AWS, escolha a região.
5. Em Versionamento de bucket, escolha Ativar.
6. Mantenha o restante das configurações como padrão e escolha Criar bucket.

Armazene as credenciais do banco de dados em AWS Secrets Manager

Armazene suas credenciais do banco de dados de origem e destino em AWS Secrets Manager. Certifique-se de replicar esses segredos para o seu Região da AWS. A DMS Schema Conversion utiliza esses segredos para conectar-se aos bancos de dados no projeto de migração.

Para armazenar suas credenciais de banco de dados em AWS Secrets Manager

1. Faça login no AWS Management Console e abra o AWS Secrets Manager console em https://console.aws.amazon.com/secretsmanager/.
2. Selecione Armazenar um novo segredo.
3. A página Escolher tipo de segredo é aberta. Em Secret type (Tipo de segredo), escolha o tipo de credenciais de banco de dados a armazenar:

   - Credenciais para o banco de dados Amazon RDS: escolha essa opção para armazenar as credenciais do banco de dados Amazon RDS. Em Credenciais, insira as credenciais do banco de dados. Em Database (Banco de dados), escolha seu banco de dados.
   - Credenciais para outro banco de dados: escolha essa opção para armazenar credenciais para os bancos de dados Oracle ou SQL Server de origem. Em Credenciais, insira as credenciais do banco de dados.
   - Outro tipo de segredo: escolha essa opção para armazenar somente o nome e a senha do usuário para conexão ao banco de dados. Escolha Adicionar linha para adicionar dois pares de chave-valor. Utilize **`username`** e **`password`** para os nomes de chaves. Para os valores relacionados a essas chaves, insira as credenciais do banco de dados.
4. Em Chave de criptografia, escolha a AWS KMS chave que o Secrets Manager usa para criptografar o valor secreto. Escolha Próximo.
5. Na página Configurar segredo, insira um Nome de segredo descritivo. Por exemplo, insira **`sc-source-secret`** ou **`sc-target-secret`**.
6. Escolha Replicar segredo e, em Região da AWS, escolha a região. Escolha Próximo.
7. Na página Configurar alternância, escolha Próximo.
8. Na página Revisar, revise os detalhes do segredo e escolha Armazenar.

Para armazenar as credenciais dos bancos de dados de origem e de destino, repita essas etapas.

Criar perfis do IAM

Crie funções AWS Identity and Access Management (IAM) para usar em seu projeto de migração. A DMS Schema Conversion utiliza esses perfis do IAM para acessar as credenciais do bucket e do banco de dados do Amazon S3 armazenadas no AWS Secrets Manager.

Como criar um perfil do IAM que fornece acesso ao bucket do Amazon S3

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles.
3. Escolha Criar Perfil.
4. Na página Selecionar tipo de entidade confiável, escolha Serviço da AWS . Escolha DMS.
5. Escolha Próximo. A página Adicionar permissões é aberta.
6. Em Políticas de filtro, insira **S3**. Escolha Amazon FullAccess S3.
7. Escolha Próximo. A página Nomear, revisar e criar é aberta.
8. Em Nome do perfil, insira um nome descritivo. Por exemplo, digite **sc-s3-role**. Selecione Criar função.
9. Na página Perfis, insira **sc-s3-role** em Nome do perfil. Escolha sc-s3-role.
10. Na página sc-s3-role, escolha a guia Relações de confiança. Escolha Editar política de confiança.
11. Na página Editar política de confiança, edite as relações de confiança do perfil para utilizar a entidade principal do serviço `schema-conversion.dms.amazonaws.com` como a entidade confiável.
12. Selecione Atualizar política de confiança.

Para criar uma função do IAM que forneça acesso a AWS Secrets Manager

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles.
3. Escolha Criar Perfil.

4. Na página Selecionar tipo de entidade confiável, escolha Serviço da AWS . Escolha DMS.
5. Escolha Próximo. A página Adicionar permissões é aberta.
6. Em Políticas de filtro, insira **`Secret`**. Escolha SecretsManagerReadWrite.
7. Escolha Próximo. A página Nomear, revisar e criar é aberta.
8. Em Nome do perfil, insira um nome descritivo. Por exemplo, digite **`sc-secrets-manager-role`**. Selecione Criar função.
9. Na página Perfis, insira **`sc-secrets-manager-role`** em Nome do perfil. Escolha sc-secrets-manager-role.
10. Na sc-secrets-manager-rolepágina, escolha a guia Relações de confiança. Escolha Editar política de confiança.
11. Na página Editar política de confiança, edite as relações de confiança para a função a ser usada `schema-conversion.dms.amazonaws.com` e seu diretor de serviço AWS DMS regional como entidades confiáveis. Esse diretor de serviço AWS DMS regional tem o seguinte formato.

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

    Substitua *`region-name`* pelo nome da sua região, como `us-east-1`.

    O exemplo de código a seguir mostra a entidade principal da região `us-east-1`.

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

    O exemplo de código a seguir mostra uma política de confiança para acessar a conversão do AWS DMS esquema.

    ```
    {
        "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. Selecione Atualizar política de confiança.

Tarefa 1: Criar um perfil de instância

Antes de criar um perfil de instância, configure um grupo de sub-redes para o perfil de instância. Para obter mais informações sobre a criação de um grupo de sub-redes para seu projeto de AWS DMS migração, consulteCriação de um grupo de sub-redes.

É possível criar um perfil de instância conforme descrito no procedimento a seguir. Nesse perfil de instância, você especifica as configurações de rede e de segurança para o projeto da DMS Schema Conversion.

Como criar um perfil de instância

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, escolha Perfis de instância e Criar perfil de instância.
3. Em Nome, insira um nome exclusivo para o perfil de instância. Por exemplo, digite **`sc-instance`**.
4. Em Tipo de rede, escolha IPv4 para criar um perfil de instância compatível somente com o endereçamento IPv4. Para criar um perfil de instância compatível com o endereçamento IPv4 e IPv6, escolha Modo de pilha dupla.
5. Em Nuvem privada virtual (VPC), escolha a VPC criada na etapa de pré-requisitos.
6. Em Grupo de sub-redes, escolha o grupo de sub-redes do perfil de instância. Para conectar-se aos bancos de dados do Amazon RDS, utilize um grupo de sub-redes que inclua sub-redes públicas. Para conectar-se aos bancos de dados on-premises, utilize um grupo de sub-redes que inclua sub-redes privadas.
7. Escolha Criar perfil de instância.

Para criar um projeto de migração, utilize esse perfil de instância.

Etapa 2: Configurar os provedores de dados

Você cria provedores de dados que descrevem os bancos de dados de origem e de destino. Para cada provedor de dados, especifique um tipo de armazenamento de dados e informações de localização. Não armazene as credenciais de banco de dados em um provedor de dados.

Como criar um provedor de dados para um banco de dados on-premises de origem

1. Faça login no AWS Management Console e abra o AWS DMS console.
2. No painel de navegação, escolha Provedores de dados e Criar provedor de dados.
3. Em Nome, insira um nome exclusivo para o provedor de dados de origem. Por exemplo, digite **`sc-source`**.
4. Em Tipo de mecanismo, escolha o tipo de mecanismo do banco de dados do provedor de dados.
5. Forneça as informações de conexão do banco de dados de origem. Os parâmetros de conexão dependem do mecanismo do banco de dados de origem. Para ter mais informações, consulte Criação de provedores de dados .
6. Em Modo Secure Socket Layer (SSL), escolha o tipo de aplicação SSL.
7. Escolha Criar provedor de dados.

Como criar um provedor de dados para um banco de dados de destino do Amazon RDS

1. Faça login no AWS Management Console e abra o AWS DMS console.
2. No painel de navegação, escolha Provedores de dados e Criar provedor de dados.
3. Em Configuração, escolha Instância do banco de dados RDS.
4. Em Banco de dados no RDS, escolha Procurar e escolha o banco de dados. A DMS Schema Conversion recupera automaticamente as informações sobre o tipo de mecanismo, o nome do servidor e a porta.
5. Em Nome, insira um nome exclusivo para o provedor de dados de destino. Por exemplo, digite **`sc-target`**.
6. Em Database name (Nome do banco de dados), insira o nome do banco de dados.
7. Em Modo Secure Socket Layer (SSL), escolha o tipo de aplicação SSL.
8. Escolha Criar provedor de dados.

Etapa 3: Criar um projeto de migração

Agora é possível criar um projeto de migração. No projeto de migração, você especifica os provedores de dados de origem e de destino e o perfil de instância.

Como criar um projeto de migração

1. Escolha Projetos de migração e Criar projeto de migração.
2. Em Nome, insira um nome exclusivo para o projeto de migração. Por exemplo, digite **`sc-project`**.
3. Em Perfil de instância, escolha **`sc-instance`**.
4. Em Origem, escolha Procurar e **`sc-source`**.
5. Para ID do segredo, escolha **`sc-source-secret`**.
6. Em IAM role (Função do IAM), escolha **`sc-secrets-manager-role`**.
7. Em Origem, escolha Procurar e **`sc-target`**.
8. Para ID do segredo, escolha **`sc-target-secret`**.
9. Em IAM role (Função do IAM), escolha **`schema-conversion-role`**.
10. Escolha Criar projeto de migração.

Etapa 4: Criar um relatório de avaliação

Para avaliar a complexidade da migração, crie o relatório de avaliação da migração do banco de dados. Esse relatório inclui a lista de todos os objetos de banco de dados que a DMS Schema Conversion não pode converter automaticamente.

Para criar um relatório de avaliação

1. Escolha Projetos de migração e **`sc-project`**.
2. Escolha Conversão de esquemas e Iniciar conversão de esquemas.
3. No painel do banco de dados de origem, escolha o esquema de banco de dados a ser avaliado. Além disso, marque a caixa de seleção do nome desse esquema.
4. No painel do banco de dados de origem, escolha Avaliar no menu Ações. A caixa de diálogo Avaliar é exibida.
5. Escolha Avaliar na caixa de diálogo para confirmar a opção.

A guia Resumo mostra o número de itens que a DMS Schema Conversion pode converter automaticamente em objetos de armazenamento de banco de dados e objetos de código de banco de dados.

6. Escolha Itens de ação para ver a lista de todos os objetos de banco de dados que a DMS Schema Conversion não pode converter automaticamente. Analise as ações recomendadas para cada item.
7. Para salvar uma cópia do relatório de avaliação, escolha Exportar resultados. Escolha um dos seguintes formatos: CSV ou PDF. A caixa de diálogo Exportar é exibida.
8. Escolha Exportar para confirmar a opção.
9. Escolha Bucket do S3. O console do Amazon S3 é aberto.
10. Escolha Baixar para salvar o relatório de avaliação.

Etapa 5: Converter o código-fonte

É possível converter o esquema de banco de dados de origem utilizando o procedimento a seguir. É possível salvar o esquema convertido como scripts SQL em um arquivo de texto.

Como converter o esquema do banco de dados

1. No painel do banco de dados de origem, escolha o esquema do banco de dados a ser convertido. Além disso, marque a caixa de seleção do nome desse esquema.
2. No painel do banco de dados de origem, escolha Converter no menu Ações. A caixa de diálogo Converter é exibida.
3. Escolha Converter na caixa de diálogo para confirmar a opção.
4. Escolha um objeto do banco de dados no painel do banco de dados de origem. A DMS Schema Conversion exibe o código-fonte e o código convertido para esse objeto. Você pode editar o código SQL convertido para um objeto de banco de dados usando o recurso Editar SQL. Para ter mais informações, consulte Como editar e gravar seu código SQL convertido .
5. No painel do banco de dados de destino, escolha o esquema do banco de dados convertido. Além disso, marque a caixa de seleção do nome desse esquema.
6. Em Ações, escolha Salvar como SQL. A caixa de diálogo Salvar é exibida.
7. Escolha Salvar como SQL para confirmar a opção.
8. Escolha Bucket do S3. O console do Amazon S3 é aberto.
9. Escolha Baixar para salvar seus scripts SQL.

Etapa 6: Aplicar o código convertido

A DMS Schema Conversion não aplica imediatamente o código convertido ao banco de dados de destino. Para atualizar seu banco de dados de destino, é possível utilizar os scripts SQL criados na etapa anterior. Como alternativa, utilize o procedimento a seguir para aplicar o código convertido na DMS Schema Conversion.

Como aplicar o código convertido

1. No painel do banco de dados de destino, escolha o esquema do banco de dados convertido. Além disso, marque a caixa de seleção do nome desse esquema.
2. Em Ações, escolha Aplicar alterações. A caixa de diálogo Aplicar alterações é exibida.
3. Escolha Aplicar para confirmar sua escolha.

Etapa 7: Limpeza e solução de problemas

Você pode usar CloudWatch a Amazon para revisar ou compartilhar seus registros de conversão do esquema DMS.

Para analisar os logs da DMS Schema Conversion

1. Faça login no AWS Management Console e abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Logs, Grupos de logs.

   O nome do grupo de logs da DMS Schema Conversion começa com `dms-tasks-sct`. É possível classificar os grupos de logs por Tempo de criação para encontrar o grupo de logs da DMS Schema Conversion.

   Além disso, o nome do grupo de logs inclui o nome do recurso da Amazon (ARN) do projeto de migração. É possível ver o ARN do projeto na página de Projetos de migração na DMS Schema Conversion. Escolha o ARN em Preferências.
3. Escolha o nome do seu grupo de logs e escolha o nome do fluxo de logs.
4. Em Ações, escolha Exportar resultados para salvar o log da DMS Schema Conversion.

Depois de concluir a conversão de esquemas na DMS Schema Conversion, limpe os recursos.

Como limpar os recursos da DMS Schema Conversion

1. Faça login no AWS Management Console e abra o AWS DMS console.
2. No painel de navegação, escolha Projetos de migração.
    a. Selecione **`sc-project`**.
    b. Escolha Conversão de esquemas e Fechar conversão de esquemas.
    c. Escolha Excluir e confirme a opção.
3. No painel de navegação, escolha Perfis de instância.
    a. Selecione **`sc-instance`**.
    b. Escolha Excluir e confirme a opção.
4. No painel de navegação, escolha Provedores de dados.
    a. Selecione **`sc-source`** e **`sc-target`**.
    b. Escolha Excluir e confirme a opção.

Além disso, certifique-se de limpar outros AWS recursos que você criou, como seu bucket do Amazon S3, segredos do banco de dados, funções do IAM e nuvem privada virtual (VPC). AWS Secrets Manager

Configurar uma rede para a DMS Schema Conversion

A DMS Schema Conversion cria uma instância de conversão de esquemas em uma nuvem privada virtual (VPC) com base no serviço Amazon VPC. Ao criar o perfil da instância, especifique a VPC a ser utilizada. É possível utilizar a VPC padrão da sua conta e Região da AWS ou criar uma nova VPC.

É possível utilizar diferentes configurações de rede para configurar a interação dos bancos de dados de origem e de destino com a DMS Schema Conversion. Essas configurações dependem da localização do provedor de dados de origem e das configurações de rede. Os tópicos a seguir fornecem descrições de configurações de rede comuns.

Tópicos

- Utilizar um AWS Direct Connect ou uma VPN para configurar uma rede para uma VPC
- Utilizar uma conexão de internet com uma VPC
- Utilizar um ambiente sem um gateway da Internet

Utilizar uma única VPC para provedores de dados de origem e de destino

A configuração de rede mais simples para a DMS Schema Conversion é a configuração de uma única VPC. Aqui, o provedor de dados de origem, o perfil da instância e o provedor de dados de destino estão todos localizados na mesma VPC. É possível utilizar essa configuração para converter o banco de dados de origem em uma instância do Amazon EC2.

Para utilizar essa configuração, verifique se o grupo de segurança da VPC utilizado pelo perfil de instância tem acesso aos provedores de dados. Por exemplo, é possível permitir um intervalo da VPC de Encaminhamento Entre Domínios Sem Classificação (CIDR) ou o endereço IP elástico para o gateway de conversão de endereços de rede (NAT).

Utilizar várias VPCs para provedores de dados de origem e de destino

Se os provedores de dados de origem e de destino estiverem em VPCs diferentes, é possível criar o perfil de instância em uma das VPCs. Vincular essas duas VPCs utilizando o emparelhamento de VPC. É possível utilizar essa configuração para converter o banco de dados de origem em uma instância do Amazon EC2.

Uma Conexão de emparelhamento da VPC é uma conexão de redes entre duas VPCs, que permite rotear utilizando os endereços IP privados de cada VPC como se estivessem na mesma rede. É possível criar uma conexão de emparelhamento da VPC entre suas próprias VPCs, com uma VPC em outra conta da AWS ou com uma VPC em uma Região da AWS diferente. Para obter mais informações sobre Emparelhamento de VPC, consulte o Guia do usuário da Amazon VPC.

Para implementar o emparelhamento de VPC, siga as instruções em Trabalhar com conexões de emparelhamento da VPC no Guia do usuário da Amazon VPC. Verifique se a tabela de rotas de uma VPC contém o bloco CIDR da outra. Por exemplo, suponha que a VPC A esteja utilizando o destino 10.0.0.0/16 e a VPC B esteja utilizando o destino 172.31.0.0. Nesse caso, a tabela de rotas da VPC A deve conter 172.31.0.0 e a tabela de rotas da VPC B deve conter 10.0.0.0/16. Para obter informações mais detalhadas, consulte Atualizar as tabelas de rotas para a conexão de emparelhamento da VPC no Guia do usuário de emparelhamento da Amazon VPC.

Utilizar um AWS Direct Connect ou uma VPN para configurar uma rede para uma VPC

Redes remotas podem se conectar a uma VPC utilizando várias opções, como o AWS Direct Connect ou uma conexão VPN de software ou de hardware. É possível utilizar essas opções para integrar os serviços existentes no local, estendendo uma rede interna para a Nuvem AWS. É possível integrar serviços locais, como monitoramento, autenticação, segurança, dados ou outros sistemas. A utilização desse tipo de extensão de rede facilita a conexão a recursos no local com recursos hospedados pela AWS, como uma VPC. É possível utilizar essa configuração para converter o banco de dados on-premises de origem.

Nessa configuração, o grupo de segurança da VPC deve incluir uma regra de roteamento que envia o tráfego destinado a um intervalo de CIDR de VPC ou a um endereço IP específico para um host. Esse host deve ser capaz de superar o tráfego da VPC na VPN on-premises. Nesse caso, o host NAT inclui suas próprias configurações de grupo de segurança. Essas configurações devem permitir tráfego do intervalo CIDR da VPC ou grupo de segurança da VPC para a instância NAT. Para obter mais informações, consulte Criar uma conexão VPN site a site no Guia do usuário do AWS Site-to-Site VPN.

Utilizar uma conexão de internet com uma VPC

Se você não utiliza uma VPN ou o AWS Direct Connect para conecta-se a recursos da AWS, poderá utilizar uma conexão com a internet. Essa configuração envolve uma sub-rede privada em uma VPC com um gateway da Internet. O gateway contém o provedor de dados de destino e o perfil de instância. É possível utilizar essa configuração para converter o banco de dados on-premises de origem.

Para adicionar um gateway da Internet à sua VPC, consulte Anexar um gateway da Internet, no Guia do usuário da Amazon VPC.

A tabela de rotas da VPC deve incluir regras de roteamento que enviem tráfego não destinado à VPC por padrão ao gateway da Internet. Nessa configuração, a conexão com o provedor de dados parece vir do endereço IP público do gateway NAT. Para obter mais informações, consulte Tabelas de rotas da VPC no Guia do usuário da Amazon VPC.

Utilizar um ambiente sem um gateway da Internet

Para criar um ambiente para conversão de esquemas sem utilizar um gateway da Internet, faça o seguinte.

1. Siga as etapas 1 a 3 do tutorial Conceitos básicos com as seguintes alterações:
   - Escolha sub-redes privadas em vez de públicas.
   - Durante a criação da instância, em Atribuir IP público, escolha Não.
2. Abra o console da Amazon VPC.
3. Escolha Endpoints e Criar endpoint.
4. Na página Criar endpoint, faça o seguinte:
   - Em Categoria do serviço, selecione Serviços da AWS.
   - Na lista Serviços, escolha com.amazonaws.***{region}***.secretsmanage
   - Na seção VPC, escolha a VPC criada.
   - Escolha as sub-redes da VPC.
   - Escolha o grupo de segurança da VPC.
   - Em Política, deixe a opção Acesso total selecionada.
5. Conclua o restante do Conceitos básicos tutorial.

Criar provedores de dados de origem na DMS Schema Conversion

Você pode usar um banco de dados Microsoft SQL Server, Oracle ou PostgreSQL como provedor de dados de origem em projetos de migração para DMS Schema Conversion. O provedor de dados de origem pode ser um mecanismo autogerenciado em execução on-premises ou em uma instância do Amazon EC2.

Configure a rede para permitir a interação entre o provedor de dados de origem e a DMS Schema Conversion. Para ter mais informações, consulte Configurar uma rede para a DMS Schema Conversion.

Tópicos

- Utilizar um banco de dados Microsoft SQL Server como origem na DMS Schema Conversion
- Utilizar um banco de dados Oracle como origem na DMS Schema Conversion
- Utilizar um banco de dados Oracle Data Warehouse como origem no DMS Schema Conversion
- Usando um banco de dados PostgreSQL como fonte na conversão de esquema do DMS
- Usando um banco de dados MySQL como fonte na conversão de esquema do DMS

Utilizar um banco de dados Microsoft SQL Server como origem na DMS Schema Conversion

É possível utilizar bancos de dados SQL Server como origem na DMS Schema Conversion.

É possível utilizar a DMS Schema Conversion para converter objetos de código de banco de dados SQL Server para os seguintes destinos:

- Aurora MySQL
- Aurora PostgreSQL
- RDS para MySQL
- RDS para PostgreSQL.

Para obter informações sobre as versões compatíveis do banco de dados SQL Server, consulte Provedores de dados de origem para a DMS Schema Conversion.

Para obter mais informações sobre como usar a conversão de esquema DMS com um banco de dados SQL Server de origem, consulte o passo a passo da migração do SQL Server para o MySQL. step-by-step

Privilégios do Microsoft SQL Server como origem

Veja a seguir a lista de privilégios obrigatórios para o Microsoft SQL Server como origem:

- VIEW DEFINITION
- VIEW DATABASE STATE

O privilégio `VIEW DEFINITION` permite que usuários com acesso público vejam as definições de objetos. A DMS Schema Conversion utiliza o privilégio `VIEW DATABASE STATE` para verificar os recursos da edição SQL Server Enterprise.

Repetir a concessão para cada banco de dados cujo esquema que você está convertendo.

Além disso, conceda os seguintes privilégios no banco de dados `master`:

- VIEW SERVER STATE
- VIEW ANY DEFINITION

A DMS Schema Conversion utiliza o privilégio `VIEW SERVER STATE` para coletar as definições e as configurações do servidor. Conceda o privilégio `VIEW ANY DEFINITION` para visualizar os provedores de dados.

Para ler as informações sobre o Microsoft Analysis Services, execute o comando a seguir no banco de dados `master`.

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

No exemplo anterior, substitua espaço reservado *<user_name>* pelo nome do usuário a quem você concedeu os privilégios necessários anteriormente.

Para ler informações sobre o SQL Server Agent, adicione seu usuário à AgentUser função SQL. Execute o comando a seguir no banco de dados `msdb`.

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

No exemplo anterior, substitua o espaço reservado *<SQLAgentRole>* pelo nome do perfil do SQL Server Agent. Substitua espaço reservado *<user_name>* pelo nome do usuário a quem você concedeu os privilégios necessários anteriormente. Para obter mais informações, consulte Adicionar um usuário à AgentUser função SQL no Guia do usuário do Amazon RDS.

Para detectar o envio de logs, conceda o privilégio `SELECT on dbo.log_shipping_primary_databases` no banco de dados `msdb`.

Para utilizar a abordagem de notificação da replicação da linguagem de definição de dados (DDL), conceda o privilégio `RECEIVE ON` *<schema_name>*`.`*<queue_name>* nos bancos de dados de origem. Neste exemplo, substitua o espaço reservado *<schema_name>* pelo nome do esquema do banco de dados. Substitua o espaço reservado *<queue_name>* pelo nome de uma tabela de filas.

Utilizar um banco de dados Oracle como origem na DMS Schema Conversion

É possível utilizar bancos de dados Oracle como origem de migração na DMS Schema Conversion.

Para conectar-se ao banco de dados Oracle, utilize o ID do sistema (SID) Oracle. Para encontrar o Oracle SID, envie a consulta a seguir para seu banco de dados Oracle:

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

É possível utilizar a DMS Schema Conversion para converter objetos de código de banco de dados do Oracle para os seguintes destinos:

- Aurora MySQL
- Aurora PostgreSQL
- RDS para MySQL
- RDS para PostgreSQL.

Para obter informações sobre as versões compatíveis do banco de dados Oracle, consulte Provedores de dados de origem para a DMS Schema Conversion.

Para obter mais informações sobre como usar a conversão de esquema DMS com um banco de dados Oracle de origem, consulte o passo a passo da migração de Oracle para PostgreSQL. step-by-step

Privilégios do Oracle como origem

Os privilégios obrigatórios para o Oracle como origem são listados a seguir:

- CONECTAR
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- SELECT ON SYS.ARGUMENT$

Utilizar um banco de dados Oracle Data Warehouse como origem no DMS Schema Conversion

Você pode usar bancos de dados Oracle Data Warehouse como origem de migração no DMS Schema Conversion para converter objetos de código de banco de dados e código de aplicação no Amazon Redshift.

Para obter informações sobre versões compatíveis do banco de dados Oracle, consulte Provedores de dados de origem para a DMS Schema Conversion. Para obter mais informações sobre como usar a conversão de esquema DMS com um banco de dados Oracle de origem, consulte o passo a passo da migração de Oracle para PostgreSQL. step-by-step

Privilégios para usar o banco de dados Oracle Data Warehouse como origem

Os seguintes privilégios são obrigatórios para o Oracle Data Warehouse como origem:

- CONECTAR
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY

Configurações de conversão do Oracle Data Warehouse para o Amazon Redshift

Para obter mais informações sobre o DMS Schema Conversion, consulte Especificar as configurações de conversão de esquemas para projetos de migração.

As configurações de conversão do Oracle Data Warehouse para o Amazon Redshift incluem as seguintes opções:

- Adicionar comentários no código convertido para os itens de ação da gravidade selecionada ou superior: essa configuração limita o número de comentários com itens de ação no código convertido. O DMS adiciona comentários no código convertido para itens de ação da gravidade selecionada e superior.

  Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.
- O número máximo de tabelas para o cluster do Amazon Redshift de destino: essa configuração define o número de tabelas que o DMS pode aplicar cluster do Amazon Redshift de destino. O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Essa configuração é compatível com os seguintes valores:
  - Auto: o DMS determinará o número de tabelas a serem aplicadas ao cluster do Amazon Redshift de destino, dependendo do tipo de nó.
  - Definir um valor: defina o número de tabelas manualmente.

  O DMS converte todas as tabelas de origem, mesmo que o número de tabelas seja maior do que o cluster do Amazon Redshift pode armazenar. O DMS armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, o DMS exibirá uma mensagem de aviso. Além disso, o DMS aplica tabelas ao cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

Para obter informações sobre cotas da tabela do Amazon Redshift, consulte Cotas e limites no Amazon Redshift.

- Usar a visualização UNION ALL: essa configuração permite definir o número máximo de tabelas de destino que o DMS pode criar para uma única tabela de origem.

  O Amazon Redshift não oferece suporte ao particionamento de tabelas. Para emular o particionamento de tabelas e acelerar a execução de consultas, o DMS pode migrar cada partição da tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, o DMS cria uma visualização que inclui dados das tabelas de destino que ele cria.

  O DMS determina automaticamente o número de partições na tabela de origem. Dependendo do tipo de particionamento da tabela de origem, esse número pode exceder a cota das tabelas que você pode aplicar ao seu cluster do Amazon Redshift. Para evitar atingir essa cota, insira o número máximo de tabelas de destino que o DMS pode criar para partições de uma única tabela de origem. A opção padrão é 368 tabelas, que representam uma partição para 366 dias do ano e mais duas tabelas para partições NON RANGE e UNKNOWN.
- Os elementos de formato de tipo de data que você usa no código Oracle são semelhantes às strings de formato de data e hora no Amazon Redshift: use essa configuração para converter funções de formatação de tipo de dados como `TO_CHAR`, `TO_DATE` e `TO_NUMBER` com elementos de formato de data e hora que o Amazon Redshift não aceita. Por padrão, o DMS usa funções do pacote de extensões para emular esses elementos de formato não aceitos no código convertido.

  O modelo de formato de data e hora no Oracle inclui mais elementos do que as strings de formato de data e hora no Amazon Redshift. Quando o código-fonte incluir somente elementos de formato de data e hora compatíveis com o Amazon Redshift, defina esse valor para evitar funções do pacote de extensões no código convertido. Quando as funções de extensão são evitadas, o código convertido é executado mais rapidamente.
- Os elementos de formato numérico que você usa no código Oracle são semelhantes às strings de formato numérico no Amazon Redshift: use essa configuração para converter funções de formatação de tipo de dados numéricos como que o Amazon Redshift não aceita. Por padrão, o DMS usa funções do pacote de extensões para emular esses elementos de formato não aceitos no código convertido.

  O modelo de formato numérico no Oracle inclui mais elementos do que as strings de formato numérico no Amazon Redshift. Quando o código-fonte incluir somente elementos de formato numérico compatíveis com o Amazon Redshift, defina esse valor para evitar funções do pacote

de extensões no código convertido. Quando as funções de extensão são evitadas, o código convertido é executado mais rapidamente.

- Usar a função NVL para emular o comportamento das funções LEAD e LAG do Oracle: se o código-fonte não usar os valores padrão para deslocamento nas funções `LAG` e `LEAD`, o DMS poderá emular essas funções com a função `NVL`. Por padrão, o DMS gera um item de ação para cada uso das funções `.LEAD` e `LAG`. Quando essas funções são emuladas com `NVL`, o código convertido é executado mais rapidamente.
- Emular o comportamento das chaves primárias e exclusivas: defina essa configuração para que o DMS emule o comportamento das restrições de chave primária e exclusiva no cluster do Amazon Redshift de destino. O Amazon Redshift não impõe restrições de chave primária e exclusiva e as utiliza apenas para fins informativos. Se o código-fonte usar restrições de chave primária ou exclusiva, defina essa configuração para garantir que o DMS emule o respectivo comportamento.
- Usar a codificação de compactação: defina essa configuração para aplicar a codificação de compactação às colunas da tabela do Amazon Redshift. O DMS atribui automaticamente a codificação de compactação às colunas usando o algoritmo padrão do Redshift. Para obter mais informações sobre codificação de compactação, consulte Codificações de compactação no Guia do desenvolvedor de banco de dados do Amazon Redshift.

  Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. Para aplicar compactação a essas colunas, defina Usar codificação de compactação para colunas de CHAVE. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Usando um banco de dados PostgreSQL como fonte na conversão de esquema do DMS

Você pode usar bancos de dados PostgreSQL como fonte de migração no DMS Schema Conversion.

Você pode usar a Conversão de Esquema DMS para converter objetos de código de banco de dados do banco de dados PostgreSQL para os seguintes destinos:

- MySQL
- Aurora MySQL

Os privilégios obrigatórios para PostgreSQL como origem são listados a seguir:

- 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>

Usando um banco de dados MySQL como fonte na conversão de esquema do DMS

Você pode usar bancos de dados MySQL como fonte de migração no DMS Schema Conversion.

Você pode usar a Conversão de Esquema DMS para converter objetos de código de banco de dados do banco de dados MySQL para os seguintes destinos:

- PostgreSQL
- Aurora PostgreSQL

Os privilégios obrigatórios do MySQL como origem são listados a seguir:

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

Configurações de conversão de MySQL para PostgreSQL

Para obter mais informações sobre o DMS Schema Conversion, consulte Especificar as configurações de conversão de esquemas para projetos de migração.

As configurações de conversão de MySQL para PostgreSQL incluem o seguinte:

- Comentários no código SQL convertido: defina essa configuração para adicionar comentários no código convertido para os itens de ação da severidade selecionada e superior.

  Valores válidos:

  - Somente erros
  - Erros e advertências
  - Todas as mensagens

Criar provedores de dados de destino no DMS Schema Conversion

É possível utilizar os bancos de dados MySQL e PostgreSQL como provedor de dados de destino em projetos de migração para o DMS Schema Conversion. O provedor de dados de destino pode ser uma instância do Amazon EC2, do Amazon RDS ou do Amazon Aurora.

Tópicos

Utilizar um banco de dados MySQL como destino no DMS Schema Conversion

É possível utilizar bancos de dados MySQL como destino da migração no DMS Schema Conversion.

Para obter informações sobre os bancos de dados de destino compatíveis, consulte Provedores de dados de destino para a DMS Schema Conversion.

Privilégios do MySQL como banco de dados de destino

Os privilégios a seguir são obrigatórios para o MySQL como destino:

- `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.*

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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';
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Em seguida, substitua *your_password* por uma senha segura.

Para usar o Amazon RDS para MySQL ou o Aurora MySQL como destino, defina o parâmetro `lower_case_table_names` como 1. Esse valor significa que o servidor MySQL manipula

identificadores de nomes de objetos como tabelas, índices, acionadores e bancos de dados sem distinção entre maiúsculas e minúsculas. Se você ativou o registro binário em sua instância de destino, defina o parâmetro `log_bin_trust_function_creators` como `1`. Nesse caso, você não precisa usar as características `DETERMINISTIC`, `READS SQL DATA` ou `NO SQL` para criar funções armazenadas. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Utilizar um banco de dados PostgreSQL como destino no DMS Schema Conversion

É possível utilizar bancos de dados PostgreSQL como destino de migração no DMS Schema Conversion.

Para obter informações sobre os bancos de dados de destino compatíveis, consulte Provedores de dados de destino para a DMS Schema Conversion.

Privilégios para o PostgreSQL como banco de dados de destino

Para utilizar o PostgreSQL como destino, o DMS Schema Conversion requer o privilégio `CREATE ON DATABASE`. Crie um usuário e conceda a esse usuário esse privilégio para cada banco de dados que deseja utilizar no projeto de migração do DMS Schema Conversion.

Para utilizar o Amazon RDS para PostgreSQL como destino, o DMS Schema Conversion requer o perfil `rds_superuser`.

Para utilizar os sinônimos públicos convertidos, altere o caminho de pesquisa padrão do banco de dados utilizando o comando a seguir.

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

Neste exemplo, substitua o espaço reservado *<db_name>* pelo nome do banco de dados.

No PostgreSQL, apenas o proprietário do esquema ou um `superuser` pode descartar um esquema. O proprietário pode descartar um esquema e todos os objetos incluídos nesse esquema, mesmo que o proprietário do esquema não possua alguns de seus objetos.

Ao utilizar usuários diferentes para converter e aplicar esquemas diferentes ao banco de dados de destino, é possível encontrar uma mensagem de erro quando o DMS Schema Conversion não pode eliminar um esquema. Para evitar essa mensagem de erro, utilize o perfil `superuser`.

Usar um cluster do Amazon Redshift como destino no DMS Schema Conversion

É possível utilizar bancos de dados do Amazon Redshift como destino no DMS Schema Conversion. Para obter informações sobre os bancos de dados de destino compatíveis, consulte Provedores de dados de destino para a DMS Schema Conversion.

Privilégios para o Amazon Redshift como destino

O uso do Amazon Redshift como destino para o DMS Schema Conversion requer os seguintes privilégios:

- CREATE ON DATABASE: permite que o DMS crie esquemas no banco de dados.
- CREATE ON SCHEMA: permite que o DMS crie objetos no esquema do banco de dados.
- GRANT USAGE ON LANGUAGE: permite que o DMS crie funções e procedimentos no banco de dados.
- GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog: fornece ao usuário informações do sistema sobre o cluster Amazon Redshift.
- GRANT SELECT ON pg_class_info: fornece ao usuário informações sobre o estilo de distribuição da tabela.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder permissões. Substitua os valores de exemplo por seus próprios valores.

```
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;
```

Repita a operação `GRANT CREATE ON SCHEMA` para cada esquema de destino em que você aplicará o código convertido ou migrará os dados.

Você pode aplicar um pacote de extensão em seu banco de dados de destino do Amazon Redshift. Um pacote de extensões é um módulo complementar que emula perfis de banco de dados de origem necessários ao converter objetos para o Amazon Redshift. Para obter mais informações, consulte Utilizar pacotes de extensão na DMS Schema Conversion.

Gerenciar projetos de migração no DMS Schema Conversion

Depois de criar um perfil de instância e provedores de dados compatíveis para a conversão de esquemas, crie um projeto de migração. Para obter mais informações, consulte Criar projetos de migração.

Para utilizar esse novo projeto no DMS Schema Conversion, na página Projetos de migração, escolha o projeto na lista. Na guia Conversão de esquemas, escolha Iniciar conversão de esquemas.

A primeira inicialização do DMS Schema Conversion requer alguma configuração. O AWS Database Migration Service (AWS DMS) inicia uma instância de conversão de esquemas, que leva até 15 minutos. Esse processo também lê os metadados dos bancos de dados de origem e de destino. Depois de uma primeira inicialização bem-sucedida, é possível acessar o DMS Schema Conversion mais rapidamente.

A Amazon encerra a instância de conversão de esquemas que o projeto de migração utiliza em três dias após a conclusão do projeto. É possível recuperar o esquema convertido e o relatório de avaliação no bucket do Amazon S3 utilizado para o DMS Schema Conversion.

Especificar as configurações do projeto de migração para o DMS Schema Conversion

Depois de criar o projeto de migração e iniciar a conversão de esquemas, é possível especificar as configurações do projeto de migração. É possível alterar configurações da conversão para melhorar o desempenho do código convertido. Além disso, é possível personalizar a visualização da conversão de esquemas.

As configurações da conversão dependem das plataformas de banco de dados de origem e de destino. Para obter mais informações, consulte Criar provedores de dados de origem e Criar provedores de dados de destino.

Para especificar quais esquemas e bancos de dados você deseja ver nos painéis dos bancos de dados de origem e de destino, utilize as configurações de visualização em árvore. É possível ocultar

esquemas vazios, bancos de dados vazios, bancos de dados de sistema e esquemas e bancos de dados definidos pelo usuário.

Para ocultar bancos de dados e esquemas na visualização em árvore

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e, na guia Conversão de esquemas, escolha Iniciar conversão de esquemas.
4. Escolha Configurações. A página Configurações é aberta.
5. Na seção Visualização em árvore, faça o seguinte:

   - Escolha Ocultar esquemas vazios para ocultar esquemas vazios.
   - Escolha Ocultar bancos de dados vazios para ocultar bancos de dados vazios.
   - Em Bancos de dados ou esquemas do sistema, escolha os esquemas e bancos de dados do sistema por nome para ocultá-los.
   - Em Bancos de dados ou esquemas definidos pelo usuário, insira os nomes dos esquemas e dos bancos de dados definidos pelo usuário que você deseja ocultar. Escolha Adicionar. Os nomes não diferenciam maiúsculas de minúsculas.

     Para adicionar vários bancos de dados ou esquemas, utilize uma vírgula para separar seus nomes. Para adicionar vários objetos com um nome semelhante, utilize a porcentagem (%) como curinga. Esse curinga substitui qualquer número de quaisquer símbolos no nome do banco de dados ou do esquema.

   Repita essas etapas para as seções Origem e Destino.
6. Escolha Aplicar e Conversão de esquemas.

Criar relatórios de avaliação de migração de banco de dados com a DMS Schema Conversion

Uma parte importante da DMS Schema Conversion é o relatório que ele gera para ajudar você a converter seu esquema. Esse relatório de avaliação de migração de banco de dados resume todas as tarefas de conversão de esquemas. Também detalha os itens de ação do esquema que

não podem ser convertidos no mecanismo de banco de dados da instância de banco de dados de destino. É possível visualizar o relatório no console do AWS DMS ou salvar uma cópia desse relatório como um arquivo PDF ou como um arquivo de valores separados por vírgulas (CSV).

O relatório de avaliação de migração inclui:

- Um resumo executivo
- Recomendações, incluindo a conversão de objetos de servidor, sugestões de backup e alterações de servidor vinculado

Quando há itens que a DMS Schema Conversion não pode converter automaticamente, o relatório fornece estimativas da quantidade de esforço necessária para escrever o código equivalente para a instância do banco de dados de destino.

Tópicos

Criar um relatório de avaliação de migração de banco de dados

Depois de criar um projeto de migração, utilize o procedimento a seguir para criar um relatório de avaliação de migração de banco de dados.

Para criar um relatório de avaliação de migração de banco de dados

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e escolha Conversão de esquemas.
4. Escolha Iniciar conversão de esquemas. A página Conversão de esquemas é aberta.
5. No painel do banco de dados de origem, escolha o esquema de banco de dados ou os itens do esquema que você deseja avaliar. Para incluir vários objetos no relatório, selecione todos os itens.

6. Depois de marcar as caixas de seleção para todos os objetos do esquema que você deseja avaliar, escolha o nó pai para os objetos selecionados. O menu Ações no painel do banco de dados de origem é disponibilizado.
7. Escolha Avaliar no menu Ações. Uma caixa de diálogo de confirmação é exibida.
8. Escolha Avaliar na caixa de diálogo para confirmar a opção.

Visualizar o relatório de avaliação de migração de banco de dados

Depois de criar um relatório de avaliação, a DMS Schema Conversion adiciona informações nas seguintes guias:

- Resumo
- Itens de ação

A guia Resumo mostra o número de itens que a DMS Schema Conversion pode converter automaticamente.

A guia Itens de ação mostra os itens que a DMS Schema Conversion não pode converter automaticamente e fornece recomendações sobre como gerenciar esses itens.

Resumo do relatório de avaliação

A guia Resumo exibe as informações de resumo do relatório de avaliação de migração do banco de dados. Ele mostra o número de itens que a DMS Schema Conversion pode converter automaticamente em objetos de armazenamento de banco de dados e em objetos de código de banco de dados.

Na maioria dos casos, a DMS Schema Conversion não pode converter automaticamente todos os itens do esquema no mecanismo de banco de dados de destino. O Resumo também inclui uma estimativa do esforço necessário para criar os itens do esquema na instância do banco de dados de destino equivalentes aos do banco de dados de origem.

Para ver o resumo da conversão de objetos de armazenamento de banco de dados, como tabelas, sequências, restrições, tipos de dados etc., escolha Objetos de armazenamento do banco de dados.

Para ver o resumo da conversão de objetos de código de banco de dados, como procedimentos, perfis, visualizações, acionadores etc., escolha Objetos de código do banco de dados.

Para alterar o escopo do relatório de avaliação, selecione o nó necessário na árvore do banco de dados de origem. A DMS Schema Conversion atualiza o resumo do relatório de avaliação para que corresponda ao escopo selecionado.

Itens de ação do relatório de avaliação

A guia Itens de ação contém uma lista de itens que a DMS Schema Conversion não pode converter automaticamente em um formato compatível com o mecanismo do banco de dados de destino. Para cada item de ação, a DMS Schema Conversion fornece a descrição do problema e a ação recomendada. A DMS Schema Conversion agrupa itens de ação semelhantes e exibe o número de ocorrências.

Para visualizar o código do objeto do banco de dados relacionado, selecione um item de ação na lista.

Salvar o relatório de avaliação de migração do banco de dados

Depois de criar um relatório de avaliação de migração de banco de dados, é possível salvar uma cópia desse relatório como um arquivo PDF ou como arquivo de valores separados por vírgula (CSV).

Como salvar o relatório de avaliação de migração de banco de dados como um arquivo PDF

1. Escolha Exportar e PDF. Examine a caixa de diálogo e escolha Exportar para PDF.
2. A DMS Schema Conversion cria um arquivo PDF e armazena-o no bucket do Amazon S3. Para alterar o bucket do Amazon S3, edite as configurações de conversão de esquema no perfil de instância.
3. Abra o arquivo do relatório de avaliação no bucket do Amazon S3.

Como salvar o relatório de avaliação de migração de banco de dados como arquivos CSV

1. Escolha Exportar e CSV. Examine a caixa de diálogo e escolha Exportar para CSV.
2. A DMS Schema Conversion cria um arquivamento com arquivos CSV e armazena esse arquivamento no bucket do Amazon S3. Para alterar o bucket do Amazon S3, edite as configurações de conversão de esquema no perfil de instância.
3. Abra os arquivos do relatório de avaliação no bucket do Amazon S3.

O arquivo PDF contém as informações do resumo e dos itens de ação.

Ao exportar o relatório de avaliação para CSV, a DMS Schema Conversion cria três arquivos CSV.

O primeiro arquivo CSV contém as seguintes informações sobre os itens de ação:

- Categoria
- Ocorrência
- Item de ação
- Assunto
- Grupo
- Descrição
- Referências da documentação
- Ação recomendada
- Linha
- Posição
- Origem
- Destino
- Endereço IP e porta do servidor
- Banco de dados
- Esquema

O segundo arquivo CSV inclui o sufixo dos `Action_Items_Summary` em seu nome e contém as seguintes informações:

- Esquema
- Item de ação
- Número de ocorrências
- Esforços da curva de aprendizado, que é a quantidade de esforço necessária para criar uma abordagem para converter cada item de ação
- Esforços para converter uma ocorrência do item de ação, que mostra o esforço necessário para converter cada item de ação, seguindo a abordagem projetada
- Descrição do item de ação
- Ação recomendada

Os valores que indicam o nível dos esforços necessários são baseados em uma escala ponderada, que varia de baixo (mínimo) a alto (máximo).

O segundo arquivo CSV inclui o `Summary` em seu nome e contém as seguintes informações:

- Categoria
- Número de objetos
- Objetos convertidos automaticamente
- Objetos com ações simples
- Objetos com ações de complexidade média
- Objetos com ações complexas
- Total de linhas de código

Utilizar a DMS Schema Conversion

A DMS Schema Conversion converte os esquemas de banco de dados existentes e a maioria dos objetos de código do banco de dados em um formato compatível com o banco de dados de destino.

A DMS Schema Conversion automatiza grande parte do processo de conversão dos esquemas de banco de dados de processamento de transações on-line (OLTP) em Amazon RDS para MySQL ou RDS para PostgreSQL. Os mecanismos de banco de dados de origem e de destino contêm diversos recursos e capacidades, e a DMS Schema Conversion tentará criar um esquema equivalente na instância sempre que possível. Para objetos de banco de dados em que a conversão direta não é possível, a DMS Schema Conversion fornece uma lista de ações a serem executadas.

Para converter o esquema do banco de dados, utilize o seguinte processo:

- Antes de converter os esquemas de banco de dados, configure regras de transformação que alteram os nomes dos objetos do banco de dados durante a conversão.
- Crie um relatório de avaliação da migração do banco de dados para estimar a complexidade da migração. Esse relatório fornece detalhes sobre os elementos do esquema que a DMS Schema Conversion não pode converter automaticamente.
- Converta o armazenamento do banco de dados de origem e os objetos de código. A DMS Schema Conversion cria uma versão local dos objetos de banco de dados convertidos. É possível acessar esses objetos convertidos em seu projeto de migração.

- Salve o código convertido em arquivos SQL para revisar, editar ou endereçar itens de ação de conversão. Opcionalmente, aplique o código convertido diretamente no banco de dados de destino.

Para converter esquemas de data warehouse, use o desktop AWS Schema Conversion Tool. Para obter mais informações, consulte Converter esquemas de data warehouse para o Amazon Redshift no Guia do usuário da AWS Schema Conversion Tool.

Tópicos

- Configurar regras de transformação na DMS Schema Conversion
- Converter esquemas de banco de dados na DMS Schema Conversion
- Especificar as configurações de conversão de esquemas para projetos de migração
- Atualizar os esquemas de banco de dados no DMS Schema Conversion
- Salvar e aplicar o código convertido na DMS Schema Conversion

Configurar regras de transformação na DMS Schema Conversion

Antes de converter o esquema do banco de dados com a DMS Schema Conversion, é possível configurar as regras de transformação. As Regras de transformação podem fazer coisas, como alterar o nome de um objeto para minúsculas ou maiúsculas, adicionar ou remover um prefixo ou sufixo e renomear objetos. Por exemplo, suponha que você tenha um conjunto de tabelas no esquema de origem chamado `test_TABLE_NAME`. É possível configurar uma regra que altera o prefixo `test_` para o prefixo `demo_` no esquema de destino.

É possível criar regras de transformação que executam as seguintes tarefas:

- Adicionar, remover ou substituir um prefixo
- Adicionar, remover ou substituir um sufixo
- Alterar o tipo de dados de uma coluna
- Alterar o nome do objeto para minúsculas ou maiúsculas
- Renomear objetos

É possível criar regras de transformação para os seguintes objetos:

- Schema

- Tabela
- Coluna

Criar regras de transformação

A DMS Schema Conversion armazena regras de transformação como parte do projeto de migração. É possível configurar regras de transformação ao criar o projeto de migração ou editá-las posteriormente.

É possível adicionar várias regras de transformação no projeto. A DMS Schema Conversion aplica as regras de transformação durante a conversão na mesma ordem em que são adicionadas.

Como criar regras de transformação

1. Na página Criar projeto de migração, escolha Adicionar regra de transformação. Para ter mais informações, consulte Criar projetos de migração.
2. Em Destino da regra, escolha o tipo de objetos de banco de dados aos quais essa regra se aplica.
3. Em Esquema de origem, escolha Inserir um esquema. Insira os nomes dos esquemas, tabelas e colunas de origem aos quais essa regra se aplica. É possível inserir um nome exato para selecionar um objeto ou inserir um padrão para selecionar vários objetos. Utilize a porcentagem (%) como curinga para substituir qualquer número de quaisquer símbolos no nome do objeto do banco de dados.
4. Em Ação, escolha a tarefa a ser executada.
5. Dependendo do tipo de regra, insira um ou dois valores adicionais. Por exemplo, para renomear um objeto, insira o novo nome do objeto. Para substituir um prefixo, insira o prefixo antigo e o novo prefixo.
6. Escolha Adicionar regra de transformação para adicionar outra regra de transformação.

   Ao concluir a adição de regras, escolha Criar projeto de migração.

Para duplicar uma regra de transformação existente, escolha Duplicar. Para editar uma regra de transformação existente, escolha a regra na lista. Para excluir uma regra de transformação existente, escolha Remover.

Editar regras de transformação

É possível adicionar novas, remover ou editar regras de transformação existentes no projeto de migração. Como a DMS Schema Conversion aplica as regras de transformação durante a inicialização da conversão de esquemas, feche a conversão de esquemas e inicie-a novamente depois de editar as regras.

Como editar regras de transformação

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração e escolha o projeto de migração.
3. Escolha Conversão de esquemas e Fechar conversão de esquemas.
4. Depois de AWS DMS fechar a conversão do esquema, escolha Modificar para editar as configurações do projeto de migração.
5. Em Regas de transformação, escolha uma das seguintes ações:
   - Escolha Duplicar para duplicar uma regra de transformação existente e adicioná-la ao final da lista.
   - Escolha Remover para remover uma regra de transformação existente.
   - Escolha a regra de transformação existente para editá-la.
6. Ao concluir a edição das regras, escolha Salvar alterações.
7. Na página Projetos de migração, escolha o projeto na lista. Escolha Conversão de esquemas e Iniciar conversão de esquemas.

Converter esquemas de banco de dados na DMS Schema Conversion

Depois de criar o projeto de migração e de conectar-se aos bancos de dados de origem e de destino, é possível converter os objetos do banco de dados de origem em um formato compatível com o banco de dados de destino. A DMS Schema Conversion exibe o esquema do banco de dados de origem no painel esquerdo em formato de visualização em árvore.

Cada nó da árvore de bancos de dados é carregado lentamente. Quando você escolhe um nó na visualização em árvore, a DMS Schema Conversion solicita as informações do esquema do banco de dados de origem naquele momento. Para carregar as informações do esquema mais rapidamente, escolha o esquema e escolha Carregar metadados no menu Ações. A DMS Schema

Conversion lê os metadados do banco de dados e armazena as informações em um bucket do Amazon S3. Agora é possível procurar os objetos do banco de dados com mais rapidez.

É possível converter todo o esquema do banco de dados ou escolher qualquer item do esquema do banco de dados de origem para converter. Se o item do esquema escolhido depender de um item pai, a DMS Schema Conversion também gerará o esquema para o item pai. Por exemplo, quando você escolhe uma tabela para converter, a DMS Schema Conversion cria a tabela convertida e o esquema de banco de dados em que a tabela está.

Converter objetos de banco de dados

É possível utilizar a DMS Schema Conversion para converter um esquema de banco de dados inteiro ou objetos de esquema de banco de dados separados.

Como converter um esquema de banco de dados inteiro

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e escolha Conversão de esquemas.
4. Escolha Iniciar conversão de esquemas. A página Conversão de esquemas é aberta.
5. No painel do banco de dados de origem, marque a caixa de seleção do nome do esquema.
6. Escolha esse esquema no painel esquerdo do projeto de migração. A DMS Schema Conversion destaca o nome do esquema em azul e ativa o menu Ações.
7. Em Ações, escolha Converter. A caixa de diálogo de conversão é exibida.
8. Escolha Converter na caixa de diálogo para confirmar a opção.

Como converter objetos do banco de dados de origem

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e escolha Conversão de esquemas.
4. Escolha Iniciar conversão de esquemas. A página Conversão de esquemas é aberta.
5. No painel do banco de dados de origem, selecione os objetos do banco de dados de origem.

6. Depois de marcar as caixas de seleção dos objetos que você deseja converter, escolha o nó pai dos objetos selecionados no painel esquerdo.

   A DMS Schema Conversion destaca o nó pai em azul e ativa o menu Ações.
7. Em Ações, escolha Converter. A caixa de diálogo de conversão é exibida.
8. Escolha Converter na caixa de diálogo para confirmar a opção.

Por exemplo, para converter duas de 10 tabelas, marque as caixas de seleção das duas tabelas que você deseja converter. Observe que o menu Ações está inativo. Depois de escolher o nó Tabelas, a DMS Schema Conversion destaca seu nome em azul e ativa o menu Ações. É possível escolher Converter nesse menu.

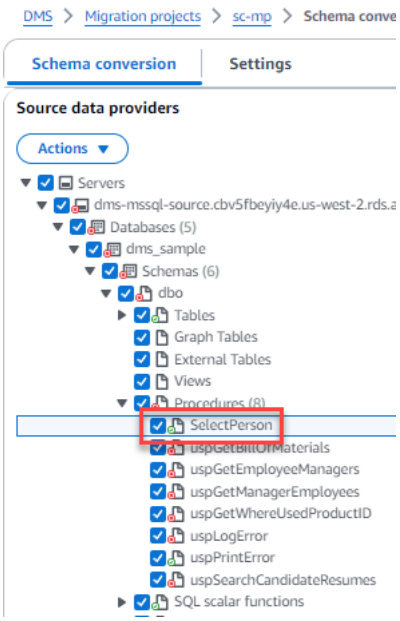
Da mesma forma, para converter duas tabelas e três procedimentos, marque as caixas de seleção dos nomes dos objetos. Escolha o nó do esquema para ativar o menu Ações e escolha Converter esquema.

Como editar e gravar seu código SQL convertido

A página de conversão de esquema permite que você edite o código SQL convertido em seus objetos de banco de dados. Use o procedimento a seguir para editar seu código de SQL convertido, aplicar as alterações e, em seguida, salvá-los.

Para editar, aplicar alterações e salvar o código SQL convertido

1. Na página Conversão de esquema, abra a visualização em árvore no painel Provedores de dados de origem para exibir um objeto de código.



2. No painel Provedores de dados de origem, escolha Ações, Converter. Confirme a ação.
3. Quando a conversão for concluída, para visualizar o SQL convertido, expanda o painel central, se necessário. Para editar o SQL convertido, escolha o ícone de edição no painel SQL de destino.

4. Depois de editar o SQL de destino, confirme suas alterações escolhendo o ícone de verificação na parte superior da página. Confirme a ação.
5. No painel Provedores de dados de destino, escolha Ações, Aplicar alterações. Confirme a ação.
6. O DMS grava o procedimento editado no armazenamento de dados de destino.

Revisar objetos convertidos do banco de dados

Depois de converter os objetos do banco de dados de origem, é possível escolher um objeto no painel esquerdo do projeto. É possível visualizar a origem e o código convertido desse objeto. A DMS Schema Conversion carrega automaticamente o código convertido para o objeto selecionado no painel esquerdo. Você também pode ver as propriedades ou os parâmetros do objeto selecionado.

A DMS Schema Conversion armazena automaticamente o código convertido como parte do projeto de migração. Ele não aplica essas alterações de código ao banco de dados de destino. Para obter mais informações sobre como aplicar o código convertido ao banco de dados de destino, consulte Aplicar o código convertido. Para remover o código convertido do projeto de migração, selecione o esquema de destino no painel direito e escolha Atualizar do banco de dados em Ações.

Depois de converter os objetos do banco de dados de origem, é possível ver o resumo da conversão e os itens de ação no painel central inferior. É possível ver as mesmas informações ao criar um relatório de avaliação. O relatório de avaliação é útil para identificar e resolver itens de esquema que o DMS Schema Conversion não pode converter. É possível salvar o resumo do relatório de avaliação e a lista de itens de ação da conversão em arquivos CSV. Para ter mais informações, consulte Relatórios de avaliação de migração de banco de dados.

Especificar as configurações de conversão de esquemas para projetos de migração

Depois de criar um projeto de migração, é possível especificar as configurações de conversão na DMS Schema Conversion. É possível alterar as configurações de conversão de esquemas para melhorar o desempenho do código convertido.

Como editar as configurações de conversão

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração. Escolha Conversão de esquemas e Iniciar conversão de esquemas.
4. Escolha Configurações. A página Configurações é aberta.
5. Na seção Conversão, altere as configurações.
6. Escolha Aplicar e Conversão de esquemas.

Para todos os pares de conversão, é possível limitar o número de comentários com itens de ação no código convertido. Para limitar o número de comentários no código convertido, abra as configurações de conversão no projeto de migração.

Em Comentários no código SQL convertido, escolha o nível de gravidade dos itens de ação. A DMS Schema Conversion adiciona comentários no código convertido para itens de ação com a gravidade selecionada e superior. Por exemplo, para minimizar o número de comentários no código convertido, escolha Somente erros.

Para incluir comentários para todos os itens de ação no código convertido, escolha Todas as mensagens.

Outras configurações de conversão são diferentes para cada par de bancos de dados de origem e de destino.

Tópicos

Configurações de conversão do Oracle para MySQL

As configurações de conversão do Oracle para MySQL na DMS Schema Conversion incluem o seguinte:

- O banco de dados Oracle de origem pode utilizar a pseudocoluna `ROWID`. O MySQL não é compatível com uma funcionalidade semelhante. A DMS Schema Conversion pode emular a pseudocoluna `ROWID` no código convertido. Para fazer isso, ative a opção Gerar ID de linha.

  Se o código do Oracle de origem não utilizar a pseudocoluna `ROWID`, desative a opção Gerar ID de linha. Nesse caso, o código convertido funciona mais rapidamente.
- O código do Oracle de origem pode incluir os perfis `TO_CHAR`, `TO_DATE` e `TO_NUMBER` com parâmetros que não são compatíveis com o MySQL. Por padrão, a DMS Schema Conversion emula a utilização desses parâmetros no código convertido.

É possível utilizar os perfis nativos do MySQL `TO_CHAR`, `TO_DATE` e `TO_NUMBER` quando o código do Oracle de origem não tem parâmetros que não são compatíveis com o MySQL. Nesse caso, o código convertido funciona mais rapidamente. Para isso, selecione os seguintes valores:

- Utilize um perfil TO_CHAR nativo do MySQL
- Utilize um perfil TO_DATE nativo do MySQL
- Utilize um perfil TO_NUMBER nativo do MySQL

- O banco de dados e as aplicações podem ser executados em fusos horários diferentes. Por padrão, a DMS Schema Conversion emula os fusos horários no código convertido. No entanto, essa emulação não é necessária quando o banco de dados e as aplicações utilizam o mesmo fuso horário. Nesse caso, selecione Melhorar o desempenho do código convertido quando o banco de dados e as aplicações utilizam o mesmo fuso horário.

Configurações da conversão do Oracle para o PostgreSQL

As configurações de conversão do Oracle para PostgreSQL na DMS Schema Conversion incluem o seguinte:

- AWS DMS pode converter visualizações materializadas Oracle em tabelas ou visualizações materializadas no PostgreSQL. Para Visões materializadas, escolha como converter as visões materializadas de origem.
- O banco de dados Oracle de origem pode utilizar a pseudocoluna `ROWID`. O PostgreSQL não é compatível com funcionalidade semelhante. A DMS Schema Conversion pode emular a `ROWID` pseudocoluna no código convertido utilizando o tipo de dados `bigint` ou `character varying`. Para fazer isso, escolha Utilizar o tipo de dados bigint para emular a pseudocoluna ROWID ou Utilizar o tipo de dados com variação de caracteres para emular a pseudocoluna ROWID em ID da linha.

  Se o código do Oracle de origem não utilizar a pseudocoluna `ROWID`, escolha Não gerar. Nesse caso, o código convertido funciona mais rapidamente.
- O código do Oracle de origem pode incluir os perfis `TO_CHAR`, `TO_DATE` e `TO_NUMBER` com parâmetros que não são compatíveis com o PostgreSQL. Por padrão, a DMS Schema Conversion emula a utilização desses parâmetros no código convertido.

  É possível utilizar perfis nativos do PostgreSQL, `TO_CHAR`, `TO_DATE` e `TO_NUMBER`, quando o código do Oracle de origem não tem parâmetros que não são compatíveis com o PostgreSQL.

Nesse caso, o código convertido funciona mais rapidamente. Para isso, selecione os seguintes valores:

- Utilizar um perfil TO_CHAR nativo do PostgreSQL
- Utilizar um perfil TO_DATE nativo do PostgreSQL
- Utilizar um perfil TO_NUMBER nativo do PostgreSQL

- O banco de dados e as aplicações podem ser executados em fusos horários diferentes. Por padrão, a DMS Schema Conversion emula os fusos horários no código convertido. No entanto, essa emulação não é necessária quando o banco de dados e as aplicações utilizam o mesmo fuso horário. Nesse caso, selecione Melhorar o desempenho do código convertido quando o banco de dados e as aplicações utilizam o mesmo fuso horário.
- Para continuar utilizando sequências no código convertido, selecione Preencher sequências convertidas com o último valor gerado no lado da origem.
- Em alguns casos, o banco de dados Oracle de origem pode armazenar somente valores inteiros nas colunas de chave primária ou estrangeira do tipo de dados `NUMBER`. Nesses casos, AWS DMS pode converter essas colunas para o tipo de `BIGINT` dados. Essa abordagem melhorará o desempenho do código convertido. Para fazer isso, selecione Converter colunas de chave primária e estrangeira do tipo de dados NUMBER para o tipo de dados BIGINT. Verifique se a origem não inclui valores de ponto flutuante nessas colunas para evitar perda de dados.
- Para ignorar acionadores e restrições desativados no código da origem, escolha Converter somente acionadores e restrições ativos.
- É possível utilizar a DMS Schema Conversion para converter variáveis de sequência de caracteres chamadas de SQL dinâmico. O código de banco de dados pode alterar os valores dessas variáveis de string. Para garantir que AWS DMS sempre converta o valor mais recente dessa variável de string, selecione Converter o código SQL dinâmico criado nas rotinas chamadas.
- As versões 10 e anteriores do PostgreSQL não são compatíveis com procedimentos. Se você não estiver familiarizado com o uso de procedimentos no PostgreSQL AWS DMS , pode converter procedimentos Oracle em funções do PostgreSQL. Para fazer isso, selecione Converter procedimentos em perfis.
- Para ver informações adicionais sobre os itens de ação ocorridos, é possível adicionar perfis específicos ao pacote de extensão. Para fazer isso, selecione Adicionar funções do pacote de extensão que geram exceções definidas pelo usuário. Escolha os níveis de gravidade para aumentar as exceções definidas pelo usuário. Não deixe de aplicar o pacote de extensão depois de converter os objetos do banco de dados de origem. Para obter mais informações sobre a extensão, consulte Utilizar pacotes de extensão.

- O banco de dados Oracle de origem pode incluir restrições com nomes gerados automaticamente. Se o código-fonte utilizar esses nomes, selecione Manter os nomes das restrições geradas pelo sistema. Se o código-fonte utilizar essas restrições, mas não utilizar seus nomes, desmarque essa opção para aumentar a velocidade da conversão.
- Se o bancos de dados de origem e de destino forem executados em fusos horários diferentes, o perfil que emula o perfil do Oracle integrado `SYSDATE` retornará valores diferentes em comparação com o perfil de origem. Para garantir que os perfis de origem e de destino retornem os mesmos valores, escolha Definir o fuso horário do banco de dados de origem.
- É possível utilizar os perfis da extensão orafce no código convertido. Para fazer isso, em Rotinas integradas do Orafce, selecione os perfis a serem utilizados. Para obter mais informações sobre orafce, consulte orafce on. GitHub

Configurações de conversão do SQL Server para MySQL

As configurações de conversão do SQL Server para o MySQL na DMS Schema Conversion incluem o seguinte:

- O banco de dados SQL Server de origem pode armazenar a saída de `EXEC` em uma tabela. A DMS Schema Conversion cria tabelas temporárias e um procedimento adicional para emular esse recurso. Para utilizar essa emulação, selecione Criar rotinas adicionais para lidar com conjuntos de dados abertos.

Configurações de conversão do SQL Server para o PostgreSQL

As configurações de conversão do SQL Server para PostgreSQL na DMS Schema Conversion incluem o seguinte:

- No SQL Server, é possível utilizar índices com o mesmo nome em tabelas diferentes. No entanto, no PostgreSQL, todos os nomes de índices utilizados no esquema devem ser exclusivos. Para garantir que a DMS Schema Conversion gere nomes exclusivos para todos os índices, selecione Gerar nomes exclusivos para índices.
- As versões 10 e anteriores do PostgreSQL não são compatíveis com procedimentos. Se você não estiver familiarizado com o uso de procedimentos no PostgreSQL AWS DMS , pode converter procedimentos do SQL Server em funções do PostgreSQL. Para fazer isso, selecione Converter procedimentos em perfis.

- O banco de dados SQL Server de origem pode armazenar a saída de EXEC em uma tabela. A DMS Schema Conversion cria tabelas temporárias e um procedimento adicional para emular esse recurso. Para utilizar essa emulação, selecione Criar rotinas adicionais para lidar com conjuntos de dados abertos.
- É possível definir o modelo a ser utilizado para os nomes dos esquemas no código convertido. Em Nomes de esquemas, selecione uma das seguintes opções:
  - DB: utiliza o nome do banco de dados SQL Server como o nome de um esquema no PostgreSQL.
  - ESQUEMA: utiliza o nome do esquema do SQL Server como o nome de um esquema no PostgreSQL.
  - DB_SCHEMA: utiliza uma combinação dos nomes do banco de dados e do esquema do SQL Server como o nome de um esquema no PostgreSQL.
- É possível manter maiúsculas e minúsculas dos nomes dos objetos de origem. Para evitar a conversão de nomes de objetos em minúsculas, selecione Manter nomes de objetos em maiúsculas ou minúsculas como estão. Essa opção se aplica somente ao ativar a opção de diferenciação de maiúsculas e minúsculas no banco de dados de destino.
- É possível manter os nomes dos parâmetros do banco de dados de origem. A DMS Schema Conversion pode adicionar aspas duplas aos nomes dos parâmetros no código convertido. Para fazer isso, selecione Manter nomes de parâmetros originais.
- Você pode manter uma série de parâmetros de rotina do banco de dados de origem. O DMS Schema Conversion cria domínios e os utiliza para especificar um tamanho para os parâmetros de rotina. Para isso, selecione Preservar tamanho dos parâmetros.

Configurações de conversão de PostgreSQL para MySQL

As configurações de conversão de PostgreSQL para MySQL na Conversão de Esquema DMS incluem o seguinte:

- Comentários no código SQL convertido: essa configuração inclui comentários no código convertido para os itens de ação da severidade selecionada e superiores. Essa configuração é compatível com os seguintes valores:
  - Errors Only (Somente erros)
  - Erros e advertências
  - Todas as mensagens

Configurações de conversão do DB2 for z/OS para DB2 LUW

As configurações de conversão do DB2 for z/OS para DB2 LUW na Conversão do Esquema DMS incluem o seguinte:

- Comentários no código SQL convertido: essa configuração inclui comentários no código convertido para os itens de ação da severidade selecionada e superiores. Essa configuração é compatível com os seguintes valores:
  - Errors Only (Somente erros)
  - Erros e advertências
  - Todas as mensagens

Atualizar os esquemas de banco de dados no DMS Schema Conversion

Depois de criar um projeto de migração, a DMS Schema Conversion armazena as informações sobre os esquemas de origem e de destino nesse projeto. A DMS Schema Conversion utiliza Carregamento lento para carregar metadados somente quando necessário, como quando você escolhe um nó na árvore de bancos de dados. É possível utilizar o carregamento rápido para carregar as informações do esquema mais rapidamente. Para fazer isso, escolha o esquema e Carregar metadados em Ações.

Depois de carregar de forma automática ou manual o objeto no projeto de migração, a DMS Schema Conversion não utiliza o carregamento lento novamente. Portanto, ao alterar objetos, como tabelas e procedimentos no banco de dados, atualize-os no projeto de migração.

Para atualizar esquemas no banco de dados, selecione os objetos que você deseja atualizar e escolha Atualizar do banco de dados em Ações. É possível atualizar objetos do banco de dados nos esquemas de bancos de dados de origem e de destino:

- Origem: se você atualizar o esquema do banco de dados de origem, escolha Atualizar no banco de dados para substituir o esquema no projeto pelo esquema mais recente no banco de dados de origem.
- Destino: se você atualizar o esquema do banco de dados de destino, a DMS Schema Conversion substituirá o esquema no projeto pelo esquema mais recente no banco de dados de destino. A DMS Schema Conversion substitui o código convertido pelo código no banco de dados de destino. Aplique o código convertido ao banco de dados de destino antes de escolher Atualizar no banco de dados. Caso contrário, converta o esquema do banco de dados de origem novamente.

Salvar e aplicar o código convertido na DMS Schema Conversion

Depois que a DMS Schema Conversion converte os objetos do banco de dados de origem, ele não aplica imediatamente o código convertido ao banco de dados de destino. Em vez disso, a DMS Schema Conversion armazena o código convertido no projeto, até que você esteja pronto para aplicá-lo no banco de dados de destino.

Antes de aplicar o código convertido, é possível atualizar o código do banco de dados de origem e converter os objetos atualizados novamente para abordar os itens de ação existentes. Para obter mais informações sobre itens que a DMS Schema Conversion não pode converter automaticamente, consulte Criar relatórios de avaliação de migração de banco de dados com a DMS Schema Conversion. Para obter mais informações sobre como atualizar os objetos do banco de dados de origem no projeto de migração da DMS Schema Conversion, consulte Atualizar os esquemas de banco de dados.

Em vez de aplicar o código convertido diretamente ao banco de dados na DMS Schema Conversion, é possível salvar o código em um arquivo, como um script SQL. É possível revisar esses scripts SQL, editá-los quando necessário e aplicá-los manualmente ao banco de dados de destino.

Salvar o código convertido em um arquivo SQL

É possível salvar o esquema convertido como scripts SQL em um arquivo de texto. É possível modificar o código convertido para abordar itens de ação que a DMS Schema Conversion não pode converter automaticamente. É possível executar os scripts SQL atualizados no banco de dados de destino para aplicar o código convertido ao banco de dados de destino.

Como salvar o esquema convertido como scripts SQL

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e escolha Conversão de esquemas.
4. Escolha Iniciar conversão de esquemas. A página Conversão de esquemas é aberta.
5. No painel direito, escolha o esquema do banco de dados de destino ou selecione os objetos convertidos que deseja salvar. Verifique se a DMS Schema Conversion destaca o nome do nó pai em azul e ativa o menu Ações para o banco de dados de destino.
6. Escolha Salvar como SQL em Ações. A caixa de diálogo Salvar é exibida.
7. Escolha Salvar como SQL para confirmar a opção.

A DMS Schema Conversion cria um arquivamento com arquivos SQL e armazena esse arquivamento no bucket do Amazon S3.

8. (Opcional) Altere o bucket do S3 para o arquivamento editando as configurações de conversão de esquema no perfil de instância.
9. Abra os scripts SQL no bucket do S3.

Aplicar o código convertido

Quando estiver pronto para aplicar o código convertido ao banco de dados de destino, escolha os objetos do banco de dados no painel à direita do projeto. É possível aplicar alterações no esquema de um banco de dados inteiro ou em objetos selecionados do esquema do banco de dados.

Depois de selecionar os objetos do banco de dados, a DMS Schema Conversion destaca o nome do nó selecionado ou do nó pai em azul. Ela ativa o menu Ações. Escolha Aplicar alterações em Ações. Na caixa de diálogo exibida, escolha Aplicar para confirmar a opção e aplicar o código convertido ao banco de dados de destino.

Aplicar o esquema do pacote de extensão

Ao aplicar o esquema convertido ao banco de dados de destino pela primeira vez, a DMS Schema Conversion também pode aplicar o esquema do pacote de extensão. O esquema do pacote de extensão emula os perfis do sistema do banco de dados de origem necessários para executar o código convertido no banco de dados de destino. Se o código convertido utilizar os perfis do pacote de extensão, aplique o esquema do pacote de extensão.

Para aplicar manualmente o pacote de extensão ao banco de dados de destino, escolha Aplicar alterações em Ações. Na caixa de diálogo exibida, selecione Confirmar para aplicar o pacote de extensão ao banco de dados de destino.

É recomendável modificar o esquema do pacote de extensão para evitar resultados inesperados no código convertido.

Para ter mais informações, consulte Utilizar pacotes de extensão na DMS Schema Conversion.

Utilizar pacotes de extensão na DMS Schema Conversion

Um pacote de extensão no DMS Schema Conversion é um módulo complementar que emula perfis de banco de dados de origem que não são compatíveis com o banco de dados de destino. Utilize

um pacote de extensão para garantir que o código convertido produza os mesmos resultados que o código-fonte. Para poder instalar o pacote de extensão, converta os esquemas de banco de dados.

Cada pacote de extensão inclui um esquema de banco de dados. Esse esquema inclui perfis, procedimentos, tabelas e visualizações do SQL para emular objetos específicos de processamento de transações on-line (OLTP) ou com perfis integrados incompatíveis com o banco de dados de origem.

Quando você converte o banco de dados de origem, a DMS Schema Conversion adiciona um esquema ao banco de dados de destino. Esse esquema implementa os perfis do sistema SQL do banco de dados de origem necessárias para executar o código convertido no banco de dados de destino. Esse esquema adicional é chamado de esquema do pacote de extensões.

O esquema do pacote de extensões é nomeado de acordo com seu banco de dados de origem da seguinte forma:

- Microsoft SQL Server – `aws_sqlserver_ext`
- Oracle – `aws_oracle_ext`

É possível aplicar o pacote de extensão de duas formas:

- A DMS Schema Conversion pode aplicar automaticamente um pacote de extensão quando você aplica o código convertido. A DMS Schema Conversion aplica o pacote de extensão antes de aplicar todos os outros objetos de esquema.
- É possível aplicar um pacote de extensão manualmente. Para fazer isso, escolha o esquema do pacote de extensão na árvore do banco de dados de destino e escolha Aplicar e Aplicar pacote de extensão.

Migrando bancos de dados para seus equivalentes do Amazon RDS com AWS DMS

As migrações de dados homogêneas em AWS Database Migration Service (AWS DMS) simplificam a migração de bancos de dados locais autogerenciados para seus equivalentes do Amazon Relational Database Service (Amazon RDS). Por exemplo, é possível utilizar migrações de dados homogêneas para migrar um banco de dados PostgreSQL on-premises para o Amazon RDS para PostgreSQL ou para o Aurora PostgreSQL. Para migrações de dados homogêneas, AWS DMS usa ferramentas de banco de dados nativas para fornecer migrações fáceis e eficientes. like-to-like

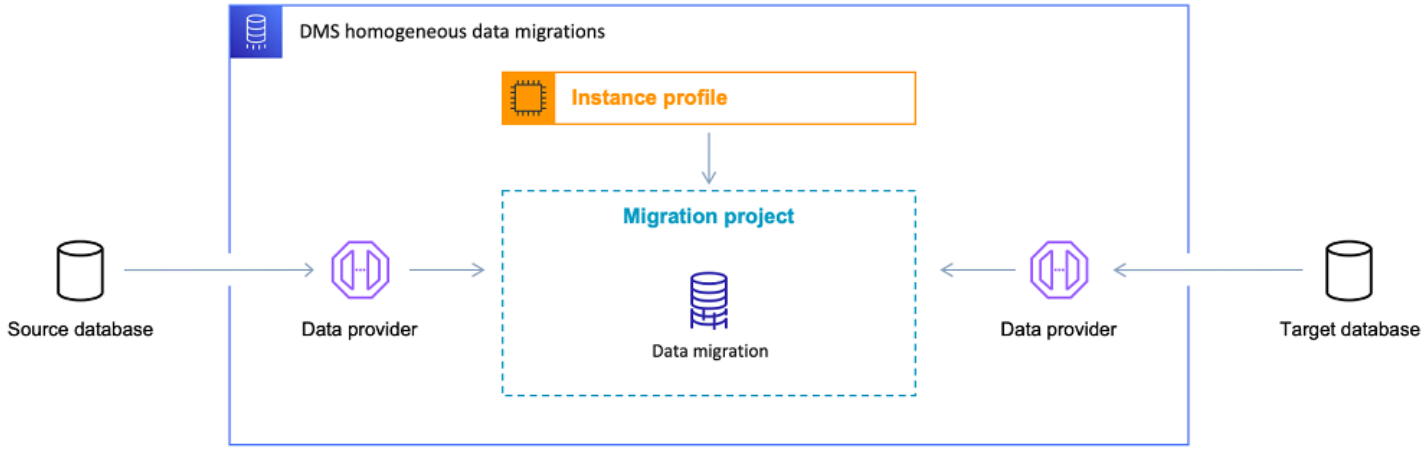
As migrações de dados homogêneas são sem servidor, o que significa que escalam AWS DMS automaticamente os recursos necessários para sua migração. Com migrações de dados homogêneas, é possível migrar dados, partições de tabelas, tipos de dados e objetos secundários, como perfis, procedimentos armazenados e assim por diante.

Em alto nível, as migrações de dados homogêneas operam com perfis de instância, provedores de dados e projetos de migração. Quando você cria um projeto de migração com os provedores de dados de origem e destino compatíveis do mesmo tipo, AWS DMS implanta um ambiente sem servidor no qual sua migração de dados é executada. Em seguida, AWS DMS conecta-se ao provedor de dados de origem, lê os dados de origem, despeja os arquivos no disco e restaura os dados usando ferramentas de banco de dados nativas. Para obter mais informações sobre perfis de instância, provedores de dados e projetos de migração, consulte Trabalhando com provedores de dados, perfis de instância e projetos de migração no AWS DMS.

Para obter a lista dos bancos de dados de origem compatíveis, consulte Origens para migrações de dados homogêneas do DMS.

Para obter a lista dos bancos de dados de destino compatíveis, consulte Destinos para migrações de dados homogêneas do DMS.

O diagrama a seguir mostra como funcionam as migrações de dados homogêneas.



As seções a seguir fornecem informações sobre o uso de migrações de dados homogêneas.

Tópicos

- Suportado Regiões da AWS
- Atributos
- Limitações de migrações de dados homogêneas
- Visão geral do processo de migração de dados homogênea no AWS DMS
- Configurar migrações de dados homogêneas no AWS DMS
- Criação de provedores de dados de origem para migrações de dados homogêneas no AWS DMS
- Criação de provedores de dados de destino para migrações de dados homogêneas no AWS DMS
- Executando migrações de dados homogêneas em AWS DMS
- Solução de problemas para migrações de dados homogêneas no AWS DMS

Suportado Regiões da AWS

Você pode executar migrações de dados homogêneas no seguinte. Regiões da AWS

Nome da região	Região
Leste dos EUA (Norte da Virgínia)	us-east-1
Leste dos EUA (Ohio)	us-east-2
Oeste dos EUA (Oregon)	us-west-2

Nome da região	Região
Ásia-Pacífico (Tóquio)	ap-northeast-1
Ásia-Pacífico (Singapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2
Europa (Frankfurt)	eu-central-1
Europa (Estocolmo)	eu-north-1
Europa (Irlanda)	eu-west-1

Atributos

As migrações de dados homogêneas fornecem os seguintes recursos:

- AWS DMS gerencia automaticamente os recursos de computação e armazenamento necessários para Nuvem AWS migrações de dados homogêneas. AWS DMS implanta esses recursos em um ambiente sem servidor quando você inicia uma migração de dados.
- AWS DMS usa ferramentas de banco de dados nativas para iniciar uma migração totalmente automatizada entre os bancos de dados do mesmo tipo.
- É possível utilizar migrações de dados homogêneas para migrar dados, bem como objetos secundários, como partições, perfis, procedimentos armazenados e assim por diante.
- É possível executar migrações de dados homogêneas nos três modos de migração a seguir: carga máxima, replicação contínua e carga máxima com replicação contínua.
- Para migrações de dados homogêneas, é possível utilizar bancos de dados on-premises, o Amazon EC2 e o Amazon RDS como a origem. É possível escolher o Amazon RDS ou o Amazon Aurora como o destino para migrações de dados homogêneas.

Limitações de migrações de dados homogêneas

As limitações a seguir se aplicam ao utilizar migrações de dados homogêneas:

- As migrações de dados homogêneas oferecem suporte somente às regras de seleção para migrações do MongoDB e do Amazon DocumentDB. O DMS não oferece suporte a regras de

seleção para outros mecanismos de banco de dados. Além disso, não é possível configurar regras que alteram o tipo de dados de colunas, movem objetos de um esquema para outro e alteram os nomes de objetos.

- As migrações de dados homogêneas não fornecem uma ferramenta integrada para validação dos dados.
- Ao usar migrações de dados homogêneas com o PostgreSQL, AWS DMS migra visualizações como tabelas para seu banco de dados de destino.
- As migrações de dados homogêneas não capturam alterações em nível de esquema durante uma replicação contínua de dados. Se você criar uma nova tabela no banco de dados de origem, não AWS DMS poderá migrar essa tabela. Para migrar essa nova tabela, reinicie a migração de dados.
- Você não pode usar migrações de dados homogêneas AWS DMS para migrar dados de uma versão superior do banco de dados para uma versão inferior do banco de dados.
- Não é possível utilizar migrações de dados homogêneas na CLI ou na API.
- As migrações de dados homogêneas não são compatíveis com o estabelecimento de uma conexão com instâncias de banco de dados em intervalos CIDR secundários da VPC.
- Não é possível usar a porta 8081 para migrações homogêneas provenientes de seus provedores de dados.
- As migrações de dados homogêneas não oferecem suporte à migração de bancos de dados e tabelas MySQL criptografados.

Visão geral do processo de migração de dados homogênea no AWS DMS

É possível utilizar migrações de dados homogêneas no AWS DMS para migrar dados entre dois bancos de dados do mesmo tipo. Utilize o fluxo de trabalho a seguir para criar e executar uma migração de dados.

1. Crie a política e o perfil do AWS Identity and Access Management (IAM) necessários. Para obter mais informações, consulte Criar recursos do IAM.
2. Configure os bancos de dados de origem e de destino e crie usuários de banco de dados com as permissões mínimas necessárias para migrações de dados homogêneas no AWS DMS. Para obter mais informações, consulte Criar provedores de dados de origem e Criar provedores de dados de destino.

3. Armazene as credenciais do banco de dados de origem e de destino no AWS Secrets Manager. Para obter mais informações, consulte Etapa 1: Criar um segredo no Guia do usuário do AWS Secrets Manager.
4. Crie um grupo de sub-redes, um perfil de instância e provedores de dados no console do AWS DMS. Para obter mais informações, consulte Criação de um grupo de sub-redes, Criação de perfis de instância e Criação de provedores de dados .
5. Crie um projeto de migração utilizando os recursos criados na etapa anterior. Para obter mais informações, consulte Criar projetos de migração.
6. Crie, configure e inicie uma migração de dados. Para obter mais informações, consulte Criar uma migração de dados.
7. Depois de concluir a carga máxima ou a replicação contínua, é possível fazer a transição para começar a usar o novo banco de dados de destino.
8. Limpe os recursos. A Amazon encerra a migração de dados no seu projeto de migração em três dias após você concluir a migração. No entanto, você precisa excluir manualmente os recursos, como o perfil da instância, os provedores de dados, a política e o perfil do IAM e os segredos no AWS Secrets Manager.

Para obter mais informações sobre migrações de dados homogêneas no AWS DMS, leia o passo a passo da migração em migrações do PostgreSQL para o Amazon RDS para PostgreSQL.

Esse vídeo apresenta as migrações de dados homogêneas no AWS DMS e ajuda você a se familiarizar com esse recurso.

Configurar migrações de dados homogêneas no AWS DMS

Para configurar migrações de dados homogêneas no AWS DMS, conclua as seguintes tarefas de pré-requisito.

Tópicos

- Criar os recursos do IAM necessários para migrações de dados homogêneas no AWS DMS
- Configurar uma rede para migrações de dados homogêneas no AWS DMS

Criar os recursos do IAM necessários para migrações de dados homogêneas no AWS DMS

Para executar migrações de dados homogêneas, crie uma política e um perfil do IAM na sua conta para interagir com outros serviços da AWS. Nesta seção, você criará esses recursos do IAM necessários.

Tópicos

Criar uma política do IAM para migrações de dados homogêneas no AWS DMS

Para acessar os bancos de dados e migrar os dados, o AWS DMS cria um ambiente com tecnologia sem servidor para migrações de dados homogêneas. Nesse ambiente, o AWS DMS exige acesso ao emparelhamento de VPC, a tabelas de rotas, a grupos de segurança e a outros recursos da AWS. Além disso, o AWS DMS armazena os logs, as métricas e o progresso de cada migração de dados no Amazon CloudWatch. Para criar um projeto de migração de dados, o AWS DMS precisa acessar esses serviços.

Nesta etapa, você cria uma política do IAM que fornece acesso aos recursos do Amazon EC2 e do CloudWatch ao AWS DMS. Crie um perfil do IAM e anexe uma política.

Como criar uma política do IAM para migrações de dados homogêneas no AWS DMS

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies (Políticas).
3. Escolha Create policy (Criar política).
4. Na página Criar política, escolha a guia JSON.
5. Cole a política o JSON a seguir no editor.

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

```
            "Action": [
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ec2:DescribePrefixLists",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "servicequotas:GetServiceQuota"
            ],
            "Resource": "arn:aws:servicequotas:*:*:vpc/L-0EA8095F"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*:log-
stream:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:DeleteRoute"
```

```
            ],
            "Resource": "arn:aws:ec2:*:*:route-table/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:security-group-rule/*",
                "arn:aws:ec2:*:*:route-table/*",
                "arn:aws:ec2:*:*:vpc-peering-connection/*",
                "arn:aws:ec2:*:*:vpc/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group-rule/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AcceptVpcPeeringConnection",
                "ec2:ModifyVpcPeeringConnectionOptions"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-peering-connection/*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": "ec2:AcceptVpcPeeringConnection",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        }
    ]
}
```

6. Selecione Próximo: tags e Próximo: revisar.
7. Insira **HomogeneousDataMigrationsPolicy** para Nome* e escolha Criar política.

Criar um perfil do IAM para migrações de dados homogêneas no AWS DMS

Nesta etapa, você cria um perfil do IAM que fornece ao AWS DMS acesso ao AWS Secrets Manager, ao Amazon EC2 e ao CloudWatch.

Como criar um perfil do IAM para migrações de dados homogêneas no AWS DMS

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles.
3. Escolha Criar função.
4. Na página Selecionar entidade confiável, em Tipo de entidade confiável, escolha Serviço da AWS. Em Casos de uso para outros serviços da AWS, escolha DMS.
5. Marque a caixa de seleção DMS e escolha Próximo.
6. Na página Adicionar permissões, escolha HomogeneousDataMigrationsPolicy criada anteriormente. Além disso, escolha SecretsManagerReadWrite. Escolha Próximo.
7. Na página Nomear, revisar e criar, insira **HomogeneousDataMigrationsRole** em Nome do perfil e escolha Criar função.
8. Na página Perfis, insira **HomogeneousDataMigrationsRole** em Nome do perfil. Escolha HomogeneousDataMigrationsRole.
9. Na página HomogeneousDataMigrationsRole, escolha a guia Relações de confiança. Escolha Editar política de confiança.
10. Na página Editar política de confiança, cole o seguinte JSON no editor, substituindo o texto existente.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

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

No exemplo anterior, substitua *your_region* pelo nome da sua Região da AWS.

A política anterior baseada em recursos fornece às entidades principais do AWS DMS as permissões para executar tarefas de acordo com as políticas SecretsManagerReadWrite gerenciada pela AWS e a HomogeneousDataMigrationsPolicy gerenciada pelo cliente.

11. Escolha Atualizar política.

Configurar uma rede para migrações de dados homogêneas no AWS DMS

O AWS DMS cria um ambiente com tecnologia sem servidor para migrações de dados homogêneas em uma nuvem privada virtual (VPC) com base no serviço Amazon VPC. Ao criar o perfil da instância, especifique a VPC a ser utilizada. É possível utilizar a VPC padrão da sua conta e Região da AWS ou criar uma nova VPC.

Para cada migração de dados, o AWS DMS estabelece uma conexão de emparelhamento de VPC com a VPC utilizada para o perfil de instância. Em seguida, o AWS DMS adiciona o bloco CIDR no grupo de segurança associado ao perfil de instância. Como o AWS DMS anexa um endereço IP público ao perfil de instância, todas as migrações de dados que utilizam o mesmo perfil de instância têm o mesmo endereço IP público. Quando a migração de dados é interrompida ou falha, o AWS DMS exclui a conexão de emparelhamento da VPC.

Para evitar que o bloco CIDR se sobreponha com a VPC da VPC do perfil de instância, o AWS DMS utiliza o prefixo `/24` de um dos seguintes blocos CIDR: `10.0.0.0/8`, `172.16.0.0/12` e `192.168.0.0/16`. Por exemplo, se você executar três migrações de dados em paralelo, o AWS DMS utilizará os seguintes blocos CIDR para estabelecer uma conexão de emparelhamento de VPC.

- `192.168.0.0/24`: para a primeira migração de dados
- `192.168.1.0/24`: para a segunda migração de dados
- `192.168.2.0/24`: para a terceira migração de dados

É possível utilizar diferentes configurações de rede para configurar a interação entre os bancos de dados de origem e de destino com o AWS DMS. Além disso, para a replicação contínua de dados, configure a interação entre os bancos de dados de origem e de destino. Essas configurações dependem da localização do provedor de dados de origem e das configurações de rede. As seções a seguir fornecem descrições de configurações de rede comuns.

Tópicos

Utilizar uma única VPC para provedores de dados de origem e de destino

Nessa configuração, o AWS DMS se conecta aos provedores de dados de origem e de destino na rede privada.

Como configurar uma rede quando seus provedores de dados de origem e de destino estão na mesma VPC

1. Crie o grupo de sub-redes no console do AWS DMS com a VPC e as sub-redes que os provedores de dados de origem e de destino utilizam. Para obter mais informações, consulte Criação de um grupo de sub-redes.
2. Crie o perfil de instância no console do AWS DMS com a VPC e o grupo de sub-redes criados. Além disso, escolha os grupos de segurança da VPC que os provedores de dados de origem e de destino utilizam. Para obter mais informações, consulte Criação de perfis de instância.

Essa configuração não exige que você utilize o endereço IP público para migrações de dados.

Utilizar VPCs diferentes para provedores de dados de origem e de destino

Nessa configuração, o AWS DMS utiliza uma rede privada para se conectar aos provedores de dados de origem e de destino. Para outro provedor de dados, o AWS DMS utiliza uma rede pública. Dependendo do provedor de dados que você tem na mesma VPC do perfil de instância, escolha uma das configurações a seguir.

Para configurar uma rede privada para o provedor de dados de origem e uma rede pública para o provedor de dados de destino

1. Crie o grupo de sub-redes no console do AWS DMS com a VPC e as sub-redes que o provedor de dados de origem utiliza. Para obter mais informações, consulte Criação de um grupo de sub-redes.
2. Crie o perfil de instância no console do AWS DMS com a VPC e o grupo de sub-redes criados. Além disso, escolha os grupos de segurança da VPC que o provedor de dados de origem utiliza. Para obter mais informações, consulte Criação de perfis de instância.
3. Abra o projeto de migração. Na guia Migrações de dados, escolha a migração de dados. Anote o Endereço IP público em Conectividade e segurança na guia Detalhes.
4. Permita acesso do endereço IP público da migração de dados no grupo de segurança do banco de dados de destino. Para obter mais informações, consulte Controlar o acesso com grupos de segurança no Guia do usuário do Amazon Relational Database Service.

Como configurar uma rede privada para o provedor de dados de origem e uma rede pública para o provedor de dados de destino

1. Crie o grupo de sub-redes no console do AWS DMS com a VPC e as sub-redes que o provedor de dados de destino utiliza. Para obter mais informações, consulte Criação de um grupo de sub-redes.
2. Crie o perfil de instância no console do AWS DMS com a VPC e o grupo de sub-redes criados. Além disso, escolha os grupos de segurança da VPC que o provedor de dados de destino utiliza. Para obter mais informações, consulte Criação de perfis de instância.
3. Abra o projeto de migração. Na guia Migrações de dados, escolha a migração de dados. Anote o Endereço IP público em Conectividade e segurança na guia Detalhes.
4. Permita o acesso do endereço IP público da migração de dados no grupo de segurança do banco de dados de origem. Para obter mais informações, consulte Controlar o acesso com grupos de segurança no Guia do usuário do Amazon Relational Database Service.

Utilizar um provedor de dados de origem on-premises

Nessa configuração, o AWS DMS se conecta ao provedor de dados de origem na rede pública. O AWS DMS utiliza uma rede privada para se conectar ao provedor de dados de destino.

Como configurar uma rede para o provedor de dados on-premises de origem

1. Crie o grupo de sub-redes no console do AWS DMS com a VPC e as sub-redes que o provedor de dados de destino utiliza. Para obter mais informações, consulte Criação de um grupo de sub-redes.
2. Crie o perfil de instância no console do AWS DMS com a VPC e o grupo de sub-redes criados. Além disso, escolha os grupos de segurança da VPC que o provedor de dados de destino utiliza. Para obter mais informações, consulte Criação de perfis de instância.
3. Abra o projeto de migração. Na guia Migrações de dados, escolha a migração de dados. Anote o Endereço IP público em Conectividade e segurança na guia Detalhes.
4. Permita acesso ao banco de dados de origem no endereço IP público da migração de dados no AWS DMS.

O AWS DMS cria regras de entrada ou de saída nos grupos de segurança da VPC. Verifique se essas regras não são excluídas, pois essa ação pode levar a uma falha na migração de dados. É possível configurar suas próprias regras nos grupos de segurança da VPC. É recomendável adicionar uma descrição às regras para poder gerenciá-las.

Configurar a replicação contínua de dados

Para executar migrações de dados do tipo Carga máxima e captura de dados de alteração (CDC) ou Captura de dados de alteração (CDC), permita a conexão entre os bancos de dados de origem e de destino.

Como configurar uma conexão entre os bancos de dados de origem e de destino acessíveis ao público

1. Anote os endereços IP públicos dos bancos de dados de origem e de destino.
2. Permita acesso ao banco de dados de origem no endereço IP público do banco de dados de destino.
3. Permita acesso ao banco de dados de destino no endereço IP público do banco de dados de origem.

Como configurar uma conexão entre os bancos de dados de origem e de destino acessíveis ao público em uma única VPC

1. Anote os endereços IP privados dos bancos de dados de origem e de destino.

   > **Important**
   >
   > Se os bancos de dados de origem e de destino estiverem em VPCs ou redes diferentes, só será possível utilizar endereços IP públicos para os bancos de dados de origem e de destino. Só é possível utilizar nomes de host ou endereços IP públicos em provedores de dados.

2. Permita acesso ao banco de dados de origem no endereço IP público do banco de dados de destino.
3. Permita acesso ao banco de dados de destino no endereço IP privado do banco de dados de origem.

Criação de provedores de dados de origem para migrações de dados homogêneas no AWS DMS

Você pode usar bancos de dados compatíveis com MySQL, PostgreSQL e MongoDB como provedor de dados de origem para o in. Migração de dados homogênea AWS DMS

Para ver as versões de banco de dados compatíveis, consulteProvedores de dados de origem para migrações de dados homogêneas do DMS.

O provedor de dados de origem pode ser um banco de dados on-premises, Amazon EC2 ou Amazon RDS.

Tópicos

- Usando um banco de dados compatível com MySQL como fonte para migrações de dados homogêneas em AWS DMS
- Usando um banco de dados PostgreSQL como fonte para migrações de dados homogêneas em AWS DMS
- Usando um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas em AWS DMS

Usando um banco de dados compatível com MySQL como fonte para migrações de dados homogêneas em AWS DMS

É possível utilizar um banco de dados compatível com MySQL (MySQL ou MariaDB) como origem da Migração de dados homogênea no AWS DMS. Nesse caso, o provedor de dados de origem pode ser um banco de dados on-premises, Amazon EC2, RDS para MySQL ou MariaDB.

Para executar migrações de dados homogêneas, utilize um usuário do banco de dados com privilégios `SELECT` para todas as tabelas de origem e objetos secundários para replicação. Para tarefas de captura de dados de alteração (CDC), esse usuário também deve ter privilégios `REPLICATION CLIENT` (`BINLOG MONITOR` para versões do MariaDB posteriores à 10.5.2) e privilégios `REPLICATION SLAVE`. Para uma migração de dados de carga máxima, esses dois privilégios não são necessários.

Utilize o script a seguir para criar um usuário de banco de dados com as permissões necessárias no banco de dados MySQL. Execute as `GRANT` consultas para todos os bancos de dados para os quais você migra. 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'@'%';
```

No exemplo anterior, substitua cada *espaço reservado para entrada de usuário* pelas suas próprias informações. Se a versão do banco de dados MySQL de origem for inferior à 8.0, é possível ignorar o comando `GRANT BACKUP_ADMIN`.

Utilize o script a seguir para criar um usuário de banco de dados com as permissões necessárias no banco de dados MariaDB. Execute as consultas GRANT para todos os bancos de dados para os quais você migra. AWS

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT, RELOAD, LOCK TABLES, REPLICATION SLAVE, BINLOG MONITOR, SHOW VIEW ON  *.*
 TO 'your_user'@'%';
```

No exemplo anterior, substitua cada *espaço reservado para entrada de usuário* pelas suas próprias informações.

As seções a seguir descrevem os pré-requisitos de configuração específicos para bancos de dados MySQL autogerenciados e gerenciados pela AWS.

Tópicos

Utilizar um banco de dados compatível com MySQL autogerenciado como origem para migrações de dados homogêneas

Esta seção descreve como configurar os bancos de dados compatíveis com MySQL hospedados on-premises ou em instâncias do Amazon EC2.

Escolha a versão do banco de dados MySQL ou MariaDB de origem. Certifique-se de que seja AWS DMS compatível com a versão de origem do banco de dados MySQL ou MariaDB, conforme descrito em. Origens para migrações de dados homogêneas do DMS

Para utilizar a CDC, ative o registro em log binário. Para habilitar o registro em log binário, os parâmetros a seguir devem ser configurados nos arquivos `my.ini` (Windows) ou `my.cnf` (UNIX) do banco de dados MySQL ou MariaDB.

Parâmetro	Valor
`server-id`	Defina este parâmetro com um valor maior ou igual a 1.
`log-bin`	Defina a rota para o arquivo de log binário, por exemplo `log-bin=E:\MySql_Logs\BinLog` . Não inclua a extensão do arquivo.
`binlog_format`	Defina este parâmetro como `ROW`. Essa configuração é recomendável durante a replicação porque, em certos casos, quando `binlog_format` está definido como `STATEMENT` , ele pode causar inconsistência ao replicar dados para o destino. O mecanismo de banco de dados também grava dados inconsistentes semelhantes no destino quando `binlog_format` está definido como `MIXED`, porque o mecanismo de

Parâmetro	Valor
	banco de dados muda automaticamente para o registro em log baseado em `STATEMENT` .
`expire_logs_days`	Defina este parâmetro com um valor maior ou igual a 1. Para evitar o uso excessivo de espaço em disco, recomendamos que você não utilize o valor padrão de 0.
`binlog_checksum`	Defina este parâmetro como `NONE`.
`binlog_row_image`	Defina este parâmetro como `FULL`.
`log_slave_updates`	Defina este parâmetro como `TRUE` se estiver utilizando uma réplica do MySQL ou do MariaDB como origem.

Usando um banco AWS de dados compatível com MySQL gerenciado como fonte para migrações de dados homogêneas em AWS DMS

Esta seção descreve como configurar as instâncias de banco de dados Amazon RDS para MySQL e Amazon RDS para MariaDB.

Ao usar um banco AWS de dados MySQL ou MariaDB gerenciado como fonte para migrações AWS DMS de dados homogêneas, verifique se você tem os seguintes pré-requisitos para o CDC:

- Para ativar os logs binários para o RDS para MySQL e MariaDB, ative backups automáticos no nível da instância. Para ativar logs binários para um cluster do Aurora MySQL, altere a variável `binlog_format` no grupo de parâmetros. Não é necessário ativar backups automáticos em um cluster do Aurora MySQL.

  Próximo, defina o parâmetro `binlog_format` como `ROW`.

  Para obter mais informações sobre como configurar backups automáticos, consulte Ativar backups automáticos no Guia do usuário do Amazon RDS.

  Para obter mais informações sobre como configurar o registro em log binário para um banco de dados Amazon RDS para MySQL ou MariaDB, consulte Configuração do formato do registro em log binário no Guia do usuário do Amazon RDS.

Para obter mais informações sobre como configurar o registro em log binário para um cluster do Aurora MySQL, consulte Como faço para ativar o registro em log binário para meu cluster do Amazon Aurora MySQL?.

- Certifique-se de que os registros binários estejam disponíveis para AWS DMS o. Como os bancos AWS de dados MySQL e MariaDB gerenciados eliminam os registros binários o mais rápido possível, você deve aumentar o tempo em que os registros permanecem disponíveis. Por exemplo, para aumentar a retenção de log para 24 horas, execute o comando a seguir.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Defina o parâmetro `binlog_row_image` como `Full`.
- Defina o parâmetro `binlog_checksum` como `NONE`.
- Se estiver utilizando uma réplica do Amazon RDS MySQL ou do MariaDB como origem, ative os backups na réplica de leitura e verifique se o parâmetro `log_slave_updates` está definido como `TRUE`.

Limitações para utilizar um banco de dados compatível com MySQL como origem para migrações de dados homogêneas

As seguintes limitações se aplicam ao utilizar um banco de dados compatível com MySQL como origem para migrações de dados homogêneas:

- Objetos MariaDB, como sequências, não são compatíveis em tarefas de migração homogêneas.
- A migração do MariaDB para o Amazon RDS MySQL/Aurora MySQL pode falhar devido a diferenças de objetos incompatíveis.
- O nome de usuário que você usa para se conectar à fonte de dados tem as seguintes limitações:
  - Pode ter de 2 a 64 caracteres de extensão.
  - Não pode ter espaços.
  - Pode incluir os seguintes caracteres: a-z, A-Z, 0-9, sublinhado (_).
  - Deve começar com a-z ou A-Z.
- A senha que você usa para se conectar à fonte de dados tem as seguintes limitações:
  - Pode ter de 1 a 128 caracteres de extensão.
  - Não pode conter nenhum dos seguintes: aspas simples (‘), aspas duplas (“), ponto e vírgula (;) ou espaço.

Usando um banco de dados PostgreSQL como fonte para migrações de dados homogêneas em AWS DMS

Utilize um banco de dados PostgreSQL como origem para Migração de dados homogênea no AWS DMS. Nesse caso, o provedor de dados de origem pode ser um banco de dados on-premises, o Amazon EC2 ou o RDS para PostgreSQL.

Para executar migrações de dados homogêneas, conceda permissões de superusuário para o usuário do banco de dados que você especificou para AWS DMS seu banco de dados de origem do PostgreSQL. O usuário do banco de dados precisa de permissões de superusuário para acessar perfil específicos de replicação na origem. Para uma migração de dados de carga máxima, o usuário do banco de dados precisa de permissões `SELECT` nas tabelas para migrá-las.

Utilize o script a seguir para criar um usuário de banco de dados com as permissões necessárias no banco de dados de origem do PostgreSQL. Execute a `GRANT` consulta para todos os bancos de dados para os quais você migra. 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;
```

No exemplo anterior, substitua cada *espaço reservado para entrada de usuário* pelas suas próprias informações.

As seções a seguir descrevem os pré-requisitos de configuração específicos para bancos de dados PostgreSQL autogerenciados e gerenciados pela AWS.

Tópicos

- Usando um banco de dados PostgreSQL autogerenciado como fonte para migrações de dados homogêneas em AWS DMS
- Usando um banco AWS de dados PostgreSQL gerenciado como fonte para migrações de dados homogêneas em AWS DMS
- Limitações para utilizar um banco de dados compatível com PostgreSQL como origem para migrações de dados homogêneas

Usando um banco de dados PostgreSQL autogerenciado como fonte para migrações de dados homogêneas em AWS DMS

Esta seção descreve como configurar os bancos de dados PostgreSQL hospedados on-premises ou em instâncias do Amazon EC2.

Verifique a versão do banco de dados PostgreSQL de origem. Certifique-se de que seja AWS DMS compatível com a versão de origem do banco de dados PostgreSQL, conforme descrito em. Origens para migrações de dados homogêneas do DMS

As migrações de dados homogêneas são compatíveis com a captura de dados de alteração (CDC) utilizando a replicação lógica. Para ativar a replicação lógica de um banco de dados de origem do PostgreSQL autogerenciado, defina os seguintes parâmetros e valores no arquivo de configuração `postgresql.conf`:

- Defina `wal_level` como `logical`.
- Defina `max_replication_slots` como um valor maior que 1.

  Defina o valor de `max_replication_slots` de acordo com o número de tarefas a serem executadas. Por exemplo, para executar cinco tarefas, defina no mínimo cinco slots. Os slots são abertos automaticamente assim que uma tarefa é iniciada e permanecem abertos até mesmo quando a tarefa não está mais em execução. Exclua manualmente os slots abertos.
- Defina `max_wal_senders` como um valor maior que 1.

  O parâmetro `max_wal_senders` define o número de tarefas simultâneas que podem ser executadas.
- O parâmetro `wal_sender_timeout` encerra as conexões de replicação que estão inativas por mais tempo do que o número de milissegundos especificado. O padrão é 60000 milissegundos (60 segundos). A definição do valor como 0 (zero) desativa o mecanismo de tempo limite e é uma configuração válida para o DMS.

Alguns parâmetros são estáticos, e você só pode defini-los na inicialização do servidor. Quaisquer alterações em suas entradas no arquivo de configuração são ignoradas até que o servidor seja reiniciado. Para obter mais informações, consulte a Documentação do PostgreSQL.

Usando um banco AWS de dados PostgreSQL gerenciado como fonte para migrações de dados homogêneas em AWS DMS

Esta seção descreve como configurar as instâncias de bancos de dados Amazon RDS para PostgreSQL.

Use a conta de usuário AWS principal da instância de banco de dados PostgreSQL como a conta de usuário do provedor de dados de origem do PostgreSQL para migrações de dados homogêneas em. AWS DMS A conta de usuário mestra tem as funções necessárias para permitir a configuração da captura de dados de alteração (CDC). Se você utilizar uma conta diferente da conta de usuário mestre, esta deverá ter os perfis `rds_superuser` e `rds_replication`. O perfil `rds_replication` concede permissões para gerenciar slots lógicos e transmitir dados utilizando slots lógicos.

Utilize o exemplo de código a seguir para conceder as perfis `rds_superuser` e `rds_replication`.

```
GRANT rds_superuser to your_user;
GRANT rds_replication to your_user;
```

No exemplo anterior, substitua *`your_user`* pelo nome do usuário do banco de dados.

Para ativar a replicação lógica, defina o parâmetro `rds.logical_replication` no grupo de parâmetros de banco de dados como 1. Esse parâmetro estático requer uma reinicialização da instância de banco de dados para entrar em vigor.

Limitações para utilizar um banco de dados compatível com PostgreSQL como origem para migrações de dados homogêneas

As seguintes limitações se aplicam ao utilizar um banco de dados compatível com PostgreSQL como origem para migrações de dados homogêneas:

- O nome de usuário que você usa para se conectar à fonte de dados tem as seguintes limitações:
  - Pode ter de 2 a 64 caracteres de extensão.
  - Não pode ter espaços.
  - Pode incluir os seguintes caracteres: a-z, A-Z, 0-9, sublinhado (_).
  - Deve começar com a-z ou A-Z.
- A senha que você usa para se conectar à fonte de dados tem as seguintes limitações:

- Pode ter de 1 a 128 caracteres de extensão.
- Não pode conter nenhum dos seguintes: aspas simples (‘), aspas duplas (“), ponto e vírgula (;) ou espaço.

Usando um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas em AWS DMS

Você pode usar um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas em. AWS DMS Nesse caso, seu provedor de dados de origem pode ser um banco de dados local, Amazon EC2 para MongoDB ou Amazon DocumentDB (com compatibilidade com o MongoDB).

Para ver as versões de banco de dados compatíveis, consulteProvedores de dados de origem para migrações de dados homogêneas do DMS.

As seções a seguir descrevem os pré-requisitos de configuração específicos para bancos de dados MongoDB autogerenciados e bancos de dados Amazon DocumentDB gerenciados. AWS

Tópicos

- Usando um banco de dados MongoDB autogerenciado como fonte para migrações de dados homogêneas em AWS DMS
- Usando um banco de dados Amazon DocumentDB como fonte para migrações de dados homogêneas em AWS DMS
- Recursos para usar um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas
- Limitações para usar um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas
- Melhores práticas para usar um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas

Usando um banco de dados MongoDB autogerenciado como fonte para migrações de dados homogêneas em AWS DMS

Esta seção descreve como configurar seus bancos de dados MongoDB hospedados localmente ou em instâncias do Amazon EC2.

Verifique a versão do seu banco de dados MongoDB de origem. Certifique-se de que seja AWS DMS compatível com a versão de origem do banco de dados MongoDB, conforme descrito em. Provedores de dados de origem para migrações de dados homogêneas do DMS

Para executar migrações de dados homogêneas com uma fonte do MongoDB, você pode criar uma conta de usuário com privilégios de root ou um usuário com permissões somente no banco de dados para migrar. Para obter mais informações sobre a criação de usuários, consultePermissões necessárias ao utilizar o MongoDB como origem do AWS DMS.

Para usar a replicação contínua ou CDC com o MongoDB, é necessário AWS DMS acesso ao log de operações do MongoDB (oplog). Para ter mais informações, consulte Configurar um conjunto de réplicas do MongoDB para a CDC.

Para obter informações sobre os métodos de autenticação do MongoDB, consulte. Requisitos de segurança ao utilizar o MongoDB como origem do AWS DMS

Para o MongoDB como fonte, as migrações de dados homogêneas oferecem suporte a todos os tipos de dados compatíveis com o Amazon DocumentDB.

Para o MongoDB como fonte, para armazenar as credenciais do usuário no Secrets Manager, você precisa fornecê-las em texto simples, usando o tipo Outro tipo de segredos. Para ter mais informações, consulte Utilizar segredos para acessar endpoints do AWS Database Migration Service.

O exemplo de código a seguir demonstra como armazenar segredos do banco de dados usando texto sem formatação.

```
{
  "username": "dbuser",
  "password": "dbpassword"
}
```

Usando um banco de dados Amazon DocumentDB como fonte para migrações de dados homogêneas em AWS DMS

Esta seção descreve como configurar suas instâncias de banco de dados Amazon DocumentDB para uso como fonte para migrações de dados homogêneas.

Use o nome de usuário principal da instância Amazon DocumentDB como a conta de usuário do provedor de dados de origem compatível com MongoDB para migrações de dados homogêneas

em. AWS DMS A conta de usuário mestra tem as funções necessárias para permitir a configuração da captura de dados de alteração (CDC). Se você usar uma conta diferente da conta de usuário principal, a conta deverá ter a função raiz. Para obter mais informações sobre a criação do usuário como conta raiz, consulteDefinir permissões para utilizar o Amazon DocumentDB como origem.

Para ativar a replicação lógica, defina o `change_stream_log_retention_duration` parâmetro no grupo de parâmetros do banco de dados para uma configuração apropriada para a carga de trabalho da transação. A alteração desse parâmetro estático exige que você reinicialize sua instância de banco de dados para entrar em vigor. Antes de iniciar a migração de dados para todos os tipos de tarefas, incluindo Full Load Only, habilite os fluxos de alteração do Amazon DocumentDB para todas as coleções em um determinado banco de dados ou somente para coleções selecionadas. Para obter mais informações sobre como habilitar fluxos de mudança para o Amazon DocumentDB, consulte Habilitando fluxos de mudança no guia do desenvolvedor do Amazon DocumentDB.

> **Note**
>
> AWS DMS usa o stream de alterações do Amazon DocumentDB para capturar alterações durante a replicação contínua. Se o Amazon DocumentDB eliminar os registros do stream de alterações antes que o DMS os leia, suas tarefas falharão. Recomendamos definir o `change_stream_log_retention_duration` parâmetro para reter as alterações por pelo menos 24 horas.

Para usar o Amazon DocumentDB para migração homogênea de dados, armazene as credenciais do usuário no Secrets Manager em Credenciais para o banco de dados Amazon DocumentDB.

Recursos para usar um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas

- Você pode migrar todos os índices secundários que o Amazon DocumentDB suporta durante a fase de carregamento total.
- AWS DMS migra coleções em paralelo. As migrações de dados homogêneas calculam segmentos em tempo de execução com base no tamanho médio de cada documento na coleção para obter o máximo desempenho.
- O DMS pode replicar os índices secundários que você cria na fase CDC. O DMS oferece suporte a esse recurso no MongoDB versão 6.0.
- O DMS oferece suporte a documentos com um nível de aninhamento maior que 97.

Limitações para usar um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas

- Os documentos não podem ter nomes de campo com $ prefixo.
- AWS DMS não oferece suporte à migração de coleções de séries temporais.
- AWS DMS não oferece suporte `create` a eventos `drop` de `rename collection` DDL durante a fase CDC.
- AWS DMS não suporta tipos de dados inconsistentes na coleção do campo. `_id` Por exemplo, a coleção não suportada a seguir tem vários tipos de dados para o `_id` campo.

```
rs0 [direct: primary] test> db.collection1.aggregate([
...   {
...     $group: {
...       _id: { $type: "$_id" },
...       count: { $sum: 1 }
...     }
...   }
... ])
[ { _id: 'string', count: 6136 }, { _id: 'objectId', count: 848033 } ]
```

- Para tarefas somente CDC, suporta AWS DMS apenas o modo de `immediate` início.
- AWS DMS não suporta documentos com caracteres UTF8 inválidos.
- AWS DMS não suporta coleções fragmentadas.

Melhores práticas para usar um banco de dados compatível com MongoDB como fonte para migrações de dados homogêneas

- Para vários bancos de dados e coleções grandes hospedados na mesma instância do MongoDB, recomendamos que você use regras de seleção para cada banco de dados e coleção para dividir a tarefa entre várias tarefas e projetos de migração de dados. Você pode ajustar suas divisões de banco de dados e coleção para obter o máximo desempenho.

Criação de provedores de dados de destino para migrações de dados homogêneas no AWS DMS

Você pode usar bancos de dados compatíveis com MySQL, PostgreSQL e Amazon DocumentDB como um provedor de dados de destino para migrações de dados homogêneas em. AWS DMS

Para ver as versões de banco de dados compatíveis, consulteProvedores de dados de destino para migrações de dados homogêneas do DMS.

O provedor de dados de destino pode ser uma instância de banco de dados Amazon RDS ou um cluster de banco de dados Amazon Aurora. Observe que a versão do banco de dados do seu provedor de dados de destino deve ser igual ou superior à versão do banco de dados do seu provedor de dados de origem.

Tópicos

Usando um banco de dados compatível com MySQL como alvo para migrações de dados homogêneas em AWS DMS

É possível utilizar um banco de dados compatível com MySQL como destino para a migração de dados homogênea no AWS DMS.

AWS DMS requer certas permissões para migrar dados para seu banco de dados Amazon RDS for MySQL ou MariaDB ou Amazon Aurora MySQL de destino. Utilize o script a seguir para criar um usuário de banco de dados com as permissões necessárias no banco de dados de destino MySQL.

```
CREATE USER 'your_user'@'%' IDENTIFIED BY 'your_password';

GRANT ALTER, CREATE, DROP, INDEX, INSERT, UPDATE, DELETE, SELECT, CREATE VIEW, CREATE
 ROUTINE, ALTER ROUTINE, EVENT, TRIGGER, EXECUTE, REFERENCES ON *.* TO 'your_user'@'%';
GRANT REPLICATION SLAVE, REPLICATION CLIENT  ON *.* TO 'your_user'@'%';
```

No exemplo anterior, substitua cada *espaço reservado para entrada de usuário* pelas suas próprias informações.

Utilize o script a seguir para criar um usuário de banco de dados com as permissões necessárias no banco de dados MariaDB. Execute as consultas GRANT para todos os bancos de dados para os quais você migra. 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'@'%';
```

No exemplo anterior, substitua cada *espaço reservado para entrada de usuário* pelas suas próprias informações.

> **Note**
>
> No Amazon RDS, ao ativar o backup automatizado para uma instância de banco de dados MySQL/Maria, você também ativa o registro em log binário. Quando essas configurações estão ativadas, a tarefa de migração de dados pode falhar com o seguinte erro ao criar objetos secundários, como perfis, procedimentos e gatilhos no banco de dados de destino. Se o banco de dados de destino tiver o registro em log binário ativado, defina `log_bin_trust_function_creators` como `true` no grupo de parâmetros do banco de dados antes de iniciar a tarefa.
>
> ```
> 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)
> ```

Limitações para utilizar um banco de dados compatível com MySQL como destino para migrações de dados homogêneas

As seguintes limitações se aplicam ao utilizar um banco de dados compatível com MySQL como destino para migrações de dados homogêneas:

- O nome de usuário que você usa para se conectar à fonte de dados tem as seguintes limitações:
  - Pode ter de 2 a 64 caracteres de extensão.

  - Não pode ter espaços.
  - Pode incluir os seguintes caracteres: a-z, A-Z, 0-9, sublinhado (_).
  - Não é possível incluir um hífen (-).
  - Deve começar com a-z ou A-Z.
- A senha que você usa para se conectar à fonte de dados tem as seguintes limitações:
  - Pode ter de 1 a 128 caracteres de extensão.
  - Não pode conter nenhum dos seguintes: aspas simples (‘), aspas duplas (“), ponto e vírgula (;) ou espaço.

Usando um banco de dados PostgreSQL como alvo para migrações de dados homogêneas em AWS DMS

É possível utilizar um banco de dados PostgreSQL como destino para migrações de dados homogêneas no AWS DMS.

AWS DMS exige certas permissões para migrar dados para seu banco de dados Amazon RDS for PostgreSQL ou Amazon Aurora PostgreSQL de destino. Utilize o script a seguir para criar um usuário de banco de dados com as permissões necessárias no banco de dados de destino do PostgreSQL.

```
CREATE USER your_user WITH LOGIN PASSWORD 'your_password';
GRANT USAGE ON SCHEMA schema_name TO your_user;
GRANT CONNECT ON DATABASE db_name to your_user;
GRANT CREATE ON DATABASE db_name TO your_user;
GRANT CREATE ON SCHEMA schema_name TO your_user;
GRANT UPDATE, INSERT, SELECT, DELETE, TRUNCATE ON ALL TABLES IN SCHEMA schema_name
 TO your_user;
           #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;
```

No exemplo anterior, substitua cada *espaço reservado para entrada de usuário* pelas suas próprias informações.

Para ativar a replicação lógica para o destino do RDS para PostgreSQL, defina o parâmetro `rds.logical_replication` no grupo de parâmetros de banco de dados como 1. Esse parâmetro estático requer uma reinicialização da instância ou do cluster do banco de dados para entrar em

vigor. Alguns parâmetros são estáticos e você só pode defini-los na inicialização do servidor. AWS DMS ignora as alterações em suas entradas no grupo de parâmetros do banco de dados até você reiniciar o servidor.

O PostgreSQL utiliza acionadores para implementar restrições de chaves estrangeiras. Durante a fase de carga total, AWS DMS carrega cada tabela uma de cada vez. É recomendável desativar as restrições de chave estrangeira no banco de dados de destino durante a carga máxima. Para fazer isso, utilize um dos seguintes métodos:

- Desative temporariamente todos os acionadores da instância e conclua a carga máxima.
- Altere o valor do parâmetro `session_replication_role` no PostgreSQL.

  Em determinado momento, um trigger pode estar em um dos seguintes estados: `origin`, `replica`, `always`, ou `disabled`. Ao definir o parâmetro `session_replication_role` como `replica`, somente os acionadores no estado `replica` ficam ativos. Caso contrário, os triggers permanecem inativos.

Limitações para utilizar um banco de dados compatível com PostgreSQL como destino para migrações de dados homogêneas

As seguintes limitações se aplicam ao utilizar um banco de dados compatível com PostgreSQL como destino para migrações de dados homogêneas:

- O nome de usuário que você usa para se conectar à fonte de dados tem as seguintes limitações:
  - Pode ter de 2 a 64 caracteres de extensão.
  - Não pode ter espaços.
  - Pode incluir os seguintes caracteres: a-z, A-Z, 0-9, sublinhado (_).
  - Deve começar com a-z ou A-Z.
- A senha que você usa para se conectar à fonte de dados tem as seguintes limitações:
  - Pode ter de 1 a 128 caracteres de extensão.
  - Não pode conter nenhum dos seguintes: aspas simples (‘), aspas duplas (“), ponto e vírgula (;) ou espaço.

Usando um banco de dados Amazon DocumentDB como destino para migrações de dados homogêneas em AWS DMS

Você pode usar um banco de dados Amazon DocumentDB (com compatibilidade com MongoDB) e um cluster DocumentDB Elastic como destino de migração para migrações de dados homogêneas em. AWS DMS

Para executar migrações de dados homogêneas para um destino do Amazon DocumentDB, você pode criar uma conta de usuário com privilégios de administrador ou um usuário com permissões de leitura/gravação somente no banco de dados a ser migrado.

As migrações de dados homogêneas oferecem suporte a todos os tipos de dados BSON compatíveis com o Amazon DocumentDB. Para obter uma lista desses tipos de dados, consulte Tipos de dados no Guia do desenvolvedor do Amazon DocumentDB.

Para usar os recursos de fragmentos do cluster DocumentDB Elastic para migrar a coleção não fragmentada da fonte, crie uma coleção de fragmentos para migrar antes de iniciar a tarefa de migração de dados. Para obter mais informações sobre a coleta de fragmentos em um cluster elástico do Amazon DocumentDB, consulte Etapa 5: Compartilhe sua coleção no Guia do desenvolvedor do Amazon DocumentDB.

Para um destino do Amazon DocumentDB, é AWS DMS compatível com os modos `none` ou `require` SSL.

Executando migrações de dados homogêneas em AWS DMS

Você pode usar Migração de dados homogênea in AWS DMS para migrar dados do seu banco de dados de origem para o mecanismo equivalente no Amazon Relational Database Service (Amazon RDS), Amazon Aurora ou Amazon DocumentDB. AWS DMS automatiza o processo de migração de dados usando ferramentas de banco de dados nativas em seus bancos de dados de origem e destino.

Depois de criar um perfil de instância e provedores de dados compatíveis para migrações de dados homogêneas, crie um projeto de migração. Para ter mais informações, consulte Criar projetos de migração.

As seções a seguir descrevem como criar, configurar e executar migrações de dados homogêneas.

Tópicos

- Criando uma migração de dados em AWS DMS
- Regras de seleção para migrações de dados homogêneas
- Gerenciando migrações de dados em AWS DMS
- Monitorando migrações de dados em AWS DMS
- Status de migrações de dados homogêneas em AWS DMS
- Migração de dados de bancos de dados MySQL com migrações de dados homogêneas em AWS DMS
- Migração de dados de bancos de dados PostgreSQL com migrações de dados homogêneas em AWS DMS
- Migração de dados de bancos de dados MongoDB com migrações de dados homogêneas em AWS DMS

Criando uma migração de dados em AWS DMS

Depois de criar um projeto de migração com provedores de dados compatíveis do mesmo tipo, é possível utilizar esse projeto para migrações de dados homogêneas. Para ter mais informações, consulte Criar projetos de migração.

Para começar a utilizar migrações de dados homogêneas, crie uma migração de dados. É possível criar várias migrações de dados homogêneas de diferentes tipos em um único projeto de migração.

AWS DMS tem o número máximo de migrações de dados homogêneas que você pode criar para o seu. Conta da AWS Consulte a seção a seguir para obter informações sobre cotas AWS DMS Cotas para o AWS Database Migration Service de serviço.

Antes de criar uma migração de dados, configure os recursos necessários, como bancos de dados de origem e de destino, uma política e um perfil do IAM, um perfil de instância e provedores de dados. Para obter mais informações, consulte Criar recursos do IAM, Criação de perfis de instância e Criação de provedores de dados .

Além disso, é recomendável não utilizar migrações de dados homogêneas para migrar dados de uma versão superior do banco de dados para uma versão inferior. Verifique as versões dos bancos de dados utilizadas para os provedores de dados de origem e de destino e atualize a versão do banco de dados de destino, se necessário.

Como criar uma migração de dados

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e, na guia Migrações de dados, escolha Criar migração de dados.
4. Em Nome, insira um nome para a migração de dados. Utilize um nome exclusivo para a migração de dados para que você possa identificá-la facilmente.
5. Em Tipo de replicação, escolha o tipo de migração de dados que deseja configurar. É possível escolher uma das seguintes opções:

   - Carga máxima: migra os dados de origem existentes.
   - Carga máxima e captura de dados alterados (CDC): migra os dados de origem existentes e replica as alterações em andamento.
   - Captura de dados alterados (CDC): replica as alterações em andamento.
6. Marque a caixa de seleção Ativar CloudWatch registros para armazenar registros de migração de dados na Amazon CloudWatch. Se você não escolher essa opção, não será possível ver os arquivos de log quando houver falha na migração de dados.
7. (Opcional) Expanda Advanced settings (Configurações avançadas). Em Número de trabalhos, insira o número de threads paralelos que AWS DMS podem ser usados para migrar seus dados de origem para o destino.
8. Em Perfil do serviço do IAM, escolha o perfil do IAM criado nos pré-requisitos. Para ter mais informações, consulte Criar um perfil do IAM para migrações de dados homogêneas no AWS DMS.
9. Configure o Modo de início para migrações de dados do tipo Captura de dados alterados (CDC). É possível escolher uma das seguintes opções:

   - Imediatamente: inicia a replicação contínua quando você inicia a migração de dados.
   - Utilizar um ponto de início nativo: inicia a replicação contínua no ponto especificado.

     Para bancos de dados PostgreSQL, insira o nome do slot de replicação lógica em Nome do slot e insira o número de sequência do log de transações para o Ponto de início nativo.

     Para bancos de dados MySQL, insira o número de sequência do log de transações para o Número de sequência do log (LSN).

10. Configure o Modo de interrupção para migrações de dados do tipo Captura de dados alterados (CDC) ou Carga máxima e captura de dados alterados (CDC). É possível escolher uma das seguintes opções:

    - Não interrompa o CDC — AWS DMS continua a replicação contínua até que você interrompa a migração de dados.
    - Usando um ponto de tempo do servidor — AWS DMS interrompe a replicação contínua no horário especificado.

      Se você escolher essa opção, em Data e hora de interrupção, insira a data e a hora em que deseja interromper automaticamente a replicação contínua.
11. Escolha Criar migração de dados.

AWS DMS cria sua migração de dados e a adiciona à lista na guia Migrações de dados em seu projeto de migração. Aqui é possível ver o status da migração de dados. Para ter mais informações, consulte Status das migrações.

> **Important**
>
> Para migrações de dados do tipo Carga completa e Carga total e captura de dados de alteração (CDC), AWS DMS exclui todos os dados, tabelas e outros objetos de banco de dados no banco de dados de destino. Faça um backup do banco de dados de destino.

Depois de AWS DMS criar sua migração de dados, o status dessa migração de dados é definido como Pronto. Para migrar os dados, inicie a migração de dados manualmente. Para fazer isso, escolha a migração de dados na lista. Em Ações, escolha Iniciar. Para ter mais informações, consulte Gerenciar migrações de dados.

O primeiro lançamento de uma migração de dados homogênea requer alguma configuração. AWS DMS cria um ambiente sem servidor para sua migração de dados. Esse processo pode demorar até 15 minutos. Depois de interromper e reiniciar sua migração de dados, AWS DMS não cria o ambiente novamente e você pode acessar sua migração de dados mais rapidamente.

Regras de seleção para migrações de dados homogêneas

Você pode usar as regras de seleção para escolher o esquema, as tabelas ou os dois que deseja incluir na replicação.

> **Note**
>
> AWS DMS só oferece suporte a regras de seleção para migrações de dados homogêneas ao usar um banco de dados compatível com MongoDB como fonte.

Ao criar a tarefa de migração de dados, escolha Adicionar regra de seleção.

Para as configurações da regra, forneça os seguintes valores:

- Esquema: escolha Inserir um esquema.
- Nome do esquema: forneça o nome do esquema que você deseja replicar ou usar `%` como curinga.
- Nome da tabela: Forneça o nome da tabela que você deseja replicar ou use `%` como curinga.

Por padrão, a única ação de regra que o DMS suporta é`Include`, e o único caractere curinga que o DMS suporta é. `%`

Example Migrar todas as tabelas em um esquema

O exemplo a seguir migra todas as tabelas de um esquema chamado `dmsst` da origem para o endpoint de destino.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "dmsst",
                "table-name": "%"
            },
            "filters": [],
            "rule-id": "1",
            "rule-name": "1"
        }
    ]
}
```

Example Migrar algumas tabelas em um esquema

O exemplo a seguir migra todas as tabelas com um nome começando com`collectionTest`, de um esquema nomeado `dmsst` em sua origem para seu endpoint de destino.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-action": "include",
            "object-locator": {
                "schema-name": "dmsst",
                "table-name": "collectionTest%"
            },
            "filters": [],
            "rule-id": "1",
            "rule-name": "1"
        }
    ]
}
```

Example Migre tabelas específicas de vários esquemas

O exemplo a seguir migra algumas das tabelas de vários esquemas nomeados `dmsst` e `Test` em sua origem para o endpoint de destino.

```
{
    "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"
        }
    ]
}
```

Gerenciando migrações de dados em AWS DMS

Depois de criar uma migração de dados, AWS DMS não inicia automaticamente a migração de dados. Inicie uma migração de dados manualmente quando necessário.

Antes de iniciar uma migração de dados, é possível modificar todas as configurações da migração de dados. Depois de iniciar a migração de dados, não é possível alterar o tipo de replicação. Para utilizar outro tipo de replicação, crie uma nova migração de dados.

Como iniciar uma migração de dados

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração. Na guia Migrações de dados, escolha a migração de dados. A página Resumo da migração de dados é aberta.
4. Em Ações, escolha Iniciar.

   Depois disso, AWS DMS cria um ambiente sem servidor para sua migração de dados. Esse processo pode demorar até 15 minutos.

Depois de iniciar uma migração de dados, AWS DMS defina seu status como Iniciando. O próximo status AWS DMS usado para sua migração de dados depende do tipo de replicação que você escolher nas configurações de migração de dados. Para ter mais informações, consulte Status das migrações.

Como modificar uma migração de dados

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração. Na guia Migrações de dados, escolha a migração de dados. A página Resumo da migração de dados é aberta.
4. Escolha Modificar.
5. Defina as configurações da migração de dados.

   > **Important**
   >
   > Depois de iniciar uma migração de dados, não é possível alterar o tipo de replicação.

6. Para visualizar seus registros de migração de dados na Amazon CloudWatch, marque a caixa de seleção Ativar CloudWatch registros.
7. Escolha Salvar alterações.

Depois de AWS DMS iniciar uma migração de dados, você pode interrompê-la. Para isso, escolha a migração de dados na guia Migrações de dados. Em Ações, escolha Interromper.

Depois de interromper uma migração de dados, AWS DMS defina seu status como Interrompendo. O AWS DMS define o status dessa migração de dados como Interrompida. Depois de AWS DMS interromper uma migração de dados, você pode modificar, retomar, reiniciar ou excluir sua migração de dados.

Para continuar a replicação de dados, escolha a migração de dados interrompida na guia Migrações de dados. Em Ações, escolha Retomar processamento.

Para reiniciar a carga de dados, escolha a migração de dados interrompida na guia Migrações de dados. Em seguida, em Ações, escolha Reiniciar. AWS DMS exclui todos os dados do seu banco de dados de destino e inicia a migração de dados do zero.

É possível excluir uma migração de dados interrompida ou que você ainda não iniciou. Para excluir uma migração de dados, escolha a guia Migrações de dados. Em Ações, escolha Excluir. Para excluir o projeto de migração, interrompa e exclua todas as migrações de dados.

Monitorando migrações de dados em AWS DMS

Depois de iniciar a migração de dados homogênea, é possível monitorar seu status e progresso. As migrações de dados de grandes conjuntos de dados, como centenas de gigabytes, demoram horas para serem concluídas. Para manter a confiabilidade, a disponibilidade e o alto desempenho da migração de dados, monitore o progresso regularmente.

Como verificar o status e o progresso da migração de dados

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e navegue até a guia Migrações de dados.
4. Para a migração de dados, consulte a coluna Status. Para obter mais informações sobre os valores nessa coluna, consulte Status das migrações.
5. Para uma migração de dados em execução, a coluna Progresso da migração exibe a porcentagem de dados migrados.

Como verificar os detalhes da migração de dados

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração. Na guia Migrações de dados, escolha a migração de dados.
4. Na guia Detalhes, é possível ver o progresso da migração. Em particular, é possível ver as seguintes métricas.

   - Endereço IP público: o endereço IP público da migração de dados. Você precisa desse valor para configurar uma rede. Para ter mais informações, consulte Configurar uma rede.
   - Tabelas carregadas: o número de tabelas carregadas com sucesso.
   - Tabelas sendo carregadas: o número de tabelas que estão sendo carregadas.
   - Tabelas enfileiradas: o número de tabelas que estão aguardando para serem carregadas.
   - Tabelas com erro: o número de tabelas que apresentaram falha no carregamento.
   - Tempo decorrido: a quantidade de tempo decorrido após o início da migração de dados.

- Latência do CDC — O tempo médio que passa entre a ocorrência de uma alteração em uma tabela de origem e o momento em que essa alteração AWS DMS é aplicada à tabela de destino.
- Migração iniciada: a hora em que você iniciou essa migração de dados.
- Migração interrompida: o momento em que você interrompeu essa migração de dados.

5. Para visualizar os arquivos de log da sua migração de dados, escolha Exibir CloudWatch registros em Configurações de migração de dados homogênea. Você pode ativar CloudWatch os registros ao criar ou modificar uma migração de dados. Para obter mais informações, consulte Criar uma migração de dados e Gerenciar migrações de dados.

Você pode usar CloudWatch os alarmes ou eventos da Amazon para acompanhar de perto sua migração de dados. Para obter mais informações, consulte O que são Amazon CloudWatch, Amazon CloudWatch Events e Amazon CloudWatch Logs? no Guia do CloudWatch usuário da Amazon. Observe que há uma cobrança pelo uso da Amazon CloudWatch.

Para migrações de dados homogêneas, AWS DMS inclui as seguintes métricas na Amazon. CloudWatch

Métrica	Descrição
OverallCDCLatency	A latência geral durante a fase de CDC. Para bancos de dados MySQL, essa métrica mostra o número de segundos decorridos entre a alteração no log binário de origem e a replicação dessa alteração. Para bancos de dados PostgreSQL, essa métrica mostra o número de segundos decorridos entre `last_msg_receipt_time` e `last_msg_send_time` na visualização `pg_stat_subscription` . Unidades: segundos
StorageConsumption	O armazenamento consumido pela migração de dados. Unidades: bytes

Status de migrações de dados homogêneas em AWS DMS

Para cada migração de dados que você executa, AWS DMS exibe o Status no AWS DMS console. A lista a seguir inclui os status disponíveis:

- `Creating`— AWS DMS está criando a migração de dados.
- `Ready`: a migração de dados está pronta para ser iniciada.
- `Starting`— AWS DMS está criando o ambiente sem servidor para sua migração de dados. Esse processo pode demorar até 15 minutos.
- `Load running`— AWS DMS está realizando a migração de carga total.
- `Load complete, replication ongoing`— AWS DMS completou a carga completa e agora replica as mudanças em andamento. AWS DMS usa esse status somente para migrações de dados do tipo carga total e captura de dados de alteração (CDC).
- `Replication ongoing`— AWS DMS está replicando as mudanças em andamento. AWS DMS usa esse status somente para migrações do tipo de captura de dados de alteração (CDC).
- `Reloading target`— AWS DMS está reiniciando uma migração de dados e executa o tipo de migração especificado.
- `Stopping`— AWS DMS está interrompendo a migração de dados. AWS DMS define esse status depois que você opta por interromper a migração de dados no menu Ações.
- `Stopped`— AWS DMS interrompeu a migração de dados.
- `Failed`: falha na migração de dados. Para obter mais informações, consulte os arquivos de log.

  Para visualizar os arquivos de log, escolha a migração de dados na guia Migrações de dados. Em seguida, escolha Exibir CloudWatch registros em Configurações homogêneas de migração de dados.

  > **Important**
  >
  > Você pode visualizar os arquivos de log marcando a caixa de seleção Ativar CloudWatch registros ao criar sua migração de dados.

- `Deleting`— AWS DMS está excluindo a migração de dados. AWS DMS define esse status depois que você opta por excluir a migração de dados no menu Ações.

Migração de dados de bancos de dados MySQL com migrações de dados homogêneas em AWS DMS

É possível utilizar Migração de dados homogênea para migrar um banco de dados MySQL autogerenciado para o RDS para MySQL ou o Aurora MySQL. O AWS DMS cria um ambiente com tecnologia sem servidor para a migração de dados. Para diferentes tipos de migrações de dados, AWS DMS usa diferentes ferramentas nativas de banco de dados MySQL.

Para migrações de dados homogêneas do tipo Full load, AWS DMS usa mydumper para ler dados do seu banco de dados de origem e armazená-los no disco conectado ao ambiente sem servidor. Depois de AWS DMS ler todos os dados de origem, ele usa myloader no banco de dados de destino para restaurar seus dados.

Para migrações de dados homogêneas do tipo Full load and change data capture (CDC), AWS DMS usa mydumper para ler dados do seu banco de dados de origem e armazená-los no disco conectado ao ambiente sem servidor. Depois de AWS DMS ler todos os dados de origem, ele usa myloader no banco de dados de destino para restaurar seus dados. Depois de AWS DMS concluir o carregamento completo, ele configura a replicação do log binário com a posição do log binário definida para o início do carregamento completo. Para evitar inconsistência de dados, defina o Número de tarefas como 1 para capturar o estado consistente dos dados existentes. Para ter mais informações, consulte Criar uma migração de dados.

Para migrações de dados homogêneas do tipo Captura de dados alterados (CDC), o AWS DMS precisa do Ponto de início da CDC nativo para iniciar a replicação. Se você fornecer o ponto inicial nativo do CDC, AWS DMS capturará as alterações desse ponto. Como alternativa, escolha Imediatamente nas configurações da migração de dados para capturar automaticamente o ponto de início da replicação quando a migração de dados real for iniciada.

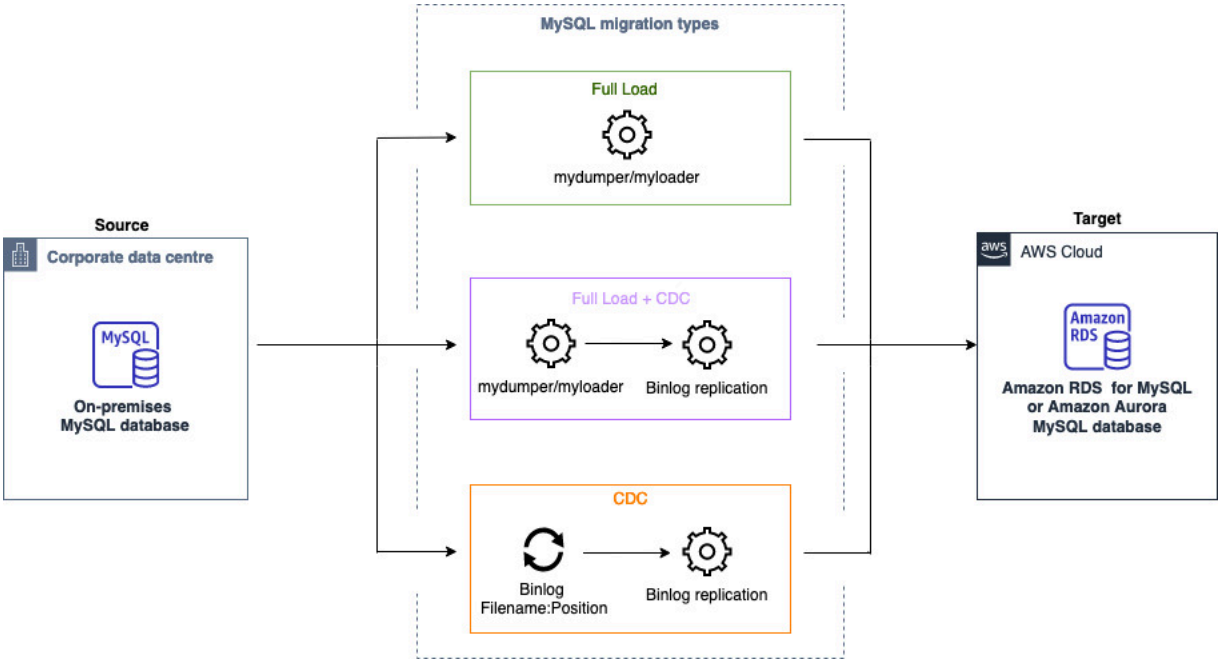
> **Note**
>
> Para que uma migração somente de CDC funcione corretamente, todos os esquemas e objetos do banco de dados de origem já devem estar presentes no banco de dados de destino. No entanto, o destino pode ter objetos que não estão presentes na origem.

É possível utilizar o exemplo de código a seguir para obter o número de sequência de log atual (LSN) no banco de dados MySQL.

```
show master status
```

Essa consulta retorna o nome e a posição de um arquivo de log binário. Para o ponto de início nativo, utilize uma combinação do nome e da posição do arquivo de log binário. Por exemplo, `mysql-bin-changelog.000024:373`. Neste exemplo, `mysql-bin-changelog.000024` é o nome do arquivo de log binário e `373` é a posição em que AWS DMS começa a capturar as alterações.

O diagrama a seguir mostra o processo de uso de migrações de dados homogêneas para migrar um banco de dados MySQL AWS DMS para o RDS for MySQL ou o Aurora MySQL.



Migração de dados de bancos de dados PostgreSQL com migrações de dados homogêneas em AWS DMS

É possível utilizar Migração de dados homogênea para migrar um banco de dados PostgreSQL autogerenciado para o RDS para PostgreSQL ou o Aurora PostgreSQL. O AWS DMS cria um ambiente com tecnologia sem servidor para a migração de dados. Para diferentes tipos de migração de dados, o AWS DMS utiliza diferentes ferramentas nativas do banco de dados do PostgreSQL.

Para migrações de dados homogêneas do tipo Full load, AWS DMS usa pg_dump para ler dados do banco de dados de origem e armazená-los no disco conectado ao ambiente sem servidor. Depois de AWS DMS ler todos os dados de origem, ele usa pg_restore no banco de dados de destino para restaurar seus dados.

Para migrações de dados homogêneas do tipo Full load and Change Data Capture (CDC), AWS DMS usa pg_dump para ler objetos de esquema sem dados de tabela do banco de dados de origem e armazená-los no disco conectado ao ambiente sem servidor. Em seguida, ele é usado `pg_restore` no banco de dados de destino para restaurar seus objetos do esquema. Depois de AWS DMS concluir o `pg_restore` processo, ele muda automaticamente para um modelo de editor e assinante para replicação lógica com a `Initial Data Synchronization` opção de copiar os dados iniciais da tabela diretamente do banco de dados de origem para o banco de dados de destino e, em seguida, inicia a replicação contínua. Nesse modelo, um ou mais assinantes assinam uma ou mais publicações em um nó do publicador.

Para migrações de dados homogêneas do tipo Change data capture (CDC), é AWS DMS necessário o ponto de partida nativo para iniciar a replicação. Se você fornecer o ponto inicial nativo, AWS DMS capturará as alterações desse ponto. Como alternativa, escolha Imediatamente nas configurações da migração de dados para capturar automaticamente o ponto de início da replicação quando a migração de dados real for iniciada.

> **Note**
>
> Para que uma migração somente de CDC funcione corretamente, todos os esquemas e objetos do banco de dados de origem já devem estar presentes no banco de dados de destino. No entanto, o destino pode ter objetos que não estão presentes na origem.

É possível utilizar o exemplo de código a seguir para obter o ponto de início nativo no banco de dados do PostgreSQL.

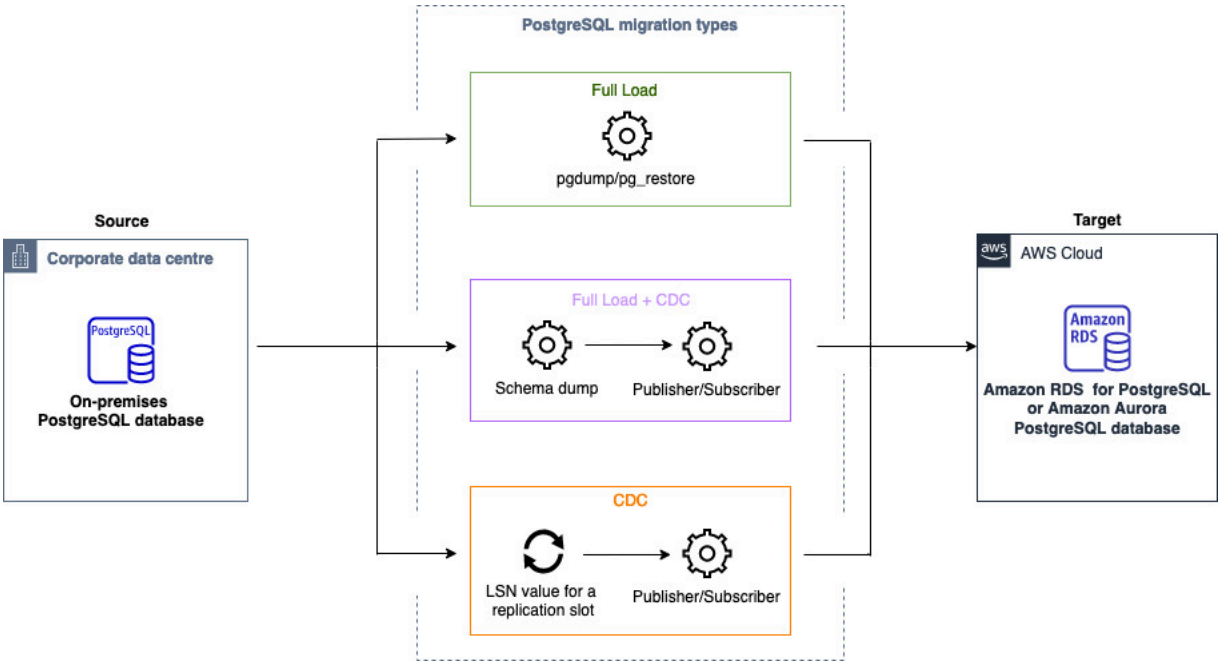
```
select confirmed_flush_lsn from pg_replication_slots where
 slot_name='migrate_to_target';
```

Essa consulta utiliza a visualização `pg_replication_slots` no banco de dados do PostgreSQL para capturar o valor do número de sequência de log (LSN).

Depois de AWS DMS definir o status da migração homogênea de dados do PostgreSQL como Parada, Falha ou Excluída, o editor e a replicação não serão removidos. Se você não quiser retomar a migração, exclua o slot de replicação e o publicador utilizando o comando a seguir.

```
SELECT pg_drop_replication_slot('migration_subscriber_{ARN}');
            DROP PUBLICATION publication_{ARN};
```

O diagrama a seguir mostra o processo de uso de migrações de dados homogêneas para migrar um banco de dados PostgreSQL AWS DMS para o RDS for PostgreSQL ou Aurora PostgreSQL.



Migração de dados de bancos de dados MongoDB com migrações de dados homogêneas em AWS DMS

Você pode usar Migração de dados homogênea para migrar um banco de dados MongoDB autogerenciado para o Amazon DocumentDB. AWS DMS cria um ambiente sem servidor para sua migração de dados. Para diferentes tipos de migrações de dados, AWS DMS usa diferentes ferramentas nativas de banco de dados MongoDB.

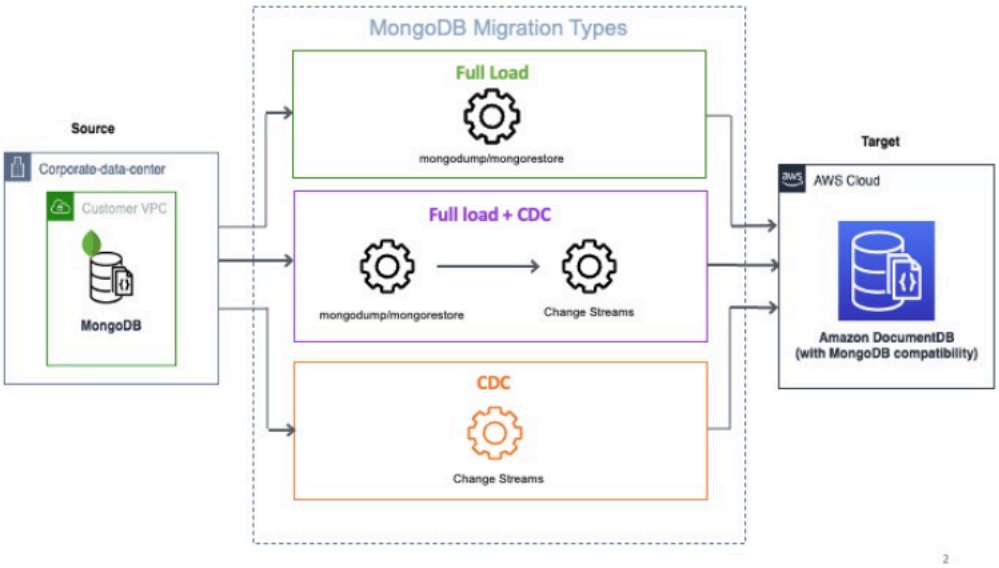
Para migrações de dados homogêneas do tipo Full load, AWS DMS usa `mongodump` para ler dados do seu banco de dados de origem e armazená-los no disco conectado ao ambiente sem servidor. Depois de AWS DMS ler todos os dados de origem, eles são usados `mongorestore` no banco de dados de destino para restaurar seus dados.

Para migrações de dados homogêneas do tipo Full Load and Change Data Capture (CDC), AWS DMS usa `mongodump` para ler dados do seu banco de dados de origem e armazená-los no disco conectado ao ambiente sem servidor. Depois de AWS DMS ler todos os dados de origem, eles são usados `mongorestore` no banco de dados de destino para restaurar seus dados. Depois de AWS DMS concluir a carga completa, ele muda automaticamente para um modelo de editor e assinante para replicação lógica. Nesse modelo, recomendamos dimensionar o oplog para reter as alterações por pelo menos 24 horas.

Para migrações de dados homogêneas do tipo Change data capture (CDC), escolha `immediately` nas configurações de migração de dados capturar automaticamente o ponto inicial da replicação quando a migração de dados real começar.

> **Note**
>
> Para qualquer coleção nova ou renomeada, você precisa criar uma nova tarefa de migração de dados para essas coleções como migrações de dados homogêneas. Para uma fonte compatível com MongoDB, AWS DMS não oferece suporte `create` e operações. `rename` `drop collection`

O diagrama a seguir mostra o processo de uso de migrações de dados homogêneas para migrar um banco de dados MongoDB AWS DMS para o Amazon DocumentDB.



Solução de problemas para migrações de dados homogêneas no AWS DMS

Na lista a seguir, é possível encontrar ações a serem tomadas ao encontrar problemas com migrações de dados homogêneas no AWS DMS.

Tópicos

- Não é possível criar uma migração de dados homogênea no AWS DMS
- Não é possível iniciar uma migração de dados homogênea no AWS DMS

- Não é possível me conectar ao banco de dados de destino ao executar uma migração de dados no AWS DMS
- O AWS DMS migra visualizações como tabelas no PostgreSQL

Não é possível criar uma migração de dados homogênea no AWS DMS

Se você receber uma mensagem de erro dizendo que AWS DMS não pode se conectar aos provedores de dados depois de escolher Criar migração de dados, verifique se você configurou o perfil do IAM necessário. Para obter mais informações, consulte Criar um perfil do IAM.

Se você configurou o perfil do IAM e ainda receber essa mensagem de erro, adicione esse perfil do IAM ao usuário de chaves na configuração da chave do AWS KMS. Para obter mais informações, consulte Permitir que usuários de chaves utilizem a chave do KMS no Guia do desenvolvedor do AWS Key Management Service.

Não é possível iniciar uma migração de dados homogênea no AWS DMS

Se você obtiver o status `Failed` ao iniciar uma migração de dados no projeto de migração, verifique as versões dos provedores de dados de origem e de destino. Para isso, execute a consulta `SELECT VERSION();` no banco de dados MySQL ou PostgreSQL. Utilize a versão compatível do banco de dados.

Para obter a lista dos bancos de dados de origem compatíveis, consulte Origens para migrações de dados homogêneas do DMS.

Para obter a lista dos bancos de dados de destino compatíveis, consulte Destinos para migrações de dados homogêneas do DMS.

Se você utilizar uma versão de banco de dados não compatível, atualize o seu banco de dados de origem ou de destino e tente novamente.

Verifique a mensagem de erro da migração de dados no console do AWS DMS. Para fazer isso, abra o projeto de migração e escolha a migração de dados. Na guia Detalhes, verifique a Última mensagem de falha em Geral.

Por fim, analise o log do CloudWatch. Para fazer isso, abra o projeto de migração e escolha a migração de dados. Na guia Detalhes, escolha Visualizar logs do CloudWatch.

Não é possível me conectar ao banco de dados de destino ao executar uma migração de dados no AWS DMS

Se você receber a mensagem de erro Não é possível se conectar ao destino, execute as seguintes ações.

1. Verifique se o grupo de segurança anexado aos bancos de dados de origem e destino contém uma regra para qualquer tráfego de entrada e de saída. Para obter mais informações, consulte Configurar a replicação contínua de dados.
2. Verifique a lista de controle de acesso (ACL) de rede e as regras da tabela de rotas.
3. O banco de dados deve estar acessível na VPC criada. Adicione endereços IP públicos aos grupos de segurança da VPC e permita conexões de entrada no firewall.
4. Na guia Migrações de dados do projeto de migração, escolha a migração de dados. Anote o Endereço IP público em Conectividade e segurança na guia Detalhes. Permita acesso do endereço IP público da migração de dados nos bancos de dados de origem e de destino.
5. Para a replicação contínua de dados, verifique se os bancos de dados de origem e de destino podem se comunicar entre si.

Para mais informações, consulte Controlar o tráfego para recursos usando grupos de segurança no Guia do usuário da Amazon Virtual Private Cloud.

O AWS DMS migra visualizações como tabelas no PostgreSQL

A migração homogênea de dados não é compatível com a migração de visualizações como visualizações no PostgreSQL. No PostgreSQL, o AWS DMS migra visualizações como tabelas.

Trabalhando com provedores de dados, perfis de instância e projetos de migração no AWS DMS

Ao usar o DMS Schema Conversion e migrações de dados homogêneas no AWS Database Migration Service, você trabalha com projetos de migração. Por sua vez, os projetos de migração AWS DMS utilizam grupos de sub-redes, perfis de instância e provedores de dados.

Uma sub-rede é um intervalo de endereços IP na VPC. Um grupo de sub-redes de replicação inclui sub-redes de diferentes zonas de disponibilidade que seu perfil de instância pode usar. Observe que um grupo de sub-redes de replicação é um recurso do DMS e é diferente dos grupos de sub-redes que a Amazon VPC e o Amazon RDS usam.

Um perfil de instância especifica as configurações de rede e de segurança do ambiente com tecnologia sem servidor em que o projeto de migração é executado.

Um provedor de dados armazena um tipo de datastore e as informações de localização do banco de dados. Depois de adicionar um provedor de dados ao seu projeto de migração, você fornece as credenciais do banco de dados de AWS Secrets Manager. AWS DMS usa essas informações para se conectar ao seu banco de dados.

Depois de criar provedores de dados, seu perfil de instância e outros AWS recursos, você pode criar um projeto de migração. Um projeto de migração descreve o perfil da instância, os provedores de dados de origem e destino e os segredos de AWS Secrets Manager. É possível criar vários projetos de migração para diferentes provedores de dados de origem e de destino.

A maior parte do trabalho é executada no projeto de migração. Na DMS Schema Conversion, você utiliza um projeto de migração para avaliar os objetos do provedor de dados de origem e convertê-los em um formato compatível com o banco de dados de destino. É possível aplicar o código convertido ao provedor de dados de destino ou salvá-lo como um script SQL. Em migrações de dados homogêneas, você utiliza um projeto de migração para migrar dados do banco de dados de origem para um banco de dados de destino do mesmo tipo na Nuvem AWS.

Os projetos de migração AWS DMS são somente sem servidor. AWS DMS provisiona automaticamente os recursos de nuvem para seus projetos de migração.

AWS DMS tem o número máximo de perfis de instância, provedores de dados e projetos de migração que você pode criar para o seu Conta da AWS. Consulte a seção a seguir para obter

informações sobre as cotas de serviço do Cotas para o AWS Database Migration Service do AWS DMS .

Tópicos

Criação de um grupo de sub-redes para um projeto de AWS DMS migração

Antes de criar um perfil de instância, configure um grupo de sub-redes para o perfil de instância.

Como criar um grupo de sub-redes

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, escolha Grupos de sub-redes e Criar sub-rede.
3. Em Nome, insira um nome exclusivo do grupo de sub-redes.
4. Em Descrição, insira uma descrição breve do grupo de sub-redes.
5. Em VPC, escolha uma VPC que tenha pelo menos uma sub-rede em pelo menos duas zonas de disponibilidade.
6. Em Adicionar sub-redes, escolha as sub-redes a serem incluídas no grupo de sub-redes. Selecione sub-redes em, pelo menos, duas zonas de disponibilidade.

   Para conectar-se aos bancos de dados do Amazon RDS, adicione sub-redes públicas ao grupo de sub-redes. Para conectar-se aos bancos de dados on-premises, adicione sub-redes privadas ao grupo de sub-redes.
7. Selecione Create subnet group (Criar grupo de sub-redes).

Criação de perfis de instância para AWS Database Migration Service

Você pode criar vários perfis de instância no AWS DMS console. Selecione um perfil de instância para utilização em cada projeto de migração criado no AWS DMS.

Como criar um perfil de instância

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, escolha Perfis de instância.
3. Escolha Criar perfil de instância.
4. Na página Criar perfil de instância, insira um valor descritivo em Nome para o perfil de instância.
5. Em Tipo de rede, escolha o Modo Dual-Stack para criar um perfil de instância compatível com o endereçamento IPv4 e IPv6. Mantenha a opção padrão para criar um perfil de instância compatível somente com o endereçamento IPv4.
6. Escolha Nuvem privada virtual (VPC) para executar a instância do tipo de rede selecionado. Escolha um Grupo de sub-redes e um Grupo de segurança da VPC para o perfil de instância.

   Para conectar-se aos bancos de dados do Amazon RDS, utilize um grupo de sub-redes que inclua sub-redes públicas. Para conectar-se aos bancos de dados on-premises, utilize um grupo de sub-redes que inclua sub-redes privadas. Certifique-se de ter configurado sua rede para que AWS DMS possa acessar seu banco de dados local de origem usando o endereço IP público do gateway NAT. Para ter mais informações, consulte Criar uma VPC com base em uma Amazon VPC.
7. (Opcional) Se você criar um projeto de migração para a DMS Schema Conversion, em Configurações de conversão de esquemas - opcional, escolha um bucket do Amazon S3 para armazenar as informações do projeto de migração. Em seguida, escolha a função AWS Identity and Access Management (IAM) que fornece acesso a esse bucket do Amazon S3. Para ter mais informações, consulte Criar um bucket do Amazon S3.
8. Escolha Criar perfil de instância.

Depois de criar o perfil de instância, é possível modificá-lo ou excluí-lo.

Como modificar um perfil de instância

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Perfis de instância. A página Perfis de instância é aberta.
3. Escolha o perfil de instância e escolha Modificar.
4. Atualize o nome do perfil de instância, edite as configurações do bucket da VPC ou do Amazon S3.
5. Escolha Salvar alterações.

Como excluir um perfil de instância

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Perfis de instância. A página Perfis de instância é aberta.
3. Escolha o perfil de instância e Excluir.
4. Escolha Excluir para confirmar a opção.

Criação de provedores de dados em AWS Database Migration Service

Você pode criar provedores de dados e usá-los em projetos de AWS DMS migração. O provedor de dados pode ser um mecanismo autogerenciado executado on-premises ou em uma instância do Amazon EC2. Além disso, o provedor de dados pode ser um mecanismo totalmente gerenciado, como o Amazon Relational Database Service (Amazon RDS) ou o Amazon Aurora.

Para cada banco de dados, é possível criar um único provedor de dados. Um único provedor de dados pode ser utilizado em vários projetos de migração.

Antes de criar um projeto de migração, crie pelo menos dois provedores de dados. Um dos provedores de dados deve estar em um AWS service (Serviço da AWS). Não é possível utilizar o AWS DMS para converter esquemas ou migrar os dados para um banco de dados on-premises.

O procedimento a seguir mostra como criar provedores de dados no assistente do AWS DMS console.

Como criar um provedor de dados

1. Faça login no e AWS Management Console, em seguida, abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha provedores de dados. A página Provedores de dados é aberta.
3. Escolha Criar provedor de dados. A tabela a seguir descreve as configurações.

Opção	Ação
Configuração	Escolha se deseja inserir manualmente informações sobre o provedor de dados ou utilizar a instância de banco de dados Amazon RDS.
Nome	Insira um nome para o provedor de dados. Utilize um nome exclusivo para o provedor de dados para que você possa identificá-lo com facilidade.
Tipo de mecanismo	Escolha o tipo de mecanismo de banco de dados do provedor de dados.
Nome do servidor	Insira o nome do serviço de nomes de domínio (DNS) ou o endereço IP do servidor de banco de dados. O nome do servidor de um provedor de dados usado para replicação homogênea deve começar com um caractere alfanumérico e só pode conter caracteres alfanuméricos, hifens (-), pontos (.) ou sublinhados (_).
Porta	Insira a porta utilizada para conectar-se ao servidor de banco de dados.
ID do serviço (SID) ou nome do serviço	Insira o ID do sistema do Oracle (SID). Para encontrar o Oracle SID, envie a consulta a seguir para seu banco de dados Oracle: `SELECT sys_context('userenv','instance_name') AS SID FROM dual;`

Opção	Ação
Database name	Insira o nome do banco de dados desse provedor de dados. O nome do banco de dados de um provedor de dados usado para uma replicação homogênea pode ter até 63 caracteres e não pode conter espaços.
Modo Secure Socket Layer (SSL)	Escolha um modo SSL se quiser ativar a criptografia de conexão desse provedor de dados. Dependendo do modo selecionado, talvez seja necessário fornecer informaçõ es sobre o certificado e sobre o certificado do servidor. Para obter mais detalhes, consulte Usando SSL com AWS Database Migration Service.
Modo de autenticação	Para uma fonte do MongoDB, o modo de autenticação usado para autenticar a AWS DMS conexão do endpoint.
Origem da autenticação	Para uma fonte do MongoDB, o nome do banco de dados MongoDB a ser usado para validar suas credenciais para autenticação.
Mecanismo de autenticação	Para uma fonte do MongoDB, o método de autenticação que o MongoDB usa para criptografar a senha.

4. Escolha Criar provedor de dados.

Depois de criar um provedor de dados, adicione as credenciais de conexão do banco de dados no AWS Secrets Manager.

Criação de projetos de migração em AWS Database Migration Service

Antes de criar um projeto de migração no AWS DMS, certifique-se de criar os seguintes recursos:

- Provedores de dados que descrevem os bancos de dados de origem e de destino
- Segredos com credenciais de banco de dados armazenadas em AWS Secrets Manager
- A função AWS Identity and Access Management (IAM) que fornece acesso ao Secrets Manager

- Um perfil de instância que inclui configurações de rede e de segurança

Como criar um projeto de migração

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha Criar projeto de migração. A tabela a seguir descreve as configurações.

Opção	Ação
Nome	Insira um nome para o projeto de migração. Utilize um nome exclusivo para o projeto de migração para que você possa identificá-lo com facilidade.
Perfil de instância	Escolha o perfil de instância a ser utilizado no projeto de migração.
Origem	Escolha Procurar e escolha o provedor de dados de origem.
ID do segredo	Escolha o nome do recurso da Amazon (ARN) do segredo no Secrets Manager que armazena as credenciais do banco de dados de origem.
Perfil do IAM	Escolha um perfil do IAM para fornecer acesso às credencia is do banco de dados de origem no Secrets Manager.
Destino	Escolha Procurar e escolha o provedor de dados de origem.
ID do segredo	Escolha o ARN do segredo no Secrets Manager que armazena as credenciais do banco de dados de destino.
Perfil do IAM	Escolha um perfil do IAM para fornecer acesso às credencia is do banco de dados de destino no Secrets Manager.
Regras de transformação	(Opcional) Se você criar um projeto de migração para a DMS Schema Conversion, escolha Adicionar regra de transform ação para configurar as regras de transformação. As regras

Opção	Ação
	de transformação permitem alterar os nomes dos objetos de acordo com a regra especificada. Para ter mais informações, consulte Configurar regras de transformação.

4. Escolha Criar projeto de migração.

Depois de AWS DMS criar seu projeto de migração, você pode usar esse projeto na conversão de esquema DMS ou em migrações de dados homogêneas. Para começar a trabalhar com o projeto de migração, na página Projetos de migração, escolha o projeto na lista.

Gerenciando projetos de migração em AWS Database Migration Service

Depois de criar o projeto de migração, é possível modificá-lo ou excluí-lo. Por exemplo, para alterar o provedor de dados de origem ou de destino, modifique o projeto de migração.

É possível modificar ou excluir o projeto de migração somente depois de fechar as operações de conversão de esquema ou de migração de dados. Para fazer isso, escolha o projeto de migração na lista e Conversão de esquemas ou Migrações de dados. Escolha Fechar conversão de esquema da DMS Schema Conversion e confirme a opção. Para migrações de dados homogêneas, escolha a migração de dados e escolha Parar no menu Ações. Depois de editar o projeto de migração, é possível iniciar a conversão de esquemas ou iniciar a migração de dados novamente.

Como modificar um projeto de migração

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e Modificar.
4. Atualize o nome do projeto, edite o perfil de instância ou altere os provedores de dados de origem e de destino. Opcionalmente, adicione ou edite as regras de migração que alteram os nomes dos objetos durante a conversão.
5. Escolha Salvar alterações.

Como excluir um projeto de migração

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Projetos de migração. A página Projetos de migração é aberta.
3. Escolha o projeto de migração e Excluir.
4. Escolha Excluir para confirmar a opção.

Práticas recomendadas do AWS Database Migration Service

Para utilizar o AWS Database Migration Service (AWS DMS) de maneira mais eficiente, consulte as recomendações desta seção sobre a maneira mais eficaz de migrar dados.

Tópicos

Planejamento de migração do AWS Database Migration Service

Ao planejar uma migração de banco de dados utilizando o AWS Database Migration Service, considere o seguinte:

- Para conectar os bancos de dados de origem e de destino a uma instância de replicação do AWS DMS, configure uma rede. Isso pode ser tão simples quanto conectar dois recursos da AWS na mesma nuvem privada virtual (VPC) que a instância de replicação. Isso pode variar para configurações mais complexas, como conectar um banco de dados on-premises a uma instância de banco de dados Amazon RDS por meio de uma rede privada virtual (VPN). Para ter mais informações, consulte Configurações de rede para migração de banco de dados.
- Endpoints de origem e de destino: é necessário saber quais informações e tabelas do banco de dados de origem devem ser migradas para o banco de dados de destino. O AWS DMS é compatível com a migração básica de esquemas, incluindo a criação de tabelas e de chaves primárias. No entanto, o AWS DMS não cria automaticamente índices secundários, chaves estrangeiras, contas de usuário etc. no banco de dados de destino. Dependendo do mecanismo dos bancos de dados de origem e de destino, poderá ser necessário configurar registro em log suplementar ou modificar outras configurações de um banco de dados de origem ou de destino. Para obter mais informações, consulte Destinos para a migração de dados e Origens para a migração de dados.
- Migração de esquema e de código: o AWS DMS não faz conversão de esquemas ou de código. É possível utilizar ferramentas, como o Oracle SQL Developer, o MySQL Workbench e o pgAdmin III para converter o esquema. Para converter um esquema existente em um mecanismo de banco de dados diferente, utilize a AWS Schema Conversion Tool (AWS SCT). Ela pode criar um esquema de destino e gerar e criar um esquema inteiro: tabelas, índices, visualizações e assim por diante. Também é possível utilizar a ferramenta para converter PL/SQL ou TSQL para PgSQL e outros formatos. Para obter mais informações sobra a AWS SCT, consulte o Guia do usuário da AWS SCT.
- Tipos de dados incompatíveis: verifique se é possível converter tipos de dados de origem em tipos de dados equivalentes para o banco de dados de destino. Para obter mais informações sobre os tipos de dados compatíveis, consulte a seção de origem ou de destino do datastore.
- Resultados do script de apoio de diagnóstico: ao planejar a migração, é recomendável executar scripts de apoio de diagnóstico. Com os resultados desses scripts, é possível encontrar informações sobre possíveis falhas na migração com antecedência.

  Se um script de apoio estiver disponível para o banco de dados, baixe-o utilizando o link no tópico do script correspondente na seção a seguir. Depois de verificar e analisar o script, é possível executá-lo de acordo com o procedimento descrito no tópico do script em seu ambiente on-premises. Quando a execução do script for concluída, será possível analisar os resultados. É recomendável executar esses scripts como a primeira etapa de qualquer tentativa de solução de

problemas. Os resultados podem ser úteis ao trabalhar com uma equipe de AWS Support. Para ter mais informações, consulte Como trabalhar com scripts de suporte a diagnóstico no AWS DMS.

- Avaliações de pré-migração: uma avaliação de pré-migração avalia componentes especificados de uma tarefa de migração de banco de dados para ajudar a identificar quaisquer problemas que possam impedir que uma tarefa de migração seja executada conforme o esperado. Ao utilizar essa avaliação, é possível identificar problemas potenciais antes de executar uma tarefa nova ou modificada. Para obter mais informações sobre como trabalhar com avaliações de pré-migração, consulte Ativar e trabalhar com avaliações de pré-migração de uma tarefa.

Conversão de esquemas

O AWS DMS não executa a conversão de esquema ou de código. Para converter um esquema existente em um mecanismo de banco de dados diferente, é possível utilizar a AWS SCT. A AWS SCT converte os objetos, as tabelas, os índices, as visualizações, os acionadores da origem e outros objetos do sistema no formato DDL (linguagem de definição de dados) do destino. Também é possível utilizar a AWS SCT para converter a maior parte do código da aplicação, como PL/SQL ou TSQL, para a linguagem equivalente do destino.

É possível obter a AWS SCT para download gratuito na AWS. Para obter mais informações sobre a AWS SCT, consulte o Guia do usuário da AWS SCT.

Se seus endpoints de origem e destino estiverem no mesmo mecanismo de banco de dados, você poderá usar ferramentas como Oracle SQL Developer, MySQL Workbench PgAdmin ou 4 para mover seu esquema.

Análise da documentação pública do AWS DMS

É altamente recomendável consultar as páginas da documentação pública do AWS DMS de seus endpoints de origem e de destino antes da primeira migração. Essa documentação pode ajudar a identificar os pré-requisitos da migração e a compreender as limitações atuais antes de você começar. Para ter mais informações, consulte Como trabalhar com endpoints do AWS DMS.

Durante a migração, a documentação pública pode ajudar a solucionar qualquer problema com o AWS DMS. As páginas de solução de problemas na documentação podem ajudar a resolver problemas comuns utilizando o AWS DMS e os bancos de dados de endpoints selecionados. Para ter mais informações, consulte Solução de problemas de tarefas de migração no AWS Database Migration Service.

Execução de uma prova de conceito

Para ajudar a descobrir problemas no ambiente nas fases iniciais da migração de banco de dados, é recomendável executar uma pequena migração de teste. Isso também pode ajudar a definir um cronograma de migração mais realista. Além disso, talvez seja necessário executar uma migração de teste em escala total para avaliar se o AWS DMS pode lidar com o throughput do banco de dados na rede. Durante esse período, é recomendável comparar e otimizar a carga máxima inicial e a replicação contínua. Isso pode ajudar a compreender a latência da rede e avaliar o desempenho geral.

Nesse ponto, você também tem a oportunidade de compreender o perfil dos dados e o tamanho do banco de dados, incluindo o seguinte:

- O número de tabelas grandes, médias e pequenas.
- Como o AWS DMS lida com conversões de tipos de dados e conjuntos de caracteres.
- A quantidade de tabelas com colunas de objetos grandes (LOB).
- Quanto tempo é necessário para executar uma migração de teste.

Aprimoramento do desempenho de uma migração do AWS DMS

Diversos fatores afetam o desempenho da migração do AWS DMS:

- Disponibilidade de recursos na origem.
- O throughput disponível da rede.
- A capacidade de recursos do servidor de replicação.
- A capacidade de ingestão de alterações pelo destino.
- O tipo e a distribuição dos dados de origem.
- O número de objetos a serem migrados.

É possível melhorar o desempenho utilizando algumas ou todas as práticas recomendadas mencionadas a seguir. A possibilidade de utilizar uma dessas práticas depende do caso de uso específico. As limitações a seguir podem ser encontradas:

Provisionar um servidor de replicação adequado

O AWS DMS é um serviço gerenciado executado em uma instância do Amazon EC2. O serviço se conecta ao banco de dados de origem, lê os dados de origem, formata os dados para consumo do banco de dados de destino e carrega os dados nesse banco de dados.

A maior parte desse processo ocorre na memória. No entanto, transações grandes podem exigir buffer no disco. Transações armazenadas em cache e arquivos de log também são gravados no disco. Nas seções a seguir, é possível encontrar o que deve ser considerado ao escolher o servidor de replicação.

CPU

O AWS DMS foi projetado para migrações heterogêneas, mas também é compatível com migrações homogêneas. Para executar uma migração homogênea, primeiro converta cada tipo de dados de origem no tipo de dados equivalente do AWS DMS. Converta cada tipo de dados do AWS DMS no tipo de dados de destino. É possível encontrar referências a essas conversões para cada mecanismo de banco de dados no Guia do usuário do AWS DMS.

Para que o AWS DMS execute essas conversões de forma ideal, a CPU deve estar disponível quando as conversões ocorrerem. Sobrecarregar a CPU e não ter recursos suficientes de CPU pode resultar em migrações lentas, o que também pode causar outros efeitos colaterais.

Classe da instância de replicação

Algumas das classes de instância menores são suficientes para testar o serviço ou para migrações pequenas. Se a migração envolver um grande número de tabelas ou se você quiser executar várias tarefas de replicação simultâneas, considere utilizar uma das instâncias maiores. Uma instância maior pode ser uma boa ideia porque o serviço consome uma boa quantidade de memória e de CPU.

As instâncias do tipo T2 são projetadas para fornecer desempenho de linha de base moderado e capacidade de intermitência para obter desempenho significativamente mais alto, conforme necessário para a workload. Elas são destinadas a workloads que não utilizam toda a CPU com frequência ou de forma consistente, mas que às vezes precisam de intermitência. As instâncias T2 são ideais para workloads de uso geral, como servidores web, ambientes de desenvolvedor e bancos de dados pequenos. Se estiver solucionando problemas de uma migração lenta e utilizando um tipo de instância T2, verifique a métrica de utilização de CPU do host. Ela pode mostrar se você está ultrapassando a linha de base desse tipo de instância.

As classes de instância C4 são projetadas para fornecer o mais alto nível de desempenho do processador para workloads de consumo intensivo. Elas alcançam desempenho significativamente mais alto de pacotes por segundo (PPS), jitter de rede mais baixo e latência de rede mais baixa. O AWS DMS pode consumir muita CPU, principalmente ao executar migrações e replicações heterogêneas, como migrar do Oracle para o PostgreSQL. As instâncias C4 podem ser uma boa opção para essas situações.

As classes de instância R4 são otimizadas para memória para workloads de consumo intensivo de memória. As replicações ou migrações contínuas de sistemas de transação de alto throughput que usam o AWS DMS, às vezes, podem consumir grandes quantidades de CPU e de memória. As instâncias R4 incluem mais memória por vCPU.

Suporte do AWS DMS para classes de instâncias R5 e C5

As classes de instâncias R5 otimizadas para memória são projetadas para fornecer desempenho rápido para workloads que processam grandes conjuntos de dados na memória. As replicações ou migrações contínuas de sistemas de transação de alto throughput que usam o AWS DMS, às vezes, podem consumir grandes quantidades de CPU e de memória. As instâncias R5 fornecem 5% a mais de memória por vCPU do que as R4, e o tamanho maior fornece 768 GiB de memória. Além disso, as instâncias R5 fornecem uma melhoria de 10% no preço por GiB e um aumento de aproximadamente 20% no desempenho da CPU em relação às R4.

As classes de instância C5 são otimizadas para cargas de trabalho com uso intensivo de computação e oferecem alto desempenho econômico a um baixo preço por taxa de computação. Elas alcançam um desempenho de rede significativamente maior. O Adaptador de Rede Elástica (ENA) fornece instâncias C5 com até 25 Gbps de largura de banda de rede e até 14 Gbps de largura de banda dedicada para o Amazon EBS. O AWS DMS pode ter um consumo intensivo de CPU, especialmente ao executar migrações e replicações heterogêneas, como a migração do Oracle para o PostgreSQL. As instâncias C5 podem ser uma boa opção para essas situações.

Armazenamento

Dependendo da classe da instância, o servidor de replicação vem com 50 GB ou 100 GB de armazenamento de dados. Esse armazenamento é usado para arquivos de log e para quaisquer alterações em cache coletadas durante a carga. Se o sistema de origem estiver ocupado ou realizar grandes transações, talvez seja necessário aumentar o armazenamento. Ao executar várias tarefas no servidor de replicação, talvez também seja necessário aumentar o armazenamento. No entanto, o valor padrão é geralmente suficiente.

Todos os volumes de armazenamento no AWS DMS são unidades GP2 ou de estado sólido de uso geral (SSDs). Os volumes GP2 vêm com um desempenho básico de três operações de E/S por segundo (IOPS), com capacidade de intermitência até 3.000 IOPS com base em crédito. Como regra geral, verifique as métricas `ReadIOPS` e `WriteIOPS` da instância de replicação. Verifique se a soma desses valores não ultrapassa o desempenho básico desse volume.

Multi-AZ

A escolha de uma instância multi-AZ pode proteger a migração contra falhas de armazenamento. A maioria das migrações é transitória e não se destina a ser executada por longos períodos. Ao utilizar o AWS DMS para fins de replicação contínua, a escolha de uma instância multi-AZ pode melhorar a disponibilidade, caso ocorra um problema de armazenamento.

Se ocorrer um failover ou uma substituição do host ao utilizar uma instância de replicação de uma única AZ ou multi-AZ durante uma CARGA MÁXIMA, a tarefa de carga máxima falhará. É possível reiniciar a tarefa a partir do ponto de falha para as tabelas restantes que não foram concluídas ou que estão em estado de erro.

Carregamento de várias tabelas em paralelo

Por padrão, o AWS DMS carrega oito tabelas por vez. É possível ver uma melhora no desempenho aumentando um pouco esse número ao utilizar um servidor de replicação muito grande, como uma instância dms.c4.xlarge ou maior. Contudo, em algum momento, o aumento do paralelismo reduz o desempenho. Se o servidor de replicação for relativamente pequeno, como um dms.t2.medium, é recomendável reduzir o número de tabelas carregadas em paralelo.

Para alterar esse número no AWS Management Console, abra o console, escolha Tarefas, escolha criar ou modificar uma tarefa e Configurações avançadas. Em Configurações de ajuste, altere a opção Número máximo de tabelas para carga em paralelo.

Para alterar esse número utilizando a AWS CLI, altere o parâmetro `MaxFullLoadSubTasks` em `TaskSettings`.

Utilização da carga máxima em paralelo

É possível utilizar uma carga paralela de origens do Oracle, do Microsoft SQL Server, do MySQL, do Sybase e do IBM Db2 LUW com base em partições e subpartições. Isso pode melhorar a duração geral da carga máxima. Além disso, ao executar uma tarefa de migração do AWS DMS, é possível acelerar a migração de tabelas grandes ou particionadas. Para isso, divida a tabela em segmentos e carregue os segmentos em paralelo na mesma tarefa de migração.

Para utilizar uma carga paralela, crie uma regra de mapeamento de tabelas do tipo `table-settings` com a opção `parallel-load`. Na regra `table-settings`, especifique os critérios de seleção da tabela ou das tabelas que você quer carregar em paralelo. Para especificar os critérios de seleção, defina o elemento `type` da `parallel-load` para uma das seguintes configurações:

- `partitions-auto`
- `subpartitions-auto`
- `partitions-list`
- `ranges`
- `none`

Para obter mais informações sobre essas configurações, consulte Regras e operações de configurações de tabelas e coleções.

Como trabalhar com índices, acionadores e restrições de integridade referencial

Índices, acionadores e restrições de integridade referencial podem afetar o desempenho da migração e fazer com que ela falhe. O modo como esses itens afetam a migração depende de se a tarefa de replicação é uma tarefa de carga máxima ou de replicação contínua (captura de dados de alteração ou CDC).

Para uma tarefa de carga máxima, é recomendável ignorar os índices de chave primária, os índices secundários, as restrições de integridade referencial e os acionadores da linguagem de manipulação de dados (DML). Ou você pode atrasar a criação até que as tarefas de carga máxima sejam concluídas. Os índices não são necessários durante uma tarefa de carga máxima e incorrerão em sobrecarga de manutenção se estiverem presentes. Como a tarefa de carga máxima carrega grupos de tabelas por vez, as restrições de integridade referencial são violadas. Do mesmo modo, a inserção, a atualização e a exclusão de acionadores pode causar erros se, por exemplo, uma inserção de linha for acionada para uma tabela carregada em massa anteriormente. Outros tipos de acionadores também afetam o desempenho devido ao processamento adicionado.

Se os volumes de dados forem relativamente pequenos e o tempo de migração adicional não for uma preocupação, será possível criar índices de chave primária e secundários antes de uma tarefa de carga máxima Sempre desative as restrições de integridade referencial e os acionadores.

Para uma tarefa de carga máxima mais CDC, é recomendável adicionar índices secundários antes da fase de CDC. Como o AWS DMS utiliza replicação lógica, verifique se os índices

secundários compatíveis com operações DML estão presentes para evitar varreduras de tabelas inteiras. Pause a tarefa de replicação antes da fase de CDC para criar índices e crie restrições de integridade referencial antes de reiniciar a tarefa.

Você deve ativar os acionadores logo antes da substituição.

Desativação dos backups e do log de transações

Ao migrar para um banco de dados Amazon RDS, convém desativar backups e multi-AZ no destino até que você esteja pronto para a transferência. Da mesma forma, ao migrar para sistemas que não sejam o Amazon RDS, convém desativar qualquer registro em log no destino até depois da substituição.

Utilização de várias tarefas

Às vezes, utilizar várias tarefas para uma única migração pode melhorar o desempenho. Quando houver conjuntos de tabelas que não participam de transações comuns, talvez seja possível dividir a migração em várias tarefas. A consistência transacional é mantida em uma tarefa, portanto, é importante que tabelas em tarefas separadas não participem de transações comuns. Além disso, cada tarefa lê o fluxo de transações de forma independente; portanto, tenha cuidado para não sobrecarregar o banco de dados de origem.

É possível utilizar várias tarefas para criar fluxos separados de replicação. Ao fazer isso, é possível paralelizar as leituras na origem, os processos na instância de replicação e as gravações no banco de dados de destino.

Otimização do processamento de alterações

Por padrão, o AWS DMS processa alterações em um modo transacional, que preserva a integridade transacional. Se houver condições para lapsos temporários em integridade transacional, você poderá usar a opção batch optimized apply. Essa opção agrupa transações de maneira eficaz e as aplica em lotes para fins de eficiência. A utilização da opção de aplicação otimizada em lote quase sempre viola as restrições de integridade referencial. Portanto, é recomendável desativar essas restrições durante o processo de migração e ativá-las novamente como parte do processo de substituição.

Utilização do seu próprio servidor de nomes on-premises

Normalmente, uma instância de replicação do AWS DMS utiliza o resolvedor do Sistema de Nomes de Domínio (DNS) em uma instância do Amazon EC2 para resolver os endpoints de domínio. No

entanto, é possível utilizar o seu próprio servidor de nomes on-premises para resolver determinados endpoints se você utilizar o Amazon Route 53 Resolver. Com essa ferramenta, é possível consultar entre on-premises e AWS utilizando endpoints de entrada e de saída, regras de encaminhamento e uma conexão privada. Os benefícios de utilizar um servidor de nomes on-premises incluem maior segurança e facilidade de uso por trás de um firewall.

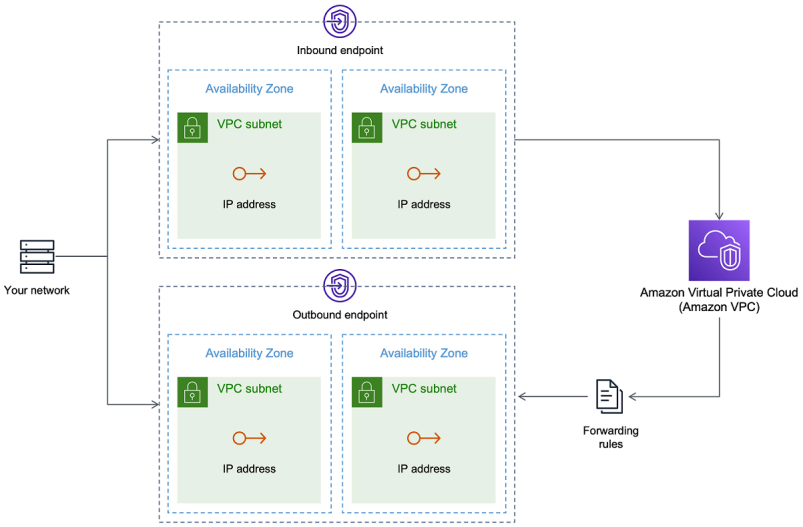
Com endpoints de entrada, é possível utilizar consultas ao DNS originadas on-premises para resolver domínios hospedados pela AWS. Para configurar os endpoints, atribua endereços IP em cada sub-rede para a qual você quer fornecer um resolvedor. Para estabelecer conectividade entre a infraestrutura do DNS on-premises e da AWS, utilize AWS Direct Connect ou uma rede privada virtual (VPN).

Os endpoints de saída se conectam ao servidor de nomes on-premises. O servidor de nomes só concede acesso aos endereços IP incluídos em uma lista de permissões e definidos em um endpoint de saída. O endereço IP do servidor de nomes é o endereço IP de destino. Ao escolher um grupo de segurança para um endpoint de saída, escolha o mesmo grupo de segurança usado pela instância de replicação.

Para encaminhar os domínios selecionados para o servidor de nomes, utilize as regras de encaminhamento. Um endpoint de saída pode processar várias regras de encaminhamento. O escopo da regra de encaminhamento é a nuvem privada virtual (VPC). Ao utilizar uma regra de encaminhamento associada a uma VPC, é possível provisionar uma seção logicamente isolada da nuvem AWS. Nessa seção logicamente isolada, é possível iniciar os recursos da AWS em uma rede virtual.

É possível configurar os domínios hospedados na infraestrutura do DNS on-premises como regras de encaminhamento condicional que configuram consultas ao DNS de saída. Quando uma consulta é feita a um desses domínios, as regras acionam uma tentativa de encaminhar solicitações de DNS para os servidores DNS que foram configurados com as regras. Novamente, é necessária uma conexão privada via AWS Direct Connect ou VPN.

O diagrama a seguir mostra a arquitetura do Route 53 Resolver.



Para obter mais informações sobre o Route 53 DNS Resolver, consulte Conceitos básicos do Route 53 Resolver no Guia do desenvolvedor do Amazon Route 53.

Utilização do Amazon Route 53 Resolver com o AWS DMS

É possível criar um servidor de nomes on-premises para que o AWS DMS resolva endpoints utilizando Amazon Route 53 Resolver.

Como criar um servidor de nomes on-premises para o AWS DMS com base no Route 53

1. Faça login no AWS Management Console e abra o console do Route 53 em https://console.aws.amazon.com/route53/.
2. No console do Route 53, escolha a região da AWS em que você deseja configurar o Route 53 Resolver. O Route 53 Resolver é específico para uma região.
3. Escolha a direção da consulta: entrada, saída ou ambas.
4. Forneça a configuração da consulta de entrada:
    a. Insira um nome de endpoint e escolha uma VPC.
    b. Atribua uma ou mais sub-redes na VPC (por exemplo, escolha duas para aumentar a disponibilidade).
    c. Atribua endereços IP específicos a serem usados como endpoints ou permita que o Route 53 Resolver os atribua automaticamente.
5. Crie uma regra para o domínio on-premises, para que as workloads na VPC possam rotear as consultas ao DNS para a sua infraestrutura de DNS.
6. Insira um ou mais endereços IP para os servidores DNS on-premises.

7. Envie sua regra.

Quando tudo estiver criado, a VPC estará associada às regras de entrada e de saída e poderá iniciar o roteamento do tráfego.

Para obter mais informações sobre o Route 53 Resolver, consulte Conceitos básicos do Route 53 Resolver no Guia do desenvolvedor do Amazon Route 53.

Migração de objetos binários grandes (LOBs)

Em geral, o AWS DMS migra dados de LOB em duas fases:

1. O AWS DMS cria uma nova linha na tabela de destino e preenche a linha com todos os dados, exceto o valor do LOB associado.
2. O AWS DMS atualiza a linha na tabela de destino com os dados de LOB.

Esse processo de migração de LOBs exige que, durante a migração, todas as colunas de LOB na tabela de destino sejam anuláveis. Isso acontece mesmo que as colunas de LOB não sejam anuláveis na tabela de origem. Se o AWS DMS criar as tabelas de destino, ele definirá as colunas de LOB como anuláveis por padrão. Em alguns casos, é possível criar as tabelas de destino utilizando algum outro mecanismo, como importação ou exportação. Nesses casos, verifique se as colunas de LOB são anuláveis antes de iniciar a tarefa de migração.

Esse requisito possui uma exceção. Suponha que você execute uma migração homogênea a partir de uma origem Oracle para um destino Oracle, e escolha Modo LOB limitado. Nesse caso, toda a linha é preenchida de uma só vez, incluindo os valores de LOB. Para esse caso, o AWS DMS pode criar as colunas de LOB da tabela de destino com restrições não anuláveis, se necessário.

Utilização do modo LOB limitado

O AWS DMS utiliza dois métodos que equilibram o desempenho e a conveniência quando a migração contém valores de LOB.

1. O Modo LOB limitado migra todos os valores de LOB até um limite de tamanho especificado pelo usuário (o padrão é 32 KB). Os valores de LOB maiores que o limite de tamanho devem ser migrados manualmente. O Modo LOB limitado, o padrão para todas as tarefas de migração, normalmente fornece o melhor desempenho. No entanto, verifique se o parâmetro Tamanho

máximo do LOB está correto. Defina esse parâmetro como o maior tamanho de LOB de todas as tabelas.

2. O Modo LOB completo migra todos os dados de LOB nas tabelas, independentemente do tamanho. O Modo LOB completo fornece a conveniência de mover todos os dados de LOB em suas tabelas, mas o processo pode ter um impacto significativo no desempenho.

Para alguns mecanismos de banco de dados, como o PostgreSQL, o AWS DMS trata tipos de dados JSON como LOBs. Verifique se você escolheu Modo LOB limitado, se a opção Tamanho máximo de LOB está definida como um valor que não trunca os dados JSON.

O AWS DMS oferece suporte total ao uso de tipos de dados de objetos grandes (BLOBs, CLOBs e NCLOBs). Os endpoints de origem a seguir têm suporte completo a LOB:

- Oracle
- Microsoft SQL Server
- ODBC

Os endpoints de destino a seguir têm suporte completo a LOB:

- Oracle
- Microsoft SQL Server

O endpoint de destino a seguir tem suporte limitado a LOB. Não é possível utilizar um tamanho ilimitado de LOB para esse endpoint de destino.

- Amazon Redshift
- Amazon S3

Para os endpoints que têm suporte completo a LOB, também é possível definir um limite de tamanho para tipos de dados de LOB.

Desempenho de LOB aprimorado

Ao migrar dados de LOB, é possível especificar as seguintes configurações diferentes de otimização de LOB.

Configurações de LOB por tabela

Utilizando as configurações de LOB por tabela, é possível substituir as configurações de LOB em nível de tarefa para algumas ou todas as tabelas. Para fazer isso, defina `lob-settings` na regra `table-settings`. Veja a seguir um exemplo de tabela que inclui alguns valores grandes de LOB.

```
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 || '0000000000000000000000000000000000';
END LOOP;
INSERT INTO TEST_CLOB (ID, C1, C2) VALUES (0, bigtextstring,'AnyValue');
END;
/
SELECT * FROM TEST_CLOB;
COMMIT
```

Crie uma tarefa de migração e modifique o tratamento de LOB da tabela utilizando a nova regra `lob-settings`. O valor de `bulk-max-siz` determina o tamanho máximo de LOB (KB). Ele será truncado se for maior que o tamanho especificado.

```
{
 "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"
   }
  }
 ]
}
```

Mesmo que essa tarefa do AWS DMS seja criada com `FullLobMode : true`, as configurações de LOB por tabela direcionam o AWS DMS a truncar os dados de LOB nessa tabela específica para 16.000. É possível verificar os logs de tarefas para confirmar isso.

```
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
```

Configurações de LOB em linha

Ao criar uma tarefa do AWS DMS, o modo LOB determina como os LOBs são processados.

Com o modo LOB completo e o modo LOB limitado. Cada um tem suas próprias vantagens e desvantagens. O modo LOB em linha combina as vantagens do modo LOB completo e do modo LOB limitado.

É possível utilizar o modo LOB em linha quando for necessário replicar LOBs pequenos e grandes, e a maioria dos LOBs forem pequenos. Ao escolher essa opção, durante a carga máxima, a tarefa do AWS DMS transfere os pequenos LOBs em linha, o que é mais eficiente. A tarefa do AWS DMS transfere os LOBs grandes executando uma pesquisa na tabela de origem.

Durante o processamento de alterações, LOBs pequenos e grandes são replicados executando uma pesquisa na tabela de origem.

Ao utilizar o modo LOB em linha, a tarefa do AWS DMS verifica todos os tamanhos de LOB para determinar quais devem ser transferidos em linha. LOBs maiores que o tamanho especificado são replicados utilizando o modo LOB completo. Portanto, se você sabe que a maioria dos LOBs é

maior que a configuração especificada, é melhor não utilizar essa opção. Em vez disso, permita um tamanho de LOB ilimitado.

Configure essa opção utilizando um atributo nas configurações da tarefa`InlineLobMaxSize`, que só está disponível quando `FullLobMode` está definido como `true`. O valor padrão de `InlineLobMaxSize` é 0, e o intervalo é de 1 a 102400 kilobytes (100 MB).

Por exemplo, é possível utilizar as seguintes configurações da tarefa AWS DMS. Aqui, a configuração de `InlineLobMaxSize` como um valor de 5 resulta em todos os LOBs menores que ou iguais a 5 KiB (5.120 bytes) sendo transferidos em linha.

```
{
  "TargetMetadata": {
  "TargetSchema": "",
  "SupportLobs": true,
  "FullLobMode": true,
  "LobChunkSize": 64,
  "LimitedSizeLobMode": false,
  "LobMaxSize": 32,
  "InlineLobMaxSize": 5,
  "LoadMaxFileSize": 0,
  "ParallelLoadThreads": 0,
  "ParallelLoadBufferSize":0,
  "BatchApplyEnabled": false,
  "TaskRecoveryTableEnabled": false},
  . . .
}
```

Melhoria do desempenho ao migrar tabelas grandes utilizando filtragem de linhas

Para melhorar o desempenho ao migrar uma tabela grande, é possível dividir a migração em mais de uma tarefa. Para dividir a migração em várias tarefas utilizando a filtragem de linhas, utilize uma chave ou uma chave de partição. Por exemplo, se tiver um ID de chave primária do tipo inteiro de 1 a 8.000.000, é possível criar oito tarefas utilizando a filtragem de linha para migrar um milhão de registros em cada.

Para aplicar a filtragem de linhas no console:

1. Abra a AWS Management Console.

2. Escolha Tarefas e crie uma nova tarefa.
3. Escolha a guia Mapeamentos de tabela e expanda a guia Regras de seleção.
4. Escolha Adicionar nova regra de seleção. Agora é possível adicionar um filtro de colunas com menor que ou igual a, maior que ou igual a, igual a ou uma condição de intervalo entre dois valores. Para obter mais informações sobre a filtragem de colunas, consulte Especificar a seleção de tabelas e as regras de transformação no console.

Se você tiver um grande tabela particionada por data, poderá migrar os dados com base em data. Por exemplo, suponha que você tenha uma tabela particionada por mês e somente os dados do mês atual estão atualizados. Nesse caso, é possível criar uma tarefa de carga máxima para cada partição mensal estática e criar uma tarefa de carga máxima CDC para a partição atualizada atualmente.

Se a tabela tiver uma chave primária de coluna única ou um índice exclusivo, você poderá fazer com que a tarefa do AWS DMS segmente a tabela utilizando uma carga paralela do tipo de intervalos para carregar os dados em paralelo. Para ter mais informações, consulte Regras e operações de configurações de tabelas e coleções.

Replicação contínua

O AWS DMS fornece replicação contínua de dados, mantendo os bancos de dados de origem e de destino em sincronia. Ele replica apenas uma quantidade limitada de instruções da linguagem de definição de dados (DDL). O AWS DMS não propaga itens como índices, usuários, privilégios, procedimentos armazenados e outras alterações de banco de dados não relacionadas diretamente a dados de tabela.

Ao planejar utilizar a replicação contínua, defina a opção Multi-AZ ao criar a instância de replicação. Ao escolher a opção Multi-AZ, você obtém alta disponibilidade e suporte a failover para a instância de replicação. No entanto, essa opção pode ter um impacto no desempenho e retardar a replicação ao aplicar alterações no sistema de destino.

Antes de atualizar os bancos de dados de origem ou de destino, é recomendável interromper todas as tarefas do AWS DMS que estão sendo executadas nesses bancos de dados. Retome as tarefas após concluir as atualizações.

Durante a replicação contínua, é essencial identificar a largura de banda da rede entre o sistema de banco de dados de origem e a instância de replicação do AWS DMS. Verifique se a rede não causa gargalos durante a replicação contínua.

Também é importante identificar a taxa de alterações e a geração do log de arquivamento por hora no sistema de banco de dados de origem. Isso pode ajudar a compreender o throughput que pode ser obtido durante a replicação contínua.

Redução da carga no banco de dados de origem

O AWS DMS utiliza alguns recursos no banco de dados de origem. Durante uma tarefa de carga máxima, o AWS DMS executa uma varredura total da tabela de origem de cada tabela processada em paralelo. Além disso, cada tarefa criada como parte de uma migração consulta a origem em busca de alterações como parte do processo de CDC. Para que o AWS DMS execute a CDC para algumas origens, como o Oracle, poderá ser necessário aumentar a quantidade de dados gravados no log de alterações do bancos de dados.

Se você descobrir que está sobrecarregando o banco de dados de origem, será possível reduzir o número de tarefas ou de tabelas de cada tarefa da migração. Cada tarefa obtém as alterações de origem de forma independente, portanto a consolidação das tarefas pode diminuir a workload da captura de alterações.

Reduzir os gargalos no banco de dados de destino

Durante a migração, tente remover todos os processos que competem por recursos de gravação no banco de dados de destino:

- Desative os acionadores desnecessários.
- Desative os índices secundários durante a carga inicial e ative-os posteriormente durante a replicação contínua.
- Com os bancos de dados do Amazon RDS, é uma boa ideia desativar os backups e o multi-AZ até a substituição.
- Ao migrar para sistemas não RDS, é uma boa ideia desativar qualquer log no destino até a substituição.

Utilização da validação de dados durante a migração

Para garantir que os dados foram migrados com precisão da origem para o destino, é altamente recomendável utilizar a validação de dados. Se você ativar a validação de dados de uma tarefa, o AWS DMS começará a comparar os dados de origem e de destino imediatamente após a execução da carga máxima de uma tabela.

A validação de dados funciona com os seguintes bancos de dados sempre que o AWS DMS é compatível com eles como endpoints de origem e de destino:

- Oracle
- PostgreSQL
- MySQL
- MariaDB
- Microsoft SQL Server
- Amazon Aurora Edição Compatível com MySQL
- Amazon Aurora Edição Compatível com PostgreSQL
- IBM Db2 LUW
- Amazon Redshift

Para ter mais informações, consulte Validação de dados do AWS DMS.

Monitoramento das tarefas do AWS DMS utilizando métricas

Há várias opções para monitorar as métricas das tarefas utilizando o console do AWS DMS:

Métricas de host

Você pode encontrar métricas do host na guia de CloudWatch métricas de cada instância de replicação específica. Aqui, é possível monitorar se a instância de replicação está dimensionada de forma adequada.

Métricas de tarefas de replicação

Métricas para tarefas de replicação, incluindo alterações recebidas e confirmadas, e latência entre o host de replicação e os bancos de dados de origem/destino, podem ser encontradas na guia de métricas de cada tarefa específica. CloudWatch

Métricas de tabela

É possível descobrir as métricas de tabela individuais na guia Estatísticas da tabela de cada tarefa individual. Essas métricas incluem os números de:

- Linhas carregadas durante a carga máxima.

- Inserções, atualizações e exclusões desde o início da tarefa.
- Operações de DDL desde o início da tarefa.

Para obter mais informações sobre o monitoramento de métricas, consulte Monitoramento de tarefas do AWS DMS.

Eventos e notificações

O AWS DMS utiliza o Amazon SNS para fornecer notificações sobre um evento ocorrido no AWS DMS, por exemplo, a criação ou a exclusão de uma instância de replicação. É possível trabalhar com essas notificações em qualquer formato compatível com o Amazon SNS de uma região da AWS. Isso pode incluir mensagens de e-mail, mensagens de texto ou chamadas para um endpoint HTTP.

Para ter mais informações, consulte Como trabalhar com eventos e notificações do Amazon SNS no AWS Database Migration Service.

Utilização do log de tarefas para solucionar problemas de migração

Em alguns casos, o AWS DMS pode encontrar problemas para os quais os avisos ou as mensagens de erro aparecem somente no log de tarefas. Especificamente, os problemas de truncamento de dados ou de rejeições de linhas devido a violações de chave estrangeira só são gravados no log de tarefas. Portanto, analise o log de tarefas ao migrar um banco de dados. Para visualizar o registro de tarefas, configure a Amazon CloudWatch como parte da criação da tarefa.

Para obter mais informações, consulte Monitoramento de tarefas de replicação usando a Amazon CloudWatch.

Solução de problemas de tarefas de replicação com o Time Travel

Para solucionar problemas de migração do AWS DMS, é possível trabalhar com o Time Travel. Para obter mais informações sobre o Time Travel, consulte Configurações de tarefa do Time Travel.

Ao trabalhar com o Time Travel, lembre-se das seguintes considerações:

- Para evitar sobrecarga em uma instância de replicação do DMS, ative o Time Travel somente para tarefas que precisam de depuração.
- Ao utilizar o Time Travel para solucionar problemas de tarefas de replicação que podem ser executadas por vários dias, monitore as métricas da instância de replicação quanto à sobrecarga

de recursos. Essa abordagem se aplica especialmente aos casos em que altas cargas de transações são executadas nos bancos de dados de origem por longos períodos. Para obter mais detalhes, consulte Monitoramento de tarefas do AWS DMS.

- Quando a configuração `EnableRawData` da tarefa do Time Travel está definida como`true`, o uso da memória da tarefa durante a replicação do DMS pode ser maior do que quando o Time Travel não está ativado. Se você ativar a Viagem no Tempo por longos períodos, monitore a tarefa.
- Atualmente, é possível ativar o Time Travel somente no nível de tarefa. As alterações em todas as tabelas são registradas nos logs do Time Travel. Se estiver solucionando problemas de tabelas específicas em um banco de dados com alto volume de transações, crie uma tarefa separada.

Alteração de usuário e de esquema de um destino do Oracle

Ao utilizar o Oracle como destino, o AWS DMS migra os dados para o esquema de propriedade do usuário do endpoint de destino.

Por exemplo, suponha que você esteja migrando um esquema chamado `PERFDATA` para um endpoint de destino do Oracle, e que o nome do usuário do endpoint de destino seja `MASTER`. O AWS DMS se conectará ao destino do Oracle como `MASTER`, e preencherá o esquema `MASTER` com objetos do banco de dados `PERFDATA`.

Para substituir esse comportamento, forneça uma transformação de esquema. Por exemplo, para migrar os objetos do esquema `PERFDATA` para um esquema `PERFDATA` no endpoint de destino, utilize a seguinte transformação:

```
{
    "rule-type": "transformation",
    "rule-id": "2",
    "rule-name": "2",
    "object-locator": {
        "schema-name": "PERFDATA"
    },
    "rule-target": "schema",
    "rule-action": "rename",
    "value": "PERFDATA"
}
```

Para obter mais informações sobre transformações, consulte Especificar a seleção de tabelas e as regras de transformação utilizando JSON.

Alteração de espaços para tabela de índice e de tabela para um destino do Oracle

Quando o Oracle é usado como destino, o AWS DMS migra todas as tabelas e índices para o espaço para tabela padrão no destino. Por exemplo, suponha que a origem seja um mecanismo de banco de dados diferente do Oracle. Todas as tabelas e índices de destino são migrados para o mesmo espaço para tabela padrão.

Para substituir esse comportamento, forneça transformações de espaço para tabela correspondentes. Por exemplo, suponha que você queira migrar tabelas e índices para espaços para tabela de índices e de tabelas no destino do Oracle que têm o mesmo nome do esquema na origem. Nesse caso, é possível utilizar transformações semelhantes às seguintes. Aqui, o esquema na origem é chamado `INVENTORY`, e os espaços para tabela de índices e de tabelas correspondentes no destino são chamados `INVENTORYTBL` e `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"
```

```
}
```

Para obter mais informações sobre transformações, consulte Especificar a seleção de tabelas e as regras de transformação utilizando JSON.

Quando o Oracle é a origem e o destino, é possível preservar as atribuições de espaço para tabela de índice ou de tabela existente definindo o atributo `enableHomogenousTablespace=true` de conexão extra da origem do Oracle. Para ter mais informações, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.

Atualização de uma versão de instância de replicação

A AWS lança periodicamente novas versões do software do mecanismo de replicação do AWS DMS, com novos recursos e melhorias de desempenho. Cada versão do software do mecanismo de replicação tem seu próprio número de versão. É essencial testar a versão existente da instância de replicação do AWS DMS executando uma workload de produção antes de atualizar a instância de replicação para uma versão posterior. Para obter mais informações sobre upgrades de versões disponíveis, consulte AWS Notas de versão do DMS.

Compreender o custo da migração

O AWS Database Migration Service ajuda a migrar bancos de dados para a AWS de forma fácil e econômica. Você paga somente pelas instâncias de replicação e por qualquer armazenamento de log adicional. Cada instância de migração de banco de dados inclui armazenamento suficiente para espaço de troca, logs de replicação e cache de dados para a maioria das replicações, e a transferência de dados de entrada é gratuita.

Talvez sejam necessários mais recursos durante a carga inicial ou durante o horário de pico de carga. É possível monitorar estreitamente a utilização dos recursos da instância de replicação utilizando as métricas do Cloud Watch. É possível aumentar e reduzir a escala verticalmente do tamanho da instância de replicação com base no uso.

Para obter mais informações sobre como estimar os custos da migração, consulte:

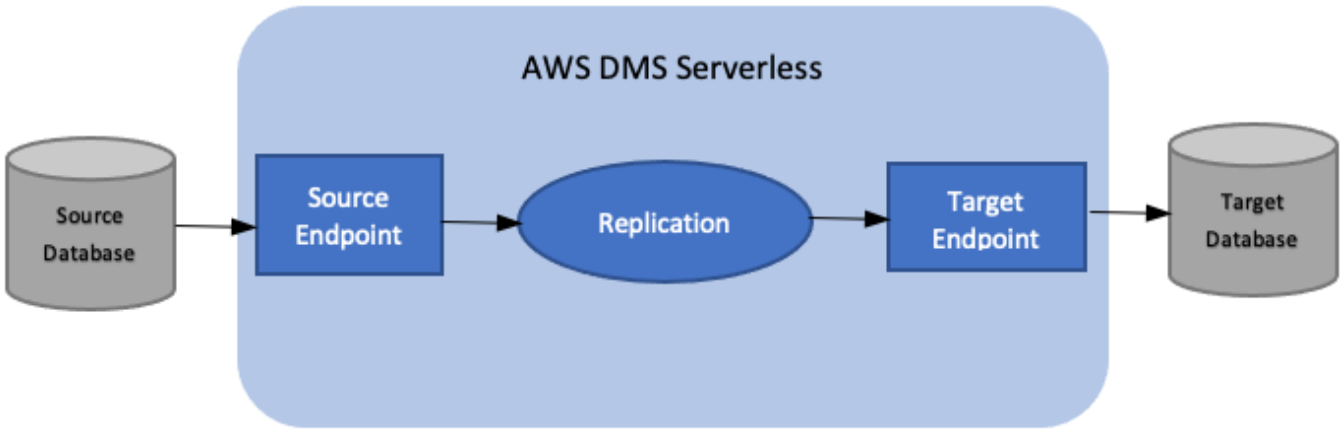
- Preços do AWS Database Migration Service
- Calculadora de preços da AWS

Trabalhando com AWS DMS Serverless

AWS DMS O Serverless é um recurso que fornece provisionamento automático, escalabilidade, alta disponibilidade integrada e um modelo de pay-for-use cobrança para aumentar a agilidade das operações e otimizar seus custos. O recurso que utiliza tecnologia sem servidor elimina as tarefas de gerenciamento de instâncias de replicação, como a estimativa de capacidade, o provisionamento, a otimização de custos e o gerenciamento de versões e de patches do mecanismo de replicação.

Com o AWS DMS Serverless, semelhante à funcionalidade atual do AWS DMS (referida neste documento como AWS DMS Padrão), você cria conexões de origem e destino usando endpoints. Depois de criar os endpoints de origem e de destino, crie uma configuração de replicação, que inclua as definições de configuração da replicação em questão. É possível gerenciar as replicações iniciando, interrompendo, modificando ou excluindo-as. Cada replicação tem configurações que podem ser definidas de acordo com os requisitos da migração do banco de dados. Você especifica essas configurações usando um arquivo JSON ou a AWS DMS seção do AWS Management Console. Para obter mais informações sobre as configurações de replicação, consulte Trabalhando com AWS DMS endpoints. Depois de iniciar a replicação, o AWS DMS com Tecnologia Sem Servidor se conecta ao banco de dados de origem e coleta os metadados do banco de dados para analisar a workload da replicação. Usando esses metadados, AWS DMS calcula e provisiona a capacidade necessária e inicia a replicação dos dados.

O diagrama a seguir mostra o processo de replicação AWS DMS sem servidor.



> **Note**
>
> AWS DMS O Serverless usa a versão padrão do mecanismo. Para obter informações sobre as versões padrão do mecanismo, consulte Notas de release.

Veja os tópicos a seguir para descobrir mais detalhes sobre o AWS DMS Serverless.

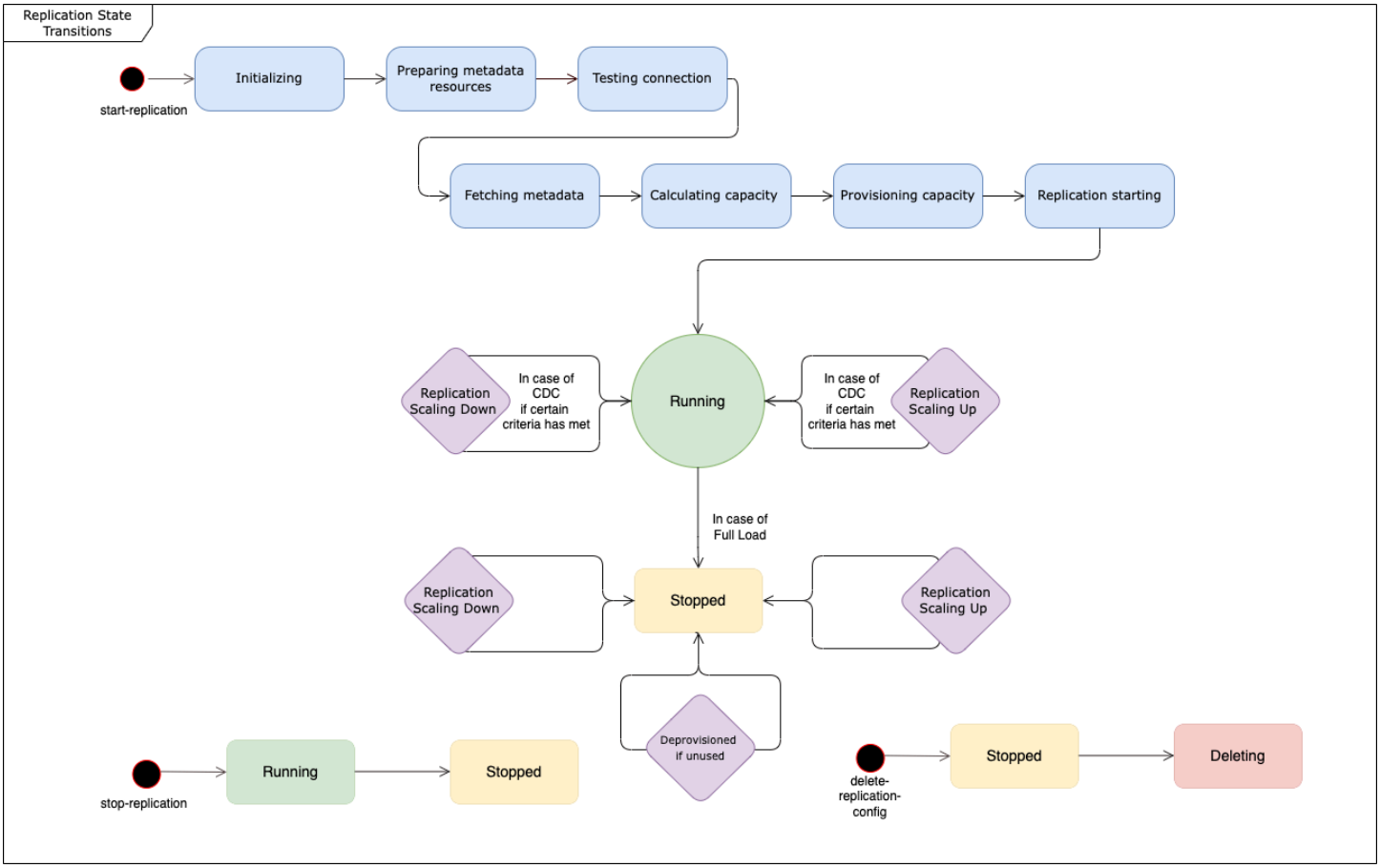
Tópicos

- AWS DMS Componentes sem servidor
- AWS DMS Limitações sem servidor

AWS DMS Componentes sem servidor

Para gerenciar os recursos necessários para realizar uma replicação, o AWS DMS Serverless tem estados granulares que revelam diferentes ações internas tomadas pelo serviço. Ao iniciar a replicação, o AWS DMS com Tecnologia Sem Servidor calcula a carga da capacidade, provisiona a capacidade calculada e inicia a replicação dos dados de acordo com os seguintes estados de replicação.

O diagrama a seguir mostra as transições de estado para uma replicação AWS DMS sem servidor.



- O primeiro estado depois de iniciar a replicação é Inicializando. Nesse estado, todos os parâmetros necessários são inicializados.
- Os estados imediatamente seguintes incluem Preparação de recursos de metadados, Teste de conexão e Busca de metadados. Nesses estados, o AWS DMS Serverless se conecta ao seu banco de dados de origem para obter as informações necessárias para prever a capacidade necessária.
  - Quando o estado de replicação é Testar conexão, o AWS DMS Serverless verifica se a conexão com seus bancos de dados de origem e destino foi configurada com êxito.
  - O estado de replicação depois de Testando a conexão é Buscando metadados. Aqui, AWS DMS recupera as informações necessárias para calcular a capacidade.
  - Depois de AWS DMS recuperar as informações necessárias, o próximo estado é Calcular a capacidade. Aqui, o sistema calcula o tamanho dos recursos subjacentes necessários para executar a replicação.
- A transição de estado após Calculando a capacidade é Provisionando a capacidade. Enquanto a replicação está nesse estado, o AWS DMS com Tecnologia Sem Servidor inicializa os recursos de computação subjacentes.

- O estado de replicação após o provisionamento bem-sucedido de todos os recursos é Iniciando a replicação. Nesse estado, o AWS DMS Serverless inicia a replicação dos dados. As fases de uma replicação incluem o seguinte:
  - Carga total: nessa fase, o DMS replica o armazenamento de dados de origem como estava quando a replicação foi iniciada.
  - CDC (inicial): Nessa fase, o DMS replica as alterações no armazenamento de dados de origem que ocorreram durante a fase de carregamento total. O DMS só executa essa fase se a configuração da `StopTaskCachedChangesNotApplied` tarefa for`false`.
  - CDC (em andamento): após a fase inicial do CDC, o DMS replica as alterações no banco de dados de origem à medida que elas ocorrem. O DMS só continua executando a replicação após a fase inicial do CDC se a configuração da `StopTaskCachedChangesApplied` tarefa for. `false`
- O estado final é Em execução. No estado Em execução, a replicação dos dados está em andamento.
- Uma replicação que você interrompe entra no estado Parado. Você pode reiniciar uma replicação interrompida nas seguintes circunstâncias:
  - Você não pode reiniciar uma replicação que o DMS tenha desprovisionado.
  - Você pode reiniciar uma replicação interrompida somente de CDC ou de carga total e de CDC usando a ação. StartReplication Você não pode reiniciar uma replicação interrompida usando o console.
  - Você não pode reiniciar uma replicação interrompida que usa o PostgreSQL como mecanismo.

Este tópico contém as seguintes seções:

Para AWS DMS Serverless, o painel de navegação esquerdo do AWS DMS console tem uma nova opção, replicações sem servidor. Para Replicações que usam tecnologia sem servidor, especifique Replicações em vez de tipos de instância de replicação ou tarefas para definir uma replicação. Além disso, especifique as unidades de capacidade (DCUs) máxima e mínima do DMS que você deseja que o DMS provisione para a replicação. Uma DCU tem 2 GB de RAM. AWS DMS cobra de sua conta cada DCU que sua replicação está usando atualmente. Para obter informações sobre AWS DMS preços, consulte Preços do AWS Database Migration Service.

AWS DMS em seguida, provisiona automaticamente os recursos de replicação com base em seus mapeamentos de tabelas e no tamanho previsto de sua carga de trabalho. Essa unidade de capacidade é um valor na faixa dos valores de unidades de capacidade mínima e máxima que você especifica.

Versões compatíveis do mecanismo

Com o AWS DMS Serverless, você não precisa escolher e gerenciar as versões do mecanismo, pois o serviço lida com essa configuração. AWS DMS O Serverless oferece suporte às seguintes fontes:

- Microsoft SQL Server
- Bancos de dados compatíveis com o PostgreSQL
- Bancos de dados compatíveis com o MySQL
- MariaDB
- Oracle
- IBM Db2

AWS DMS O Serverless oferece suporte aos seguintes destinos:

- Microsoft SQL Server
- PostgreSQL
- Bancos de dados compatíveis com o MySQL
- Oracle
- Amazon S3
- Amazon Redshift
- Amazon DynamoDB
- Amazon Kinesis Data Streams

- Amazon Managed Streaming for Apache Kafka
- OpenSearch Serviço Amazon
- Amazon DocumentDB (compatível com MongoDB)
- Amazon Neptune

Como parte do AWS DMS Serverless, você tem acesso aos comandos do console que permitem criar, configurar, iniciar e gerenciar replicações sem AWS DMS servidor. Para executar esses comandos utilizando a seção Replicações que usam tecnologia sem servidor do console, faça o seguinte:

- Configure uma nova política AWS Identity and Access Management (IAM) e uma função do IAM para anexar essa política.
- Use um AWS CloudFormation modelo para fornecer o acesso de que você precisa.

AWS DMS O Serverless exige que uma função vinculada ao serviço (SLR) exista em sua conta. AWS DMS gerencia a criação e o uso dessa função. Para obter mais informações sobre como garantir que você tem a SLR necessária, consulte Perfil vinculado a serviço do AWS DMS com Tecnologia Sem Servidor.

Criar uma replicação que utiliza tecnologia sem servidor

Para criar uma replicação sem servidor entre dois AWS DMS endpoints existentes, faça o seguinte. Para obter informações sobre a criação de AWS DMS endpoints, consulteCriar endpoints de origem e de destino.

Criar uma replicação que utiliza tecnologia sem servidor

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, escolha Replicações que usam tecnologia sem servidor e Criar replicação.
3. Na página Criar replicação, especifique a configuração da replicação que utiliza tecnologia sem servidor:

Opção	Ação
Nome	Insira um nome para identificar a replicação, como **`DMS-replication`** .
Nome do recurso da Amazon (ARN) descritivo: opcional	Você pode usar esse parâmetro opcional para fornecer uma descrição da replicação.
Endpoint do banco de dados de origem	Escolha endpoints existentes na sua conta. Observe que o AWS DMS Serverless oferece suporte apenas a um subconjunto dos tipos de endpoints compatíveis com o padrão. AWS DMS
Endpoint do banco de dados de destino	Escolha endpoints existentes na sua conta. Observe que o AWS DMS Serverless oferece suporte apenas a um subconjunto dos tipos de endpoints compatíveis com o padrão. AWS DMS
Tipo de replicação	Escolha um tipo de replicação com base em seus requisitos: • Carga total: AWS DMS migra somente os dados existentes. • Carga total e captura de dados de alteração (CDC): AWS DMS migra os dados existentes e as alteraçõe s que ocorrem durante a replicação. • Captura de dados de alterações (CDC): migra AWS DMS somente as alterações que ocorrem após o início da replicação.

Na seção Configurações, defina as configurações necessárias para a replicação.

Na seção Mapeamentos de tabelas, configure o mapeamento de tabela para definir as regras para selecionar e filtrar os dados que você está replicando. Antes de especificar o mapeamento, analise a seção da documentação sobre o mapeamento de tipo de dados do banco de dados de origem e de destino. Para obter informações sobre o mapeamento de tipos de dados para seus

bancos de dados de origem e destino, consulte a seção de tipos de dados para seus tipos de endpoint de origem e destino no Como trabalhar com endpoints do AWS DMS tópico.

Na seção Configurações de computação, defina as seguintes configurações. Para obter mais informações sobre como definir essas configurações, consulte Configuração da computação.

Opção	Ação
VPC	Escolha uma VPC existente.
Grupo de sub-redes	Escolha um grupo de sub-redes existente.
Grupo(s) de segurança da VPC	Escolha padrão se ainda não estiver escolhido.
AWS Chave KMS	Escolha uma chave KMS apropriada. Para obter informações sobre chaves KMS, consulte Criação de chaves na Referência da AWS Key Management Service API.
Implantação	Deixe como está.
Zona de disponibilidade	Deixe como está.
Unidades de capacidade do DMS (DCU) mínima - (opcional)	Deixe em branco para utilizar o valor padrão de 1 DCU.
Unidades de capacidade máxima do DMS (DCU)	Escolha 16 DCU.

Deixe as configurações de Manutenção como estão.

4. Escolha Criar replicação.

AWS DMS cria uma replicação sem servidor para realizar sua migração.

Modificando replicações AWS DMS sem servidor

Para modificar a configuração da replicação, utilize a ação `modify-replication-config`. Você só pode modificar uma configuração AWS DMS de replicação que esteja nos `FAILED` estados

CREATEDSTOPPED, ou. Para obter informações sobre a modify-replication-config ação, consulte ModifyReplicationConfig na Referência da AWS Database Migration Service API.

Para modificar uma configuração de replicação sem servidor usando o AWS Management Console

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, escolha Replicações que usam tecnologia sem servidor.
3. Escolha a instância de replicação que deseja modificar. A tabela a seguir descreve as modificações que podem ser feitas com base no estado atual da replicação.

Configuração	Descrição	Estados permitidos
Nome	É possível alterar o nome da instância da replicação. Insira um nome para a instância de replicação que contenha de 8 a 16 caracteres ASCII imprimíveis (excluindo /, " e @). O nome deve ser exclusivo para a sua conta na região da AWS selecionada. Você pode optar por adicionar alguns detalhes ao nome, como incluir a AWS região e a tarefa que você está executando, por exemplo:**west2-mysql2mysql-config1** .	ReplicationState é CREATED, STOPPED ou FAILED.
Endpoint do banco de dados de origem	Escolha um novo endpoint de origem existente como a origem da replicação.	ReplicationState é CREATED ou FAILED quando Provision State for null.
Endpoint do banco de dados de destino	Escolha um novo endpoint de destino existente como o destino da replicação.	ReplicationState é CREATED ou FAILED quando Provision State for null.

Configuração	Descrição	Estados permitidos
Tipo de replicação	É possível modificar o tipo de uma replicação que utiliza tecnologia sem servidor.	`ReplicationState` é `CREATED` ou `FAILED` quando `Provision State` for `null`.
Configurações da replicação	É possível modificar as configurações da replicação, incluindo o modo de preparação da tabela de destino, se incluir colunas de LOB na replicação, tamanho máximo de LOB, validação e registro em log. Para ter mais informações, consulte Configurações de tarefa.	`ReplicationState` é `CREATED`, `STOPPED` ou `FAILED`.
Mapeamentos de tabelas	É possível modificar as configurações de mapeamentos de tabelas para uma replicação que utiliza tecnologia sem servidor, incluindo as regras de seleção e as regras de transformação. Para ter mais informações, consulte Mapeamento de tabela.	`ReplicationState` é `CREATED`, `STOPPED` ou `FAILED`.

Configuração	Descrição	Estados permitidos
Configuração da computação	É possível modificar as configurações da computação para uma replicação sem servidor, incluindo as configurações de rede, de escalabilidade e de manutenção. Para obter mais informações sobre como definir essas configurações, consulte Configuração da computação.	• É possível modificar as seguintes configurações de ajuste da escala, de manutenção e de rede quando `ReplicationState` for `CREATED`, `STOPPED` ou `FAILED`: • `MinCapacityUnits` • `MaxCapacityUnits` • `MultiAZ` • `PreferredMaintenanceWindow` • `VpcSecurityGroupIds` • É possível modificar as seguintes configurações de rede e de segurança quando `ReplicationState` for `CREATED` ou `FAILED` quando `ProvisionState` for `null`: • `AvailabilityZone` • `DnsNameServers` • `KmsKeyId`

Configuração	Descrição	Estados permitidos
		• `ReplicationSubnetGroupId`

Configuração da computação

Você configura o provisionamento da replicação utilizando o parâmetro Compute Config ou a seção do console. Os campos no objeto Compute Config incluem o seguinte:

Opção	Descrição
MinCapacityUnidades	Esse é o número mínimo de unidades de capacidade (DCU) do DMS que AWS DMS serão provisionadas. Essa também é a DCU mínima para a qual o ajuste de escala automático pode reduzir a escala verticalmente.
MaxCapacityUnidades	Essa é a quantidade máxima de unidades de capacidade do DMS (DCU) que o AWS DMS pode provisionar, dependendo da previsão de capacidade da replicação. Essa também é a DCU máxima para a qual o ajuste de escala automático pode aumentar a escala verticalmente.
KmsKeyIdentificação	Escolha a chave de criptografia a ser utilizada para criptografar o armazenamento da replicação e as informações de conexão. Se você escolher (Padrão) aws/dms, AWS DMS usa a chave KMS padrão associada à sua conta e. Região da AWS Uma descrição e o número da sua conta aparecem juntamente com o ARN da chave. Para obter mais informações sobre a utilizaçã o da chave de criptografia, consulte Configurando uma chave de criptografia e especificando permissões AWS KMS. Neste tutorial, deixe (Padrão) aws/dms escolhido.
ReplicationSubnetGroupId	Escolha o grupo de sub-rede de replicação na VPC selecionada em que deseja que instância de replicação

Opção	Descrição
	seja criada. Se o banco de dados de origem estiver em uma VPC, escolha o grupo de sub-redes que contém o banco de dados de origem como o local da instância de replicação. Para obter mais informações sobre grupos de sub-rede de replicação, consulte Criar um grupo de sub-rede de replicação.
VpcSecurityGroupIds	A instância de replicação é criada em um VPC. Se o banco de dados de origem estiver em uma VPC, selecione o grupo de segurança da VPC que fornece acesso à instância de banco de dados em que o banco de dados está localizado.
PreferredMaintenanceJanela	Esse parâmetro define um período semanal durante o qual a manutenção do sistema pode ocorrer, em Universal Coordinated Time (UTC). O padrão é uma janela de 30 minutos selecionada aleatoriamente a partir de um bloco de 8 horas por vez Região da AWS, ocorrendo em um dia aleatório da semana.
MultiAZ	A configuração desse parâmetro opcional cria uma réplica em espera da replicação em outra zona de disponibilidade para suporte a failover. Se você pretender utilizar a captura de dados de alteração (CDC) ou a replicação contínua, ative essa opção.

Entendendo o escalonamento automático sem servidor AWS DMS

Depois de provisionar uma replicação e ela estar no `RUNNING` estado, o AWS DMS serviço gerencia a capacidade dos recursos subjacentes de se adaptarem às mudanças nas cargas de trabalho. Esse gerenciamento escala os recursos da replicação com base nas seguintes configurações da replicação:

- `MinCapacityUnits`
- `MaxCapacityUnits`

As replicações aumentam a escala verticalmente após um período de utilização excessiva e reduzem a escala verticalmente quando a utilização da capacidade fica abaixo do limite mínimo de utilização da capacidade por um período mais longo.

> **Note**
>
> As replicações sem servidor não podem ser reduzidas automaticamente enquanto uma carga completa está em andamento.

Ajustando o escalonamento automático sem servidor AWS DMS

Para ajustar seus parâmetros de escalonamento automático de replicação, recomendamos que você `MaxCapacityUnits` defina o valor máximo e deixe AWS DMS gerenciar o provisionamento de recursos. É recomendável escolher a maior configuração da capacidade máxima de DCU para permitir maior benefício no ajuste de escala automático, a fim de acomodar picos no volume de transações. A calculadora de preços mostra o custo mensal máximo se a replicação utilizar continuamente a DCU máxima. A DCU máxima não representa o custo real, pois você paga apenas pela capacidade utilizada.

Se sua replicação não estiver usando seus recursos em sua capacidade total, AWS DMS desprovisionará gradualmente os recursos para economizar seus custos. No entanto, como o provisionamento e o desprovisionamento de recursos demoram, é recomendável definir a configuração `MinCapacityUnits` como um valor que possa tratar qualquer pico repentino esperado na workload de replicação. Isso evitará que sua replicação seja subprovisionada e, ao mesmo tempo, provisionará recursos para o nível mais alto da AWS DMS carga de trabalho.

Se você subprovisionar a replicação com uma configuração de capacidade máxima muito baixa para os requisitos dos dados ou uma capacidade mínima muito baixa para lidar com picos repentinos na workload de replicação, é provável que a métrica `CapacityUtilization` se evidencie consistentemente em seu valor máximo. Isso pode fazer com que a replicação falhe. Se sua replicação falhar devido a recursos subprovisionados, AWS DMS cria um out-of-memory evento em seus registros de replicação. Se a out-of-memory condição ocorreu devido a um aumento repentino na carga de trabalho de replicação, a replicação será escalonada e reiniciada automaticamente.

Monitorando AWS DMS replicações sem servidor

AWS fornece várias ferramentas para monitorar suas replicações AWS DMS sem servidor e responder a possíveis incidentes:

- AWS DMS métricas de replicação sem servidor
- AWS DMS registros de replicação sem servidor

AWS DMS métricas de replicação sem servidor

O monitoramento da replicação sem servidor inclui CloudWatch métricas da Amazon para as seguintes estatísticas. Essas estatísticas são agrupadas por cada replicação que utiliza tecnologia sem servidor.

Métrica	Unidades	Descrição
CapacityUtilization	Percentual	A porcentagem de memória utilizada pela replicação que utiliza tecnologia sem servidor
CDC IncomingChanges	Percentual	O número total de eventos de alteração em um point-in-time que estão aguardando para serem aplicados ao alvo. Observe que isso não é o mesmo que uma medida da taxa de alteração de transação do endpoint de origem. Um grande número dessa métrica geralmente indica que não AWS DMS é possível aplicar as alterações capturadas em tempo hábil, causando alta latência alvo.
CDC LatencySource	Segundos	O intervalo, em segundos, entre o último evento capturado no endpoint de origem e o timestamp atual do sistema da instância do AWS DMS . O CDC LatencySource representa a latência entre a origem e a instância de replicação. Um CDC alto LatencySource significa que o processo de captura de alterações da fonte está atrasado. Para identificar a latência em uma replicação contínua, você pode visualizar essa métrica junto com o CDC. LatencyTarget Se o CDC LatencySource e o CDC LatencyTarget estiverem altos, investigue primeiro o CDCLatencySource. O CDC LatencySource pode ser 0 quando não há atraso de replicação entre a origem e a replicação.

Métrica	Unidades	Descrição
		O CDC também LatencySource pode se tornar zero quando a replicação tenta ler o próximo evento no log de transações da origem e não há novos eventos em comparação com a última vez em que foi lida da fonte. Quando isso acontece, a replicação redefine o CDC LatencySource para 0.
CDC LatencyTarget	Segundos	O intervalo, em segundos, entre o primeiro timestamp de evento em espera de confirmação no destino e o timestamp atual da instância do AWS DMS . A latência do destino é a diferença entre a hora do servidor da instância de replicação e o ID do evento não confirmado mais antigo encaminhado para um componente de destino. Em outras palavras, a latência de destino é a diferença do timestamp entre a instância de replicação e o evento mais antigo aplicado, mas não confirmado pelo endpoint de TRG (99%). Quando o CDC LatencyTarget está alto, isso indica que o processo de aplicação de eventos de mudança ao alvo está atrasado. Para identificar a latência em uma replicação contínua, você pode visualizar essa métrica junto com o CDC. LatencySource Se o CDC LatencyTarget estiver alto, mas o CDC LatencySource não, investigue se: • Não existe nenhuma chave primária ou índice no destino • Os gargalos de recursos ocorrem no destino ou na instância de replicação • Os problemas de rede residem entre a replicação e o destino

Métrica	Unidades	Descrição
Alvo do CDC ThroughputBandwidth	KB por segundo	Dados de saída transmitidos para o destino em KB por segundo. O CDC ThroughputBandwidth registra os dados de saída transmitidos nos pontos de amostragem. Se nenhum tráfego de rede for encontrado, o valor será zero. Como a CDC não emite transações prolongadas, o tráfego de rede pode não ser registrado.
Fonte CDC ThroughputRows	Linhas por segundo	As alterações de entrada na origem em linhas por segundo.
Alvo do CDC ThroughputRows	Linhas por segundo	As alterações de saída para o destino em linhas por segundo.
FullLoadThroughputBandwidthAlvo	KB por segundo	Dados de saída transmitidos de uma carga máxima para o destino em KB por segundo.
FullLoadThroughputRowsAlvo	Linhas por segundo	As alterações de saída de uma carga completa para o destino em linhas por segundo.

AWS DMS registros de replicação sem servidor

Você pode usar CloudWatch a Amazon para registrar informações de replicação durante um processo de AWS DMS migração. Você ativa o registro em log quando seleciona as configurações da replicação.

As replicações sem servidor carregam registros de status em sua CloudWatch conta para fornecer maior visibilidade do progresso da replicação e ajudar na solução de problemas.

AWS DMS carrega registros vinculados sem servidor para um grupo de registros dedicado com o prefixo. `dms-serverless-replication-`*`<your replication config resource ID>`* Nesse desse grupo de logs, há um fluxo de logs chamado `dms-serverless-replication-orchestrator-`*`<your replication config resource ID>`*. Esse fluxo de logs relata o estado da replicação e uma mensagem associada fornecendo mais detalhes sobre o trabalho que

está sendo realizado nesse estágio. Para obter exemplos de entradas de logs, consulte o Exemplos de logs de replicações que usam tecnologia sem servidor a seguir.

> **Note**
>
> AWS DMS não cria o grupo de logs nem o stream até que você execute a replicação. AWS DMS não cria o grupo de registros ou o stream se você criar apenas a replicação.

Para visualizar logs de uma tarefa executada, siga estas etapas:

1. Abra o AWS DMS console e escolha Replicações sem servidor no painel de navegação. A caixa de diálogo Replicações que usam tecnologia sem servidor é exibida.
2. Vá para a seção Configuração e escolha Visualizar logs sem servidor na coluna Geral. O grupo CloudWatch de registros é aberto.
3. Localize a seção Registros de tarefas de migração e escolha Exibir CloudWatch registros.

Se sua replicação falhar, AWS DMS cria uma entrada de registro com um estado de `failed` replicação e uma mensagem descrevendo o motivo da falha. Você deve verificar seus CloudWatch registros como a primeira etapa para solucionar uma falha na replicação.

> **Note**
>
> Assim como no AWS DMS Classic, você tem a opção de ativar um registro mais granular sobre o progresso da própria migração de dados; ou seja, os registros emitidos pela tarefa de replicação subjacente. É possível habilitar esses logs nas suas configurações de replicação definindo o `EnableLogging` no campo `Logging` como `true`. Veja o exemplo de JSON a seguir:
>
> ```
> {
>   "Logging": {
>     "EnableLogging": true
>   }
> }
> ```
>
> Se você ativar esses logs, eles só começarão a aparecer durante o estágio `running` da replicação que utiliza tecnologia sem servidor. Eles aparecerão no mesmo grupo de logs do fluxo de logs anterior, mas estarão no novo fluxo de logs `dms-serverless-serv-res-`

`id-{unique identifier}`. Consulte a seção a seguir para obter informações sobre como interpretar logs de replicações que usam tecnologia sem servidor.

Exemplos de logs de replicações que usam tecnologia sem servidor

Esta seção inclui um exemplo de entradas de log para replicações que usam tecnologia sem servidor.

Exemplo: Início da replicação

Quando você executa uma replicação sem servidor, AWS DMS cria uma entrada de registro semelhante à seguinte:

```
{'replication_state':'initializing', 'message': 'Initializing the replication
 workflow.'}
```

Exemplo: Falha na replicação

Se um dos endpoints da replicação não estiver configurado corretamente, AWS DMS cria uma entrada de registro semelhante à seguinte:

```
{'replication_state':'failed', 'message': 'Test connection failed for endpoint X.',
 'failure_message': 'X'}
```

Se você vir essa mensagem no log após uma falha, verifique se o endpoint especificado está íntegro e configurado corretamente.

Taxa de transferência aprimorada para migrações de carga completa de Oracle para Amazon Redshift

AWS DMS fornece um desempenho de taxa de transferência significativamente aprimorado para migrações de carga total da Oracle para o Amazon Redshift. O DMS ativa automaticamente esse recurso para tabelas sem a `custom parallel-load` opção em seus mapeamentos de tabela. Para tabelas com opções personalizadas de carregamento paralelo, o DMS serverless distribui a carga da tabela com base nas configurações de mapeamento de tabela fornecidas. Para usar a taxa de transferência aprimorada, faça o seguinte:

- Forneça regras de seleção que não façam referência a partições ou limites. Por exemplo, se as configurações da tabela nos mapeamentos de tabela contiverem`parallel-load`, o DMS

Serverless não usará o recurso de taxa de transferência aprimorada. Para ter mais informações, consulte Regras de seleção e ações.

- `WriteBufferSize`Defina `MaxFileSize` e para 64 MB. Para ter mais informações, consulte Configurações de endpoint ao utilizar o Amazon Redshift como destino do AWS DMS.
- Recomendamos configurar `CompressCsvFiles` `true` para um armazenamento de dados com dados esparsos e `false` para um armazenamento de dados com dados densos.
- Defina as seguintes configurações de tarefa como`0`:
  - `ParallelLoadThreads`
  - `ParallelLoadQueuesPerThread`
  - `ParallelApplyThreads`
  - `ParallelApplyQueuesPerThread`
  - `ParallelLoadBufferSize`
- `MaxFullLoadSubTasks`Defina como 49 para suportar a migração paralela de dados.
- Defina `LOB mode` como `inline`. Para ter mais informações, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS.

AWS DMS não fornece desempenho aprimorado de taxa de transferência para as seguintes replicações:

- Replicações com tabelas usando carga paralela. Para ter mais informações, consulte Utilizar carga paralela para tabelas, visualizações e coleções selecionadas.
- Replicações com regras de transformação de dados.
- Replicações com regras de filtro.
- Replicações com a regra de `change-data-type` transformação.

AWS DMS Limitações sem servidor

AWS DMS O Serverless tem as seguintes limitações:

- Você só pode modificar uma configuração AWS DMS de replicação que esteja nos `FAILED` estados `CREATED``STOPPED`, ou. Para obter detalhes sobre quais configurações podem ser alteradas sob quais condições, consulte Modificando replicações AWS DMS sem servidor.
- Você só pode excluir uma configuração AWS DMS de replicação que esteja nos `STOPPED` `FAILED` estados ou.

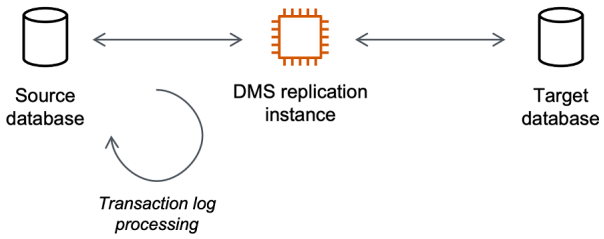
- Um armazenamento estático alocado de 100 GB está disponível para uma replicação. Se a replicação utilizar mais memória do que isso, devido a requisitos, como transações de longa execução ou armazenamento em cache, é recomendável particionar a workload em replicações que usam tecnologia sem servidor separadas. É possível particionar a workload por tabela ou por requisito, por exemplo, colocando toda a replicação que envolver LOBs em uma replicação que utiliza tecnologia sem servidor separada.
- Diferentemente das instâncias de replicação, as replicações AWS DMS sem servidor não têm um endereço IP público para tarefas de gerenciamento. Você gerencia replicações que usam tecnologia sem servidor utilizando o console.
- Essa versão do AWS DMS serverless não oferece suporte a todos os tipos de endpoints de origem e destino compatíveis com o AWS DMS padrão. Para obter uma lista dos tipos de mecanismo compatível, consulte AWS DMS Componentes sem servidor.
- As replicações que usam tecnologia sem servidor precisam acessar dependências utilizando endpoints da VPC. Utilize endpoints da VPC para acessar os seguintes tipos de endpoint:
  - Amazon S3
  - Amazon Kinesis
  - AWS Secrets Manager
  - Amazon DynamoDB
  - Amazon Redshift
  - OpenSearch Serviço Amazon

  Para obter informações sobre como configurar endpoints da VPC, consulte Configurar endpoints da VPC como endpoints de origem e de destino do AWS.
- AWS DMS serverless não oferece suporte a visualizações com regras de seleção e transformação.
- AWS DMS O serverless não oferece suporte ao uso de chaves gerenciadas pelo AWS cliente. AWS DMS O serverless suporta apenas o uso da chave DMS padrão. Para ter mais informações, consulte Proteção de dados em AWS Database Migration Service.
- O DMS Serverless não oferece suporte a conexões SSL para endpoints do DB2.

Trabalhando com uma instância de AWS DMS replicação

Quando você AWS DMS cria uma instância de AWS DMS replicação, cria-a em uma instância do Amazon EC2 em uma nuvem privada virtual (VPC) baseada no serviço Amazon VPC. Essa instância de replicação é usada para executar a migração do seu banco de dados. Ao utilizar a instância de replicação, é possível obter alta disponibilidade e suporte a failover utilizando uma implantação multi-AZ ao selecionar a opção Multi-AZ.

Em uma implantação Multi-AZ, provisiona e mantém AWS DMS automaticamente uma réplica em espera síncrona da instância de replicação em uma zona de disponibilidade diferente. A instância de replicação primária é replicada em sincronia pelas Zonas de disponibilidade para a réplica em espera. Essa abordagem fornece redundância de dados, elimina congelamentos de E/S e minimiza picos de latência.



AWS DMS usa uma instância de replicação para se conectar ao armazenamento de dados de origem, ler os dados de origem e formatar os dados para consumo pelo armazenamento de dados de destino. A instância de replicação também carrega os dados no armazenamento de dados de destino. A maior parte desse processo ocorre na memória. No entanto, transações grandes podem exigir buffer no disco. Transações armazenadas em cache e arquivos de log também são gravados no disco.

Você pode criar uma instância AWS DMS de replicação nas seguintes AWS regiões.

Nome da região	Região	Endpoint	Protocolo	
Leste dos EUA (Ohio)	us-east-2	dms.us-east-2.amazonaws.com dms-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
Leste dos EUA	us-east-1	dms.us-east-1.amazonaws.com	HTTPS	

Nome da região	Região	Endpoint	Protocolo
(Norte da Virgínia)		dms-fips.us-east-1.amazonaws.com	HTTPS
Oeste dos EUA (N. da Califórnia)	us-west-1	dms.us-west-1.amazonaws.com dms-fips.us-west-1.amazonaws.com	HTTPS HTTPS
Oeste dos EUA (Oregon)	us-west-2	dms.us-west-2.amazonaws.com dms-fips.us-west-2.amazonaws.com	HTTPS HTTPS
África (Cidade do Cabo)	af-south-1	dms.af-south-1.amazonaws.com	HTTPS
Ásia-Pacífico (Hong Kong)	ap-east-1	dms.ap-east-1.amazonaws.com	HTTPS
Ásia-Pacífico (Hyderabad)	ap-south-2	dms.ap-south-2.amazonaws.com	HTTPS
Ásia-Pacífico (Jacarta)	ap-southeast-3	dms.ap-southeast-3.amazonaws.com	HTTPS
Ásia-Pacífico (Melbourne)	ap-southeast-4	dms.ap-southeast-4.amazonaws.com	HTTPS

Nome da região	Região	Endpoint	Protocolo	
Ásia-Pacífico (Mumbai)	ap-south-1	dms.ap-south-1.amazonaws.com	HTTPS	
Ásia-Pacífico (Osaka)	ap-northeast-3	dms.ap-northeast-3.amazonaws.com	HTTPS	
Ásia-Pacífico (Seul)	ap-northeast-2	dms.ap-northeast-2.amazonaws.com	HTTPS	
Ásia-Pacífico (Singapura)	ap-southeast-1	dms.ap-southeast-1.amazonaws.com	HTTPS	
Ásia-Pacífico (Sydney)	ap-southeast-2	dms.ap-southeast-2.amazonaws.com	HTTPS	
Ásia-Pacífico (Tóquio)	ap-northeast-1	dms.ap-northeast-1.amazonaws.com	HTTPS	
Canadá (Central)	ca-central-1	dms.ca-central-1.amazonaws.com	HTTPS	
Oeste do Canadá (Calgary)	ca-west-1	dms.ca-west-1.amazonaws.com	HTTPS	
Europa (Frankfurt)	eu-central-1	dms.eu-central-1.amazonaws.com	HTTPS	

Nome da região	Região	Endpoint	Protocolo	
Europa (Irlanda)	eu-west-1	dms.eu-west-1.amazonaws.com	HTTPS	
Europa (Londres)	eu-west-2	dms.eu-west-2.amazonaws.com	HTTPS	
Europa (Milão)	eu-south-1	dms.eu-south-1.amazonaws.com	HTTPS	
Europa (Paris)	eu-west-3	dms.eu-west-3.amazonaws.com	HTTPS	
Europa (Espanha)	eu-south-2	dms.eu-south-2.amazonaws.com	HTTPS	
Europa (Estocolmo)	eu-north-1	dms.eu-north-1.amazonaws.com	HTTPS	
Europa (Zurique)	eu-central-2	dms.eu-central-2.amazonaws.com	HTTPS	
Israel (Tel Aviv)	il-central-1	dms.il-central-1.amazonaws.com	HTTPS	
Oriente Médio (Barém)	me-south-1	dms.me-south-1.amazonaws.com	HTTPS	
Oriente Médio (Emirados Árabes Unidos)	me-central-1	dms.me-central-1.amazonaws.com	HTTPS	

Nome da região	Região	Endpoint	Protocolo	
América do Sul (São Paulo)	sa-east-1	dms.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud (Leste dos EUA)	us-gov-east-1	dms.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	dms.us-gov-west-1.amazonaws.com	HTTPS	

AWS DMS suporta uma AWS região especial chamada AWS GovCloud (US) que foi projetada para permitir que agências e clientes do governo dos EUA movam cargas de trabalho confidenciais para a nuvem. AWS GovCloud (US) aborda os requisitos regulatórios e de conformidade específicos do governo dos EUA. Para obter mais informações sobre AWS GovCloud (US), consulte O que é AWS GovCloud (EUA)?

Veja a seguir mais detalhes sobre instâncias de replicação.

Tópicos

- Escolhendo a instância de replicação AWS DMS certa para sua migração
- Seleção do melhor tamanho para uma instância de replicação
- Trabalhar com as versões de mecanismos de replicação
- Instâncias de replicação públicas e privadas
- Endereçamento IP e tipos de rede
- Configurar uma rede para uma instância de replicação
- Configurar uma chave de criptografia para a instância de replicação
- Criar uma instância de replicação

- Modificar uma instância de replicação
- Reinicializar uma instância de replicação
- Excluir uma instância de replicação
- Como trabalhar com a janela de manutenção do AWS DMS

Escolhendo a instância de replicação AWS DMS certa para sua migração

AWS DMS cria a instância de replicação em uma instância do Amazon EC2. AWS DMS atualmente oferece suporte às classes de instância T2, T3, C4, C5, C6i, R4, R5 e R6i do Amazon EC2 para instâncias de replicação:

- As instâncias T2 são instâncias de desempenho expansível com capacidade de intermitência que oferecem um nível básico de desempenho de CPU com capacidade de intermitência acima da linha de base. O desempenho de linha de base e a capacidade de intermitência são governados por créditos de CPU. As instâncias T2 recebem créditos de CPU continuamente a uma taxa definida, dependendo do tamanho da instância. Elas acumulam créditos de CPU quando estão ociosas e consomem créditos de CPU quando estão ativas.

  As instâncias T2 são uma boa opção para uma variedade de workloads de uso geral. Os exemplos incluem microsserviços, aplicações interativas de baixa latência, bancos de dados pequenos e médios, áreas de trabalho virtuais, ambientes de desenvolvimento, de criação e de preparação, repositórios de código e protótipos de produtos.
- As instâncias T3 são o tipo de instância de uso geral intermitente de próxima geração. Esse tipo fornece um nível de linha de base de desempenho de CPU com a capacidade de intermitência de uso de CPU a qualquer momento e pelo tempo necessário. As instâncias T3 oferecem recursos equilibrados de computação, de memória e de rede e são projetadas para aplicações com uso moderado de CPU que experimentam picos temporários de uso. As instâncias T3 acumulam créditos de CPU quando uma workload está operando abaixo do limite da linha de base. Cada crédito de CPU ganho oferece à instância T3 a oportunidade de apresentar intermitência de desempenho de um núcleo de CPU completo por um minuto, quando necessário.

  As instâncias T3 podem apresentar intermitência a qualquer momento pelo tempo que for necessário no modo `unlimited`. Para obter mais informações sobre o modo `unlimited`, consulte Como trabalhar com o modo ilimitado de instâncias de desempenho expansível.

- As instâncias C4 são otimizadas para workloads com uso intensivo de computação e oferecem alto desempenho e grande economia a um preço baixo por taxa de computação. Eles alcançam um desempenho significativamente maior de pacotes por segundo (PPS), menor instabilidade de rede e menor latência de rede. AWS DMS também pode consumir muita CPU, especialmente ao realizar migrações e replicações heterogêneas, como migrar do Oracle para o PostgreSQL. As instâncias C4 podem ser uma boa opção para essas situações.
- As instâncias C5 são o tipo de instância de próxima geração que oferece alto desempenho econômico a uma taxa de preço baixo por computação para executar workloads avançadas com uso intensivo de computação. Isso inclui workloads como servidores web de alto desempenho, computação de alta performance (HPC), processamento em lote, veiculação de anúncios, jogos multijogador altamente escaláveis e codificação de vídeo. Outras instâncias C5 de workloads são adequadas para incluir modelagem científica, análise distribuída e inferência de aprendizado de máquina e aprendizado profundo. As instâncias C5 estão disponíveis com uma variedade de processadores da Intel e da AMD.
- As instâncias C6i oferecem desempenho de preço de computação até 15% melhor do que as instâncias de quinta geração comparáveis para uma ampla variedade de workloads e criptografia de memória sempre ativa. As instâncias C6i são ideais para workloads com uso intensivo de computação, como processamento em lote, análise distribuída, computação de alta performance (HPC), veiculação de anúncios, jogos multijogador altamente escaláveis e codificação de vídeo.
- As instâncias R4 são otimizadas para memória para workloads com uso intensivo de memória. As replicações ou migrações contínuas de sistemas de transações de alto throughput que usam o AWS DMS , às vezes, também podem consumir grandes quantidades de CPU e de memória. As instâncias R4 incluem mais memória por vCPU do que os tipos de instância da geração anterior.
- As instâncias R5 são a próxima geração de tipos de instância otimizada para memória do Amazon EC2. As instâncias R5 são ideais para aplicações com uso intensivo de memória, como bancos de dados de alto desempenho, caches na memória distribuídos em escala de web, bancos de dados na memória de médio porte, análise de big data em tempo real e outras aplicações empresariais. As migrações ou replicações contínuas do uso de sistemas de transações de alto rendimento também AWS DMS podem consumir grandes quantidades de CPU e memória.
- As instâncias R6i oferecem desempenho de preço computacional até 15% melhor do que as instâncias de quinta geração comparáveis para uma ampla variedade de workloads e criptografia de memória sempre ativa. As instâncias R6i são certificadas pela SAP e são ideais para workloads, como bancos de dados SQL e noSQL, caches na memória distribuídos em escala web, como Memcached e Redis, bancos de dados na memória, como SAP HANA, e análise de big data em tempo real, como clusters Hadoop e Spark.

Cada instância de replicação tem uma configuração específica de memória e de vCPU. A tabela a seguir mostra a configuração de cada tipo de instância de replicação. Para obter informações sobre preços, consulte a página Preços do serviço AWS Database Migration Service.

Tipos de instância de replicação de uso geral

Tipo	vCPU	Memória (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

Tipos de instância de replicação otimizada para computação

Tipo	vCPU	Memória (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

Tipo	vCPU	Memória (GiB)
dms.c5.2xlarge	8	16
dms.c5.4xlarge	16	32
dms.c5.9xlarge	36	72
dms.c5.12xlarge	48	96
dms.c5.18xlarge	72	144
dms.c5.24xlarge	96	192
dms.c6i.large	2	4
dms.c6i.xlarge	4	8
dms.c6i.2xlarge	8	16
dms.c6i.4xlarge	16	32
dms.c6i.8xlarge	32	64
dms.c6i.12xlarge	48	96
dms.c6i.16xlarge	64	128
dms.c6i.24xlarge	96	192
dms.c6i.32xlarge	128	256

Tipos de instância de replicação otimizada para memória

Tipo	vCPU	Memória (GiB)
dms.r4.large	2	15.25
dms.r4.xlarge	4	30.5

Tipo	vCPU	Memória (GiB)
dms.r4.2xlarge	8	61
dms.r4.4xlarge	16	122
dms.r4.8xlarge	32	244
dms.r5.large	2	16
dms.r5.xlarge	4	32
dms.r5.2xlarge	8	64
dms.r5.4xlarge	16	128
dms.r5.8xlarge	32	256
dms.r5.12xlarge	48	384
dms.r5.16xlarge	64	512
dms.r5.24xlarge	96	768
dms.r6i.large	2	16
dms.r6i.xlarge	4	32
dms.r6i.2xlarge	8	64
dms.r6i.4xlarge	16	128
dms.r6i.8xlarge	32	256
dms.r6i.12xlarge	48	384
dms.r6i.16xlarge	64	512
dms.r6i.24xlarge	96	768
dms.r6i.32xlarge	128	1024

As tabelas acima listam todos os tipos de instância de AWS DMS replicação, mas os tipos disponíveis na sua região podem variar. Para ver os tipos de instância de replicação disponíveis na sua região, execute o seguinte comando da AWS CLI:

```
aws dms describe-orderable-replication-instances --region your_region_name
```

Tópicos

Como decidir a classe de instância a ser usada

Para ajudar a determinar qual classe de instância de replicação pode funcionar melhor para você, vamos examinar o processo de captura de dados de alteração (CDC) que AWS DMS usa.

Vamos supor que você esteja executando uma tarefa de carga completa mais CDC (carga em lote mais replicação contínua). Nesse caso, a tarefa tem seu próprio repositório SQLite para armazenar metadados e outras informações. Antes de AWS DMS iniciar uma carga completa, estas etapas ocorrem:

- AWS DMS começa a capturar as alterações nas tabelas que está migrando do registro de transações do mecanismo de origem (chamamos essas alterações em cache). Assim que a carga máxima é concluída, essas alterações armazenadas em cache são coletadas e aplicadas no destino. Dependendo do volume de alterações em cache, essas alterações podem ser aplicadas diretamente da memória, onde são coletadas primeiro, até um limite definido. Como alternativa, elas podem ser aplicadas no disco, onde as alterações são gravadas quando não podem ser mantidas na memória.
- Depois que as alterações em cache são aplicadas, por padrão, AWS DMS inicia um processo de aplicação transacional na instância de destino.

Durante a fase de alterações aplicadas em cache e a fase de replicações contínuas, AWS DMS usa dois buffers de fluxo, um para dados de entrada e saída. AWS DMS também usa um componente importante chamado classificador, que é outro buffer de memória. Veja a seguir duas utilizações importantes do componente classificador (que também possui outras):

- Ele monitora todas as transações e garante que encaminha somente as transações relevantes ao buffer de saída.

- Ele garante que as transações são encaminhadas na mesma ordem de confirmação como na origem.

Como é possível ver, temos três buffers de memória importantes nessa arquitetura para a CDC no AWS DMS. Se qualquer um desses buffers de memória apresentar pressão de memória, a migração pode ter problemas de desempenho que podem causar falhas.

Ao conectar workloads pesadas com um alto número de transações por segundo (TPS) nessa arquitetura, a memória adicional fornecida pelas instâncias R5 e R6i pode ser útil. Use instâncias R5 e R6i para manter o grande número de transações na memória e evitar problemas de pressão de memória durante as replicações contínuas.

Como trabalhar com o modo ilimitado de instâncias de desempenho expansível

Uma instância de desempenho expansível configurada como `unlimited`, como uma instância T3, pode sustentar alta utilização de CPU por qualquer período, sempre que necessário. O preço por hora da instância pode cobrir automaticamente todos os picos de uso da CPU. Isso ocorre se a utilização média de CPU da instância for igual ou menor que a linha de base durante um período contínuo de 24 horas ou durante a vida útil da instância, o que for menor.

Na grande maioria das workloads de uso geral, as instâncias configuradas como `unlimited` fornecem um desempenho suficiente sem cobranças adicionais. Se a instância funcionar com maior utilização de CPU por um período prolongado, ela poderá fazer isso por uma taxa adicional uniforme por hora de vCPU. Para obter informações sobre preços de instâncias T3, consulte “Créditos de CPU T3” no AWS Database Migration Service.

Para obter mais informações sobre o `unlimited` modo para instâncias T3, consulte Modo ilimitado para instâncias de desempenho intermitente no Guia do usuário do Amazon EC2.

> **Important**
>
> Se você utilizar uma instância `dms.t3.micro` da oferta de nível gratuito da AWS e utilizá-la no modo `unlimited`, poderão ser aplicadas cobranças. Especificamente, as cobranças poderão ser aplicadas se a sua utilização média durante um período contínuo de 24 horas exceder a utilização de linha de base da instância. Para obter mais informações, consulte Utilização básica no Guia do usuário do Amazon EC2.

As instâncias T3 são executadas como `unlimited` por padrão. Se a média de uso de CPU em um período de 24 horas exceder a linha de base, você incorrerá em cobranças por créditos excedentes. Em alguns casos, é possível executar instâncias spot T3 como `unlimited` e planejar utilizá-las imediatamente e por um curto período. Ao fazer isso sem tempo ocioso para acumular créditos de CPU, serão cobrados créditos excedentes. É recomendável iniciar as instâncias spot T3 no modo padrão para evitar custos mais altos. Para obter mais informações, consulte Créditos excedentes podem incorrer em cobranças, Instâncias T3 Spot e Modo padrão para instâncias de desempenho intermitente no Guia do usuário do Amazon EC2.

Seleção do melhor tamanho para uma instância de replicação

A escolha da instância de replicação adequada depende de vários fatores do seu caso de uso. Para ajudar a entender como os recursos da instância de replicação são utilizados, consulte a discussão a seguir. Ela abrange o cenário comum de uma tarefa de carga máxima + CDC.

Durante uma tarefa de carga completa, AWS DMS carrega as tabelas individualmente. Por padrão, oito tabelas são carregadas por vez. AWS DMS captura as alterações contínuas na origem durante uma tarefa de carga total para que as alterações possam ser aplicadas posteriormente no endpoint de destino. As alterações são armazenados em cache na memória; se a memória disponível for esgotada, as alterações são armazenadas em cache no disco. Quando uma tarefa de carga completa é concluída para uma tabela, as alterações em cache são aplicadas AWS DMS imediatamente à tabela de destino.

Depois que todas as alterações em cache pendentes para uma tabela forem aplicadas, o endpoint de destino está em um estado transacionalmente consistente. Nesse ponto, o destino está sincronizado com o endpoint de origem em relação às últimas alterações em cache. AWS DMS em seguida, inicia a replicação contínua entre a origem e o destino. Para fazer isso, AWS DMS pega as operações de alteração dos registros de transações de origem e as aplica ao destino de maneira transacionalmente consistente. (Esse processo pressupõe que a aplicação otimizada em lote não esteja selecionada). AWS DMS transmite as alterações contínuas pela memória na instância de replicação, se possível. Caso contrário, AWS DMS grava as alterações no disco na instância de replicação até que elas possam ser aplicadas no destino.

Você tem algum controle sobre como a instância de replicação manipula o processamento de alterações e como a memória é usada nesse processo. Para obter mais informações sobre como

ajustar o processamento de alterações, consulte Configurações de ajuste de processamento de alterações.

Fatores a serem considerados

Memória e espaço em disco são fatores-chave na seleção de uma instância de replicação apropriada para o seu caso de uso. Veja uma discussão a seguir sobre as características do caso de uso a serem analisadas para escolher uma instância de replicação.

- Tamanho do banco de dados e das tabelas

  O volume de dados ajuda a determinar a configuração da tarefa para otimizar o desempenho da carga máxima. Por exemplo, para dois esquemas de 1 TB, é possível particionar tabelas em quatro tarefas de 500 GB e executá-las em paralelo. O paralelismo possível depende do recurso de CPU disponível na instância de replicação. Por isso, é uma boa ideia compreender o tamanho do banco de dados e das tabelas para otimizar o desempenho da carga máxima. Isso ajuda a determinar o número de tarefas possíveis que você pode ter.

- Objetos grandes

  Os tipos de dados presentes no escopo da migração podem afetar o desempenho. Particularmente, os objetos grandes (LOBs) afetam o desempenho e o consumo de memória. Para migrar um valor de LOB, AWS DMS executa um processo de duas etapas. Primeiro, AWS DMS insere a linha no destino sem o valor LOB. Em segundo lugar, AWS DMS atualiza a linha com o valor LOB. Isso tem um impacto na memória, portanto, é importante identificar as colunas de LOB na origem e analisar o tamanho dessas colunas.

- Frequência da carga e tamanho das transações

  A frequência da carga e as transações por segundo (TPS) influenciam o uso da memória. Um grande número de atividades de TPS ou da linguagem de manipulação de dados (DML) resultam em uma alta utilização de memória. Isso ocorre porque o DMS armazena em cache as alterações até que elas sejam aplicadas ao destino. Durante a CDC, isso resulta em troca (gravação no disco físico devido ao estouro de memória), o que provoca latência.

- Chaves de tabelas e integridade referencial

  As informações sobre as chaves de tabela determinam o modo da CDC (aplicação em lote ou aplicação transacional) que você utiliza para migrar os dados. Em geral, a aplicação transacional é mais lenta do que a aplicação em lote. Para transações de longa execução, pode haver muitas mudanças na migração. Quando você usa a aplicação transacional, AWS DMS pode ser

necessária mais memória para armazenar as alterações em comparação com a aplicação em lote. Se você migrar tabelas sem chaves primárias, a aplicação em lote falhará e a tarefa do DMS será movida para o modo de aplicação transacional. Quando a integridade referencial está ativa entre as tabelas durante o CDC, AWS DMS usa a aplicação transacional por padrão. Para obter mais informações sobre a aplicação em lote em comparação com a aplicação transacional, consulte Como posso utilizar o recurso de aplicação em lote do DMS para melhorar o desempenho da replicação de CDC?

Use essas métricas para determinar se é necessário que a instância de replicação seja otimizada para computação ou para memória.

Problemas comuns

É possível enfrentar os seguintes problemas comuns que causam contenção de recursos na instância de replicação durante a migração. Para obter informações sobre as métricas instância de replicação, consulte Métricas de instâncias de replicação.

- Se a memória em uma instância de replicação se tornar insuficiente, isso resultará na gravação dos dados no disco. A leitura de disco pode causar latência, o que pode ser evitado com o dimensionamento da instância de replicação com memória suficiente.
- O tamanho do disco atribuído à instância de replicação pode ser menor do que o necessário. O tamanho do disco é utilizado quando os dados na memória transbordam. Ele também é utilizado para armazenar os logs de tarefas. O IOPS máximo também depende disso.
- A execução de várias tarefas ou tarefas com alto paralelismo afeta o consumo de CPU da instância de replicação. Isso retarda o processamento das tarefas e resulta em latência.

Práticas recomendadas

Considere estas duas práticas recomendadas mais comuns ao dimensionar uma instância de replicação. Para ter mais informações, consulte Práticas recomendadas do AWS Database Migration Service.

1. Dimensione a workload e compreenda se ela tem consumo intensivo de computação ou de memória. Com base nisso, é possível determinar a classe e o tamanho da instância de replicação:
   - AWS DMS processa LOBs na memória. Essa operação requer uma boa quantidade de memória.

- O número de tarefas e o número de threads afetam o consumo da CPU. Evite utilizar mais de oito `MaxFullLoadSubTasks` durante a operação de carga máxima.

2. Aumente o espaço em disco atribuído à instância de replicação quando tiver uma workload alta durante a carga máxima. Isso permite que a instância de replicação utilize o máximo de IOPS atribuído a ela.

As orientações anteriores não abrangem todos os cenários possíveis. É importante considerar os detalhes do seu caso de uso específico ao determinar o tamanho da instância de replicação.

Os testes anteriores mostram que a CPU e a memória variam com diferentes workloads. Particularmente, os LOBs afetam a memória, e a contagem de tarefas ou o paralelismo afetam a CPU. Depois que a migração estiver em execução, monitore a CPU, a memória liberável, o armazenamento livre e as IOPS da instância de replicação. Com base nos dados coletados, é possível dimensionar a instância de replicação para cima ou para baixo, conforme necessário.

Trabalhar com as versões de mecanismos de replicação

O mecanismo de replicação é o AWS DMS software principal executado em sua instância de replicação e executa as tarefas de migração que você especificar. AWS lança periodicamente novas versões do software do mecanismo de AWS DMS replicação, com novos recursos e melhorias de desempenho. Cada versão do software do mecanismo de replicação tem seu próprio número de versão, para diferenciá-lo de outras versões.

Quando você executa uma nova instância de replicação, ela executa a versão mais recente do AWS DMS mecanismo, a menos que você especifique o contrário. Para ter mais informações, consulte Trabalhando com uma instância de AWS DMS replicação.

Se você tiver uma instância de replicação em execução no momento, poderá atualizá-la para uma versão mais recente do mecanismo. (AWS DMS não suporta downgrades de versão do motor.) Para obter mais informações sobre versões do mecanismo de replicação, consulte AWS Notas de versão do DMS.

Atualizar a versão do mecanismo utilizando o console

Você pode atualizar uma instância AWS DMS de replicação usando o. AWS Management Console

Para atualizar uma instância de replicação utilizando o console

1. Abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

2. No painel de navegação, selecione Replication instances.
3. Escolha seu mecanismo de replicação e, em seguida, escolha Modify.
4. Em Versão do mecanismo, escolha o número da versão e escolha Modificar.

> **Note**
>
> É recomendável interromper todas as tarefas antes de atualizar a instância de replicação. Se você não interromper a tarefa, a AWS DMS interromperá automaticamente antes da atualização. Ao interromper a tarefa manualmente, será necessário iniciá-la manualmente após a conclusão da atualização. A atualização da instância de replicação pode levar alguns minutos. Quando a instância estiver pronta, seu status mudará para available.

Atualizando a versão do motor usando o AWS CLI

Você pode atualizar uma instância de AWS DMS replicação usando o AWS CLI, da seguinte forma.

Para atualizar uma instância de replicação usando o AWS CLI

1. Determine o nome de recurso da Amazon (ARN) da instância de replicação utilizando o comando a seguir.

   ```
   aws dms describe-replication-instances \
   --query "ReplicationInstances[*].
   [ReplicationInstanceIdentifier,ReplicationInstanceArn,ReplicationInstanceClass]"
   ```

   No resultado, anote o ARN referente à instância de replicação que você deseja atualizar, por exemplo: `arn:aws:dms:us-east-1:123456789012:rep:6EFQQO6U6EDPRCPKLNPL2SCEEY`
2. Determine quais versões da instância de replicação estão disponíveis utilizando o comando a seguir.

   ```
   aws dms describe-orderable-replication-instances \
   --query "OrderableReplicationInstances[*].[ReplicationInstanceClass,EngineVersion]"
   ```

   No resultado, anote o número ou os números de versão do mecanismo disponíveis para a classe de instância de replicação. Você encontra essas informações no resultado da Etapa 1.

3. Atualize uma instância de replicação utilizando o seguinte comando.

```
aws dms modify-replication-instance \
--replication-instance-arn arn \
--engine-version n.n.n
```

Substitua *arn* pelo ARN da instância de replicação da etapa anterior.

Substitua *n.n.n* pelo número da versão do mecanismo desejado, por exemplo: `3.4.5`

> **Note**
>
> A atualização da instância de replicação pode levar alguns minutos. É possível visualizar o status da instância de replicação com o seguinte comando.
>
> ```
> aws dms describe-replication-instances \
> --query "ReplicationInstances[*].
> [ReplicationInstanceIdentifier,ReplicationInstanceStatus]"
> ```
>
> Quando a instância de replicação estiver pronta, seu status mudará para available.

Instâncias de replicação públicas e privadas

É possível especificar se uma instância de replicação tem um endereço IP público ou privado que ela utiliza para se conectar aos bancos de dados de origem e de destino.

Uma instância de replicação privada tem um endereço IP privado que não pode ser acessado fora da rede de replicação. Utilize uma instância privada quando os bancos de dados de origem e de destino estão na mesma rede conectada à nuvem privada virtual (VPC) da instância de replicação. A rede pode ser conectada à VPC usando uma rede privada virtual (VPN) ou emparelhamento de AWS Direct Connect VPC.

Uma conexão de emparelhamento de VPC é uma conexão de rede entre duas VPCs. Ela permite o roteamento utilizando os endereços IP privados de cada VPC como se estivessem na mesma rede. Para obter mais informações sobre emparelhamento de VPC, consulte Emparelhamento de VPC, no Guia do usuário da Amazon VPC.

Uma instância de replicação pública pode utilizar o grupo de segurança da VPC da instância de replicação e o endereço IP público da instância de replicação ou o endereço IP público do gateway NAT. Essas conexões formam uma rede usada para a migração de dados.

Endereçamento IP e tipos de rede

AWS DMS sempre cria sua instância de replicação em uma Amazon Virtual Private Cloud (VPC). Ao criar a VPC, é possível determinar o endereçamento IP a ser utilizado: IPv4, IPv6 ou ambos. Ao criar ou modificar uma instância de replicação, é possível especificar a utilização de um protocolo de endereço IPv4 ou de um protocolo de endereço IPv6 utilizando o Modo de pilha dupla.

Endereços IPv4

Ao criar a VPC, é possível especificar um intervalo de endereços IPv4 para a VPC na forma de um bloco de Encaminhamento Entre Domínios Sem Classificação (CIDR); como 10.0.0.0/16. Um grupo de sub-redes define o intervalo de endereços IP nesse bloco CIDR. Esses endereços IP podem ser públicos ou privados.

Um endereço IPv4 privado é um endereço IP que não é acessível pela internet. É possível utilizar endereços IPv4 privados para a comunicação entre a instância de replicação e os outros recursos, como instâncias do Amazon EC2, na mesma VPC. Cada instância de replicação tem um endereço IP privado para comunicação na VPC.

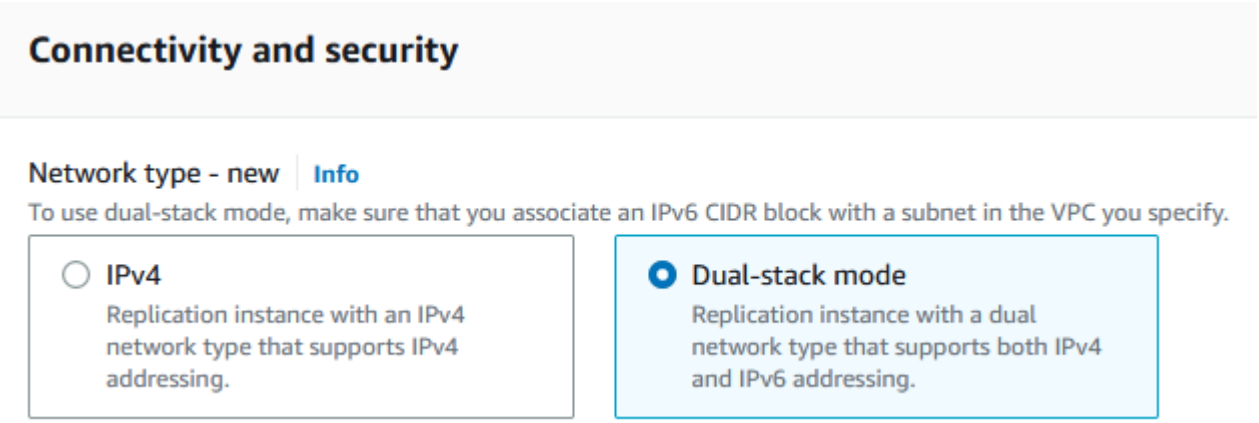
Um endereço IP público é um endereço IPv4 que é acessível pela internet. É possível utilizar endereços públicos para comunicação entre as instâncias de replicação e os recursos na internet. Você controla se a instância de replicação recebe um endereço IP público.

Modo de pilha dupla e endereços IPv6

Quando houver recursos que precisam se comunicar com a instância de replicação pelo IPv6, utilize o modo de pilha dupla. Para utilizar o modo de pilha dupla, verifique se cada sub-rede no grupo de sub-redes que você associa à instância de replicação tem um bloco CIDR IPv6 associado a ela. É possível criar um grupo de sub-redes de replicação ou modificar um existente para atender a esse requisito. Todo endereço IPv6 é globalmente exclusivo. O bloco CIDR IPv6 da VPC é atribuído automaticamente do grupo de endereços IPv6 da Amazon. Você não pode escolher o intervalo.

O DMS desativa o acesso ao gateway da Internet para endpoints IPv6 de instâncias de replicação do modo de pilha dupla. O DMS faz isso para garantir que os endpoints IPv6 sejam privados e possam ser acessados somente de dentro da VPC.

Você pode usar o AWS DMS console para criar ou modificar uma instância de replicação e especificar o modo de pilha dupla na seção Tipo de rede. A imagem a seguir mostra a seção Tipo de rede no console.



Referências

- Para obter mais informações sobre endereços IPv4 e IPv6, consulte Endereçamento IP no Guia do usuário da Amazon VPC.
- Para obter mais informações sobre como criar uma instância de replicação utilizando o modo de pilha dupla, consulte Criar uma instância de replicação.
- Para obter mais informações sobre como modificar uma instância de replicação, consulte Modificar uma instância de replicação.

Configurar uma rede para uma instância de replicação

AWS O DMS sempre cria a instância de replicação em uma VPC baseada na Amazon VPC. Especifique a VPC em que a instância de replicação está localizada. Você pode usar sua VPC padrão para sua conta e AWS região, ou você pode criar uma nova VPC.

Verifique se a interface de rede elástica alocada para a VPC da instância de replicação está associada a um grupo de segurança. Além disso, verifique se as regras desse grupo de segurança permitem que todo o tráfego em todas as portas saia da VPC. Essa abordagem permite a comunicação da instância de replicação com os endpoints dos bancos de dados de origem e de destino, desde que as regras de entrada corretas estejam ativadas neles. É recomendável que você utilize as configurações padrão para os endpoints, que permitem a saída em todas as portas para todos os endereços.

Os endpoints de origem e de destino acessam a instância de replicação que está dentro do VPC, conectando-se ao VPC ou estando no VPC. Os endpoints do banco de dados devem incluir listas de controle de acesso (ACLs) das regras de rede e regras de grupo de segurança (se aplicável) que permitam o acesso de entrada na instância de replicação. A forma como você configura isso depende da configuração da rede utilizada. É possível utilizar o grupo de segurança da VPC da instância de replicação, o endereço IP público ou privado da instância de replicação ou o endereço IP público do gateway NAT. Essas conexões formam uma rede usada para a migração de dados.

> **Note**
>
> Como um endereço IP pode mudar como resultado de alterações na infraestrutura subjacente, é recomendável utilizar um intervalo de CIDR de VPC ou rotear o tráfego de saída da instância de replicação por meio de um IP elástico associado ao GW NAT. Para obter mais informações sobre como criar uma VPC, incluindo um bloco CIDR, consulte Como trabalhar com VPCs e sub-redes no Guia do usuário da Amazon Virtual Private Cloud. Para obter informações sobre endereços IP elásticos, consulte Endereços IP elásticos no Guia do usuário do Amazon Elastic Compute Cloud.

Configurações de rede para migração de banco de dados

Você pode usar várias configurações de rede diferentes com o AWS Database Migration Service. A seguir, veja configurações comuns de uma rede usada para a migração de banco de dados.

Tópicos

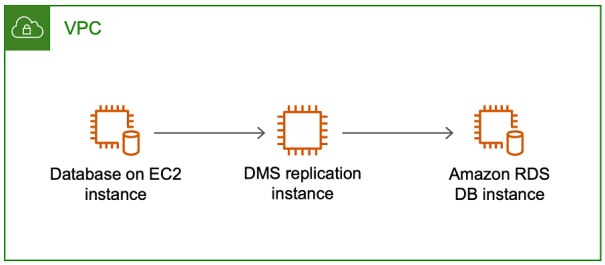
- Configuração com todos os componentes de migração de banco de dados em uma VPC
- Configuração com várias VPCs
- Configuração com VPCs compartilhadas
- Configuração de uma rede para uma VPC usando AWS Direct Connect ou uma VPN
- Configuração de uma rede para uma VPC utilizando a internet
- Configuração com uma instância de banco de dados RDS fora de uma VPC para uma instância de banco de dados em uma VPC usando ClassicLink

Quando possível, é recomendável criar uma instância de replicação do DMS na mesma região do endpoint de destino e na mesma VPC ou sub-rede do endpoint de destino.

Configuração com todos os componentes de migração de banco de dados em uma VPC

A rede mais simples de migração de banco de dados é para que o endpoint de origem, a instância de replicação e o endpoint de destino estejam no mesmo VPC. Essa configuração será adequada se os endpoints de origem e de destino estiverem em uma instância de banco de dados Amazon RDS ou em uma instância do Amazon EC2.

A ilustração a seguir mostra uma configuração em que um banco de dados em uma instância do Amazon EC2 se conecta à instância de replicação, e os dados são migrados para uma instância de banco de dados Amazon RDS.



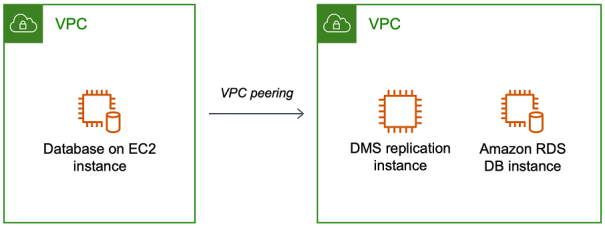
O grupo de segurança da VPC utilizado nessa configuração deve permitir a entrada na porta do banco de dados na instância de replicação. É possível fazer isso de duas maneiras. É possível verificar se o grupo de segurança utilizado pela instância de replicação tem acesso aos endpoints. Ou permitir o intervalo CIDR da VPC, o IP elástico do GW NAT ou o endereço IP privado da instância de replicação, se estiver utilizando um. Mas não é recomendável utilizar o endereço IP privado da instância de replicação, pois isso poderá interromper a replicação se o endereço IP da replicação for alterado.

Configuração com várias VPCs

Se os endpoints de origem e de destino estiverem em VPCs diferentes, crie a instância de replicação em uma das VPCs. É possível vincular as duas VPCs utilizando o emparelhamento de VPC.

Uma conexão de emparelhamento de VPC é uma conexão de redes entre duas VPCs, que permite roteamento utilizando endereços IP privados de cada VPC como se estivessem na mesma rede. Você pode criar uma conexão de emparelhamento de VPC entre suas próprias VPCs, com uma VPC em outra AWS conta ou com uma VPC em uma região diferente. AWS Para obter mais informações sobre emparelhamento de VPC, consulte Emparelhamento de VPC, no Guia do usuário da Amazon VPC.

A ilustração a seguir mostra um exemplo de configuração utilizando o emparelhamento de VPCs. Aqui, o banco de dados de origem em uma instância do Amazon EC2 em uma VPC se conecta pelo emparelhamento de VPC a uma VPC. Essa VPC contém a instância de replicação e o banco de dados de destino em uma instância de banco de dados Amazon RDS.



Para implementar o emparelhamento de VPC, siga as instruções em Como trabalhar com conexões de emparelhamento de VPC encontradas na documentação do Amazon Virtual Private Cloud, emparelhamento de VPC. Verifique se a tabela de rotas de uma VPC contém o bloco CIDR da outra. Por exemplo, se a VPC A estiver utilizando o destino 10.0.0.0/16 e a VPC B estiver utilizando o destino 172.31.0.0, a tabela de rotas da VPC A deverá conter 172.31.0.0, e a tabela de rotas da VPC B deverá conter 10.0.0.0/16. Para obter informações mais detalhadas, consulte Atualizar as tabelas de rotas para a conexão de emparelhamento da VPC na documentação da Amazon Virtual Private Cloud, emparelhamento de VPC.

Os grupos de segurança da VPC utilizados nessa configuração devem permitir a entrada na porta do banco de dados da instância de replicação ou permitir a entrada no bloco CIDR da VPC que está sendo emparelhada.

Configuração com VPCs compartilhadas

AWS DMS trata as sub-redes que são compartilhadas com uma conta de cliente participante em uma organização da mesma forma que as sub-redes normais na mesma conta. Abaixo está uma descrição de como AWS DMS lida com VPCs, sub-redes e como você pode usar VPCs compartilhadas.

É possível configurar a rede para operar em sub-redes ou VPCs personalizadas criando objetos `ReplicationSubnetGroup`. Ao criar um `ReplicationSubnetGroup`, é possível optar por especificar sub-redes de uma VPC específica na sua conta. A lista de sub-redes que você especificar deve incluir pelo menos duas sub-redes que estejam em zonas de disponibilidade separadas, e todas as sub-redes devem estar na mesma VPC. Ao criar um`ReplicationSubnetGroup`, os clientes especificam apenas sub-redes. AWS DMS determinará a VPC em seu nome, pois cada sub-rede está vinculada a exatamente uma VPC.

Ao criar um AWS DMS `ReplicationInstance` ou um AWS DMS `ReplicationConfig`, você pode optar por especificar um `ReplicationSubnetGroup` e/ou um grupo de segurança da VPC no qual a replicação `ReplicationInstance` ou a replicação sem servidor opera. Se não for especificado, AWS DMS escolhe o padrão do cliente `ReplicationSubnetGroup` (que é AWS DMS criado em seu nome se não for especificado para todas as sub-redes na VPC padrão) e o grupo de segurança da VPC padrão.

É possível optar por executar as migrações em uma zona de disponibilidade especificada por você ou em qualquer uma das zonas de disponibilidade em seu `ReplicationSubnetGroup`. Ao AWS DMS tentar criar uma instância de replicação ou iniciar uma replicação sem servidor, isso traduz as zonas de disponibilidade de suas sub-redes em zonas de disponibilidade na conta de serviço principal, para garantir que executemos instâncias na zona de disponibilidade correta, mesmo que os mapeamentos da zona de disponibilidade não sejam idênticos entre as duas contas.

Se você utilizar uma VPC compartilhada, precisará garantir que cria os objetos do `ReplicationSubnetGroup` mapeados para as sub-redes que deseja utilizar a partir em uma VPC compartilhada. Ao criar um `ReplicationInstance` ou um `ReplicationConfig`, especifique um `ReplicationSubnetGroup` para a VPC compartilhada e especifique um grupo de segurança de VPC que você criou para a VPC compartilhada com a solicitação Create.

Observe o seguinte sobre a utilização de uma VPC compartilhada:

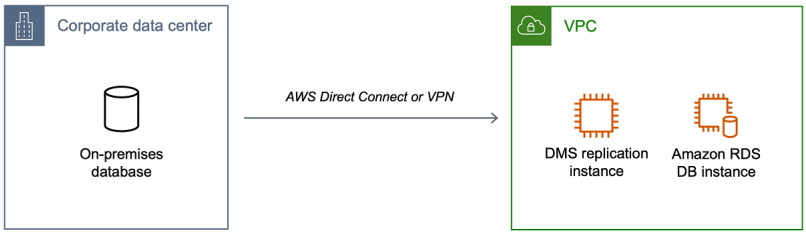
- O proprietário da VPC não pode compartilhar um recurso com um participante, mas o participante pode criar um recurso de serviço na sub-rede do proprietário.
- O proprietário da VPC não pode acessar um recurso (como uma instância de replicação) criado pelo participante, porque todos os recursos são específicos da conta. No entanto, desde que você crie a instância de replicação na VPC compartilhada, ela poderá acessar os recursos na VPC independentemente da conta proprietária, desde que o endpoint ou a tarefa de replicação tenha as permissões corretas.
- Como os recursos são específicos da conta, outros participantes não podem acessar recursos pertencentes a outras contas. Não há permissões que você possa conceder a outras contas para permitir que elas acessem os recursos criados na VPC compartilhada com a sua conta.

Configuração de uma rede para uma VPC usando AWS Direct Connect ou uma VPN

As redes remotas podem se conectar a uma VPC usando várias opções, como AWS Direct Connect ou uma conexão VPN de software ou hardware. Essas opções costumam ser usadas para integrar

serviços locais existentes, como monitoramento, autenticação, segurança, dados ou outros sistemas, estendendo uma rede interna para a nuvem AWS . Usar esse tipo de extensão de rede facilita a conexão integrada a recursos hospedados na AWS, como uma VPC.

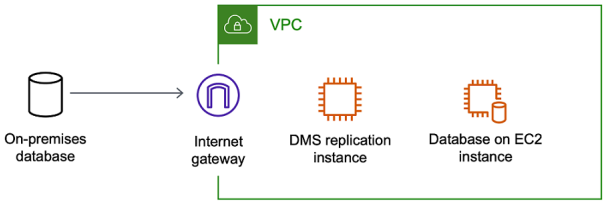
A ilustração a seguir mostra uma configuração em que o endpoint de origem é um banco de dados on-premises em um datacenter corporativo. Ele é conectado utilizando o AWS Direct Connect ou uma VPN a uma VPC que contém a instância de replicação e um banco de dados de destino em uma instância de banco de dados Amazon RDS.



Nessa configuração, o grupo de segurança da VPC deve incluir uma regra de roteamento que envia o tráfego destinado a um intervalo de CIDR de VPC ou a um endereço IP específico para um host. Esse host deve ser capaz de superar o tráfego da VPC na VPN local. Nesse caso, o host NAT inclui suas próprias configurações de grupo de segurança. Essas configurações devem permitir o tráfego do intervalo CIDR de VPC ou do endereço IP privado ou do grupo de segurança da instância de replicação para a instância do NAT. Mas não é recomendável utilizar o endereço IP privado da instância de replicação, pois isso poderá interromper a replicação se o endereço IP da replicação for alterado.

Configuração de uma rede para uma VPC utilizando a internet

Se você não usa uma VPN ou se conecta AWS Direct Connect a AWS recursos, pode usar a Internet para migrar seu banco de dados. Nesse caso, é possível migrar para uma instância do Amazon EC2 ou para uma instância de banco de dados Amazon RDS. Essa configuração envolve uma instância de replicação pública em um VPC com um gateway da Internet que contém o endpoint de destino e a instância de replicação.



Para adicionar um gateway da Internet à sua VPC, consulte Anexar um gateway da Internet, no Guia do usuário da Amazon VPC.

A tabela de rotas da VPC deve incluir regras de roteamento que enviem tráfego não destinado à VPC por padrão ao gateway da Internet. Nessa configuração, a conexão ao endpoint parecerá vir do endereço IP público da instância de replicação, não do endereço IP privado. Para obter mais informações, consulte Tabelas de rotas da VPC no Guia do usuário da Amazon VPC.

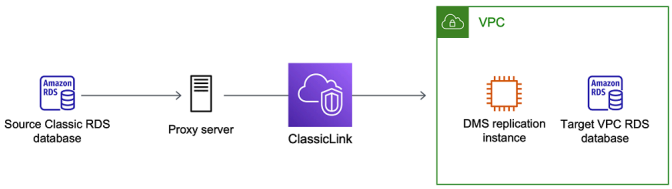
Configuração com uma instância de banco de dados RDS fora de uma VPC para uma instância de banco de dados em uma VPC usando ClassicLink

> Estamos aposentando o EC2-Classic em 15 de agosto de 2022. É recomendável migrar do EC2-Classic para uma VPC. Para ter mais informações, consulte Migrate from EC2-Classic to a VPC (Migrar do EC2-Classic para uma VPC) no Guia do usuário do Amazon EC2 e o blog EC2-Classic Networking is Retiring – Here's How to Prepare (O EC2-Classic Networking será descontinuado. Veja como se preparar).

Para conectar uma instância de banco de dados Amazon RDS que não esteja em uma VPC a um servidor de replicação do DMS e uma instância de banco de dados em uma VPC, você pode usar com um servidor proxy. ClassicLink

ClassicLink permite vincular uma instância de banco de dados EC2-Classic a uma VPC em sua conta, dentro da mesma região. AWS Após criar o link, a instância do banco de dados de origem pode se comunicar com a instância de replicação dentro da VPC utilizando os endereços IP privados.

Como a instância de replicação na VPC não pode acessar diretamente a instância de banco de dados de origem na plataforma EC2-Classic ClassicLink usando, você usa um servidor proxy. O servidor de proxy conecta instância de banco de dados de origem à VPC que contém a instância de replicação e a instância de banco de dados de destino. O servidor proxy usa ClassicLink para se conectar à VPC. O encaminhamento de portas no servidor de proxy permite a comunicação entre a instância de banco de dados de origem e a instância de banco de dados de destino na VPC.



Usando ClassicLink com o AWS Database Migration Service

Você pode conectar uma instância de banco de dados Amazon RDS que não esteja em uma VPC a um servidor de replicação do DMS e AWS uma instância de banco de dados que estejam em uma VPC. Para fazer isso, você pode usar o Amazon EC2 ClassicLink com um servidor proxy.

O procedimento a seguir mostra como usar ClassicLink para essa finalidade. Esse procedimento conecta uma instância de banco de dados de origem do Amazon RDS que não está em uma VPC a uma VPC contendo uma instância de replicação do DMS e AWS uma instância de banco de dados de destino.

- Crie uma instância de replicação do AWS DMS em uma VPC. (Todas as instâncias de replicação são criadas em VPCs.)
- Associe um grupo de segurança de VPC à instância de replicação e à instância de banco de dados de destino. Quando duas instâncias compartilham um grupo de segurança de VPC, elas podem se comunicar normalmente.
- Configure um servidor de proxy em uma instância EC2 clássica.
- Crie uma conexão usando ClassicLink entre o servidor proxy e a VPC.
- Crie endpoints AWS do DMS para os bancos de dados de origem e destino.
- Crie uma tarefa AWS DMS.

Para usar para ClassicLink migrar um banco de dados em uma instância de banco de dados que não esteja em uma VPC para um banco de dados em uma instância de banco de dados em uma VPC

1. Crie uma instância de replicação do AWS DMS e atribua um grupo de segurança da VPC:

   a. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

      Se você estiver conectado como usuário AWS Identity and Access Management (IAM), verifique se você tem as permissões apropriadas para acessar AWS DMS. Para obter mais informações sobre as permissões necessárias para a migração de banco de dados, consulte Permissões do IAM necessárias para utilizar o AWS DMS.

   b. Na página Painel, selecione Instância de Replicação. Siga as instruções da seção Etapa 1: Criar uma instância de replicação utilizando o console do AWS DMS para criar uma instância de replicação.

c. Depois de criar a instância de replicação do AWS DMS, abra o console de serviço do EC2. Escolha Interfaces de rede no painel de navegação.

d. Escolha o DMS eNetworkInterface, em seguida, escolha Alterar grupos de segurança no menu Ações.

e. Escolha o grupo de segurança que deseja utilizar para a instância de replicação e a instância de banco de dados de destino.

2. Associe o grupo de segurança da última etapa à instância do banco de dados de destino.

a. Abra o console de serviço do Amazon RDS. No painel de navegação, escolha Instâncias.

b. Escolha a instância do banco de dados de destino. Para Ações de instância, escolha Modificar.

c. Para o parâmetro Grupo de segurança, selecione o grupo de segurança utilizado na etapa anterior.

d. Escolha Continuar e escolha Modificar instância de banco de dados.

3. Etapa 3: Configurar um servidor de proxy em uma instância do EC2 Classic utilizando NGINX. Use um AMI de sua escolha para ativar uma instância EC2 clássica. O exemplo abaixo baseia-se no AMI Ubuntu Server 14.04 LTS (HVM).

Para configurar um servidor de proxy em uma instância EC2 clássica

a. Conecte-se à instância EC2 Classic e instale o NGINX utilizando os seguintes comandos:

```
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. Edite o arquivo daemon do NGINX, `/etc/init/nginx.conf`, utilizando o código a seguir:

```
# /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. Crie um arquivo de configuração do NGINX em `/usr/local/nginx/conf/nginx.conf`. No arquivo de configuração, insira o seguinte:

```
# /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. Na linha de comando, inicie o NGINX utilizando os seguintes comandos:

```
Prompt> sudo initctl reload-configuration
Prompt> sudo initctl list | grep nginx
Prompt> sudo initctl start nginx
```

4. Crie uma ClassicLink conexão entre o servidor proxy e a VPC de destino que contenha a instância de banco de dados de destino e a instância de replicação:

   a. Abra o console do EC2 e selecione a instância do EC2 Classic que está executando o servidor de proxy.

   b. Em Ações, escolha e ClassicLink, em seguida, escolha Vincular à VPC.

   c. Selecione o grupo de segurança utilizado anteriormente neste procedimento.

   d. Escolha Vincular à VPC.

5. Etapa 5: Crie endpoints do AWS DMS usando o procedimento em. Etapa 2: Especificar endpoints de origem e de destino Utilize o nome do host de proxy DNS interno do EC2 como o nome do servidor ao especificar o endpoint de origem.

6. Crie uma tarefa AWS DMS usando o procedimento emEtapa 3: Criar uma tarefa e migrar os dados.

Criar um grupo de sub-rede de replicação

Como parte da rede a ser utilizada para a migração de banco de dados, é necessário especificar as sub-redes na nuvem privada virtual (VPC) que você pretende utilizar. Essa VPC deve se basear no serviço Amazon VPC. Uma sub-rede é um intervalo de endereços IP no seu VPC em uma determinada zona de disponibilidade. Essas sub-redes podem ser distribuídas entre as zonas de disponibilidade da AWS região em que sua VPC está localizada.

Ao criar uma instância de replicação ou um perfil de instância no console do AWS DMS, você pode usar a sub-rede que você escolher.

É possível criar um grupo de sub-redes de replicação para definir as sub-redes que serão utilizadas. Selecione sub-redes em, pelo menos, duas zonas de disponibilidade.

Como criar um grupo de sub-rede de replicação

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Se estiver conectado como um usuário do IAM, verifique se você possui as permissões necessárias para acessar o AWS DMS. Para obter mais informações sobre as permissões necessárias para a migração de banco de dados, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, escolha Grupos de sub-redes.
3. Selecione Create subnet group (Criar grupo de sub-redes).
4. Na página Editar grupo de sub-redes de replicação, especifique as informações do grupo de sub-rede de replicação. A tabela a seguir descreve as configurações.

Opção	Ação
Nome	Digite um nome para o grupo de sub-redes de replicação contendo de 8 a 16 caracteres ASCII imprimíveis (excluindo /, " e @). O nome deve ser exclusivo para sua conta na AWS região que você selecionou. Você pode optar por adicionar alguma inteligência ao nome, como incluir a AWS região e a tarefa que você está executando, por exemplo**`DMS-default-VPC`** .
Descrição	Digite uma breve descrição do grupo de sub-redes de replicação.
VPC	Selecione a VPC que deseja utilizar para a migração do banco de dados. Lembre-se de que o VPC deve ter, pelo menos, uma sub-rede em, pelo menos, duas Zonas de disponibilidade.
Adicionar sub-redes	Escolha as sub-redes que deseja incluir no grupo de sub-rede de replicação. Selecione sub-redes em, pelo menos, duas zonas de disponibilidade.

5. Selecione Create subnet group (Criar grupo de sub-redes).

Resolver endpoints de domínio utilizando o DNS

Normalmente, uma instância de AWS DMS replicação usa o resolvedor do Sistema de Nomes de Domínio (DNS) em uma instância do Amazon EC2 para resolver endpoints de domínio. Se a resolução de DNS for necessária, você poderá utilizar o Amazon Route 53 Resolver. Para obter mais informações sobre como utilizar o Route 53 DNS Resolver, consulte Introdução ao Route 53 Resolver.

Para obter informações sobre como utilizar seu próprio servidor de nomes on-premises para resolver determinados endpoints utilizando o Amazon Route 53 Resolver, consulte Utilização do seu próprio servidor de nomes on-premises.

Configurar uma chave de criptografia para a instância de replicação

AWS O DMS criptografa o armazenamento usado por uma instância de replicação e as informações de conexão do endpoint. Para criptografar o armazenamento usado por uma instância de replicação, o AWS DMS usa um AWS KMS key que é exclusivo para sua conta. AWS Você pode visualizar e gerenciar essa chave KMS com AWS Key Management Service (AWS KMS). É possível utilizar a chave padrão do KMS na sua conta (`aws/dms`) ou criar uma chave personalizada do KMS. Se você tiver uma chave de AWS KMS criptografia existente, também poderá usar essa chave para criptografia.

Você pode especificar sua própria chave de criptografia fornecendo um identificador de chave KMS para criptografar seus AWS recursos do DMS. Quando você especifica a sua própria chave de criptografia, a conta de usuário usada para realizar a migração do banco de dados deve ter acesso a ela. Para obter mais informações sobre como criar suas próprias chaves de criptografia e conceder acesso a uma chave de criptografia para os usuários, consulte o Guia do desenvolvedor do AWS KMS.

Se você não especificar um identificador de chave KMS, o AWS DMS usará sua chave de criptografia padrão. O KMS cria a chave de criptografia padrão para o AWS DMS da sua AWS conta. Sua AWS conta tem uma chave de criptografia padrão diferente para cada AWS região.

Para gerenciar as chaves usadas para criptografar seus recursos AWS DMS, você usa. AWS KMS Você pode AWS KMS encontrá-lo pesquisando AWS Management Console por KMS no painel de navegação.

AWS KMS combina hardware e software seguros e de alta disponibilidade para fornecer um sistema de gerenciamento de chaves dimensionado para a nuvem. Usando AWS KMS, você pode criar

chaves de criptografia e definir as políticas que controlam como essas chaves podem ser usadas. AWS KMS suporta AWS CloudTrail, para que você possa auditar o uso das chaves para verificar se as chaves estão sendo usadas adequadamente. Suas AWS KMS chaves podem ser usadas em combinação com o AWS DMS e outros AWS serviços compatíveis. Os serviços da AWS compatíveis incluem o Amazon RDS, o Amazon S3, o Amazon Elastic Block Store (Amazon EBS) e o Amazon Redshift.

Depois de criar seus recursos do AWS DMS com uma chave de criptografia específica, você não pode alterar a chave de criptografia desses recursos. Certifique-se de determinar seus requisitos de chave de criptografia antes de criar seus recursos de AWS DMS.

Criar uma instância de replicação

A primeira tarefa na migração de um banco de dados é criar uma instância de replicação. Essa instância de replicação exige armazenamento e capacidade de processamento suficientes para executar as tarefas que você atribui para migrar os dados do banco de dados de origem para o banco de dados de destino. O tamanho necessário dessa instância varia de acordo com a quantidade de dados necessários para migrar e as tarefas que ela deve realizar. Para obter mais informações sobre as instâncias de replicação, consulte Trabalhando com uma instância de AWS DMS replicação.

Para criar uma instância de replicação usando o console AWS

1. Escolha Instâncias de replicação no painel de navegação do AWS DMS console e, em seguida, escolha Criar instância de replicação.
2. Na página Criar instância de replicação, especifique as informações da instância de replicação. A tabela a seguir descreve as configurações que podem ser feitas.

Opção	Ação
Nome	Insira um nome para a instância de replicação que contenha de 8 a 16 caracteres ASCII imprimíveis (excluindo /, " e @). O nome deve ser exclusivo para a sua conta na região da AWS selecionada. Você pode optar por adicionar alguma inteligência ao nome, como incluir a AWS região e a tarefa que você está

Opção	Ação
	executando, por exemplo**`west2-mysql2mysql-instance1`** .
Nome do recurso da Amazon (ARN) descritivo: Opcional	Um nome amigável para substituir o ARN padrão do DMS. Não é possível modificá-lo após a criação.
Descrição	Insira uma breve descrição da instância de replicação.
Instance class	Escolha uma classe de instância com a configuração necessária para a migração. Lembre-se de que a instância deve ter poder de armazenamento, de rede e de processamento suficiente para concluir a migração com êxito. Para obter mais informações sobre como determinar a melhor classe de instância para a migração, consulte Trabalhando com uma instância de AWS DMS replicação.
Versão do mecanismo	No AWS DMS console, você pode escolher qualquer versão de mecanismo compatível que desejar. A partir do AWS CLI, a instância de replicação executa a versão não beta mais recente do mecanismo de AWS DMS replicação, a menos que você especifique uma versão diferente do mecanismo no. AWS CLI
Alta disponibilidade	Use este parâmetro opcional para criar uma réplica em espera da instância de replicação em outra zona de disponibilidade para suporte a failover. Se você planejar utilizar a captura de dados de alteração (CDC) ou a replicação contínua, ative essa opção.

Opção	Ação
Armazenamento alocado (GiB)	O armazenamento é consumido principalmente por arquivos de log e transações armazenadas em cache. Para transações armazenadas em cache, o armazenamento só é usado quando elas devem ser gravadas em disco. Portanto, o AWS DMS não usa uma quantidade significativa de armazenamento. Algumas exceções incluem o seguinte: • Tabelas muito grandes que incorrem em uma carga de transação significativa. Carregar uma tabela grande pode demorar um tempo, então transaçõe s armazenadas em cache têm mais chances de ser gravadas em disco durante o carregamento de uma tabela grande. • Tarefas configuradas para pausar antes de carregar transações em cache. Nesse caso, todas as transações são armazenadas em cache até que o carregamento seja concluído para todas as tabelas. Com essa configuração, uma boa parte do armazenamento pode ser consumida por transaçõe s armazenadas em cache. • Tarefas configuradas com tabelas sendo carregada s no Amazon Redshift. Contudo, essa configuração não é um problema quando o destino é o Amazon Aurora. Na maioria dos casos, a alocação padrão de armazenamento é suficiente. No entanto, é sempre uma boa ideia prestar atenção às métricas relaciona das a armazenamento. Aumente a escala verticalm ente do armazenamento se descobrir que você está consumindo mais do que a alocação padrão.

Opção	Ação
Tipo de rede	O DMS é compatível com o tipo de rede de protocolo de endereçamento IPv4 e com os tipos de rede de protocolo de endereçamento IPv4 e IPv6 no modo de Pilha dupla. Quando houver recursos que precisam se comunicar com a instância de replicação utilizando o tipo de rede de protocolo de endereçamento IPv6, utilize o modo de Pilha dupla. Para obter informações sobre as limitações do modo de pilha dupla, consulte Limitações para instâncias de banco de dados de rede de pilha dupla no guia do usuário do Amazon Relational Database Service.
VPC	Escolha a VPC desejada. Se os bancos de dados de origem e de destino estiverem em uma VPC, escolha-os. Se os bancos de dados de origem e de destino estiverem em VPCs diferentes, verifique se ambos estão em sub-redes públicas e são publicamente acessíveis. Escolha a VPC em que a instância de replicação deve ser localizada. A instância de replicação deve poder acessar os dados no VPC de origem. Se o banco de dados de origem ou de destino não estiverem em uma VPC, selecione a VPC em que a instância de replicação deve estar.
Replication Subnet Group	Escolha o grupo de sub-rede de replicação no VPC selecionado onde deseja que instância de replicação seja criada. Se o banco de dados de origem está em um VPC, escolha o grupo de sub-rede que contém o banco de dados de origem como local da instância de replicação. Para obter mais informações sobre grupos de sub-rede de replicação, consulte Criar um grupo de sub-rede de replicação.

Opção	Ação
Publicly accessible	Escolha essa opção se desejar que a instância de replicação seja acessível pela internet. O padrão é acessível ao público e, depois que a opção é escolhida, você não pode modificá-la depois de criar a instância de replicação.

3. Selecione a guia Advanced para definir valores para configurações de rede e criptografia, se necessários. A tabela a seguir descreve as configurações.

Opção	Ação
Availability zone	Escolha a zona de disponibilidade onde o seu banco de dados de origem está.
Grupo(s) de segurança da VPC	A instância de replicação é criada em um VPC. Se o banco de dados de origem estiver em uma VPC, selecione o grupo de segurança da VPC que fornece acesso à instância de banco de dados em que o banco de dados reside.
Chave do KMS	Escolha a chave de criptografia a ser usada para criptografar o armazenamento de replicação e as informações de conexão. Se você escolher (Padrão) aws/dms, a chave padrão AWS Key Management Service (AWS KMS) associada à sua conta e AWS região será usada. Uma descrição e o número da sua conta aparecem juntamente com o ARN da chave. Para obter mais informações sobre o uso da chave de criptografia, consulte Configurando uma chave de criptografia e especificando permissões AWS KMS.

4. Especifique as configurações de Manutenção. A tabela a seguir descreve as configurações. Para obter mais informações sobre as configurações de manutenção, consulte Como trabalhar com a janela de manutenção do AWS DMS.

Opção	Ação
Atualização automática da versão	AWS DMS não diferencia entre versões principais e secundárias. Por exemplo, a atualização da versão 3.4.x para a 3.5.x não é considerada uma atualização importante, portanto, todas as alterações devem ser compatíveis com versões anteriores. Quando a Atualização automática da versão está ativada, o DMS atualizará automaticamente a versão da instância de replicação durante a janela de manutenção, se ela estiver obsoleta. Quando AutoMinorVersionUpgradeativado, o DMS usa a versão atual do mecanismo padrão quando você cria uma instância de replicação. Por exemplo, se você definir a Versão do mecanismo como um número de versão menor que a versão padrão atual, o DMS usará a versão padrão. Se AutoMinorVersionUpgradenão estiver habilitad o quando você cria uma instância de replicação, o DMS usa a versão do mecanismo especificada pelo parâmetro Engine version.
Janela de manutenção	Escolha o período semanal durante o qual pode ocorrer a manutenção do sistema, em UTC (Universal Coordinated Time). Padrão: uma janela de 30 minutos selecionada aleatoriamente a partir de um bloco de 8 horas por AWS região, ocorrendo em um dia aleatório da semana.

5. Selecione Create replication instance.

Modificar uma instância de replicação

É possível modificar as configurações de uma instância de replicação para, por exemplo, alterar a classe de instância ou para aumentar o armazenamento.

Ao modificar uma instância de replicação, é possível aplicar as alterações imediatamente. Para aplicar as alterações imediatamente, escolha a opção Aplicar imediatamente no AWS Management Console. Ou use o `--apply-immediately` parâmetro ao chamar o AWS CLI ou defina o `ApplyImmediately` parâmetro como `true` ao usar a API do DMS.

Se você não optar por aplicar as alterações imediatamente, elas serão colocadas na fila de modificações pendentes. Durante a próxima janela de manutenção, todas as alterações pendentes na fila serão aplicadas.

> **Note**
>
> Se você optar por aplicar as alterações imediatamente, todas as alterações na fila de modificações pendentes também serão aplicadas. Se qualquer uma das alterações pendentes exigir tempo de inatividade, escolher Apply changes immediately poderá causar um tempo de inatividade inesperado.

Para modificar uma instância de replicação usando o console AWS

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, selecione Replication instances.
3. Escolha a instância de replicação que você deseja modificar. A tabela a seguir descreve as modificações que podem ser feitas.

Opção	Ação
Nome	É possível alterar o nome da instância de replicaçã o. Insira um nome para a instância de replicação que contenha de 8 a 16 caracteres ASCII imprimíveis (excluindo /, " e @). O nome deve ser exclusivo para a sua conta na região da AWS selecionada. Você pode optar por adicionar alguma inteligência ao nome,

Opção	Ação
	como incluir a AWS região e a tarefa que você está executando, por exemplo**`west2-mysql2mysql-instance1`** .
Descrição	Revise ou insira uma breve descrição da instância de replicação.
Instance class	É possível alterar a classe da instância. Escolha uma classe de instância com a configuração necessária para a migração. Alterar a classe de instância faz com que a instância de replicação seja reinicializada. Essa reinicialização ocorre durante a próxima janela de manutenção ou pode ocorrer imediatamente, se você selecionar a opção Aplicar alterações imediatamente. Para obter mais informações sobre como determina r a melhor classe de instância para a migração, consulte Trabalhando com uma instância de AWS DMS replicação.
Versão do mecanismo	É possível atualizar a versão do mecanismo usada pela instância de replicação. Atualizar a versão do mecanismo de replicação faz com que a instância de replicação seja encerrada durante sua atualização.
Multi-AZ	É possível alterar essa opção para criar uma réplica de espera da instância de replicação em outra zona de disponibilidade para suporte a failover. Ou pode remover essa opção. Se você pretende utilizar a captura de dados de alteração (CDC) ou a replicação contínua, ative esta opção.

Opção	Ação
Armazenamento alocado (GiB)	O armazenamento é consumido principalmente por arquivos de log e transações armazenadas em cache. Para transações armazenadas em cache, o armazenamento só é usado quando elas devem ser gravadas em disco. Portanto, o AWS DMS não usa uma quantidade significativa de armazenamento. Algumas exceções incluem o seguinte: • Tabelas muito grandes que incorrem em uma carga de transação significativa. Carregar uma tabela grande pode demorar um tempo, então transaçõe s armazenadas em cache têm mais chances de ser gravadas em disco durante o carregamento de uma tabela grande. • Tarefas configuradas para pausar antes de carregar transações em cache. Nesse caso, todas as transações são armazenadas em cache até que o carregamento seja concluído para todas as tabelas. Com essa configuração, uma boa parte do armazenamento pode ser consumida por transaçõe s armazenadas em cache. • Tarefas configuradas com tabelas sendo carregada s no Amazon Redshift. Contudo, essa configuração não é um problema quando o destino é o Amazon Aurora. Na maioria dos casos, a alocação padrão de armazenamento é suficiente. Entretanto, convém prestar atenção em métricas relacionadas ao armazenamento e expandi-lo se achar que você está consumindo mais do que a alocação padrão.

Opção	Ação
Tipo de rede	O DMS é compatível com o tipo de rede de protocolo de endereçamento IPv4 e com os tipos de rede de protocolo de endereçamento IPv4 e IPv6 no modo de Pilha dupla. Quando você tem recursos que precisam se comunicar com a instância de replicação utilizando um tipo de rede de protocolo de endereçamento IPv6, escolha o modo Pilha dupla. Para obter informações sobre as limitações do modo de pilha dupla, consulte Limitações para instâncias de banco de dados de rede de pilha dupla no guia do usuário do Amazon Relational Database Service.
VPC Security Group(s)	A instância de replicação é criada em um VPC. Se o banco de dados de origem estiver em uma VPC, selecione o grupo de segurança da VPC que fornece acesso à instância de banco de dados em que o banco de dados reside.

Opção	Ação
Atualização automática da versão	AWS DMS não diferencia entre versões principais e secundárias. Por exemplo, a atualização da versão 3.4.x para a 3.5.x não é considerada uma atualizaç ão importante, portanto, todas as alterações devem ser compatíveis com versões anteriores. Quando a Atualização automática da versão está ativada, o DMS atualizará automaticamente a versão da instância de replicação durante a janela de manutenção, se ela estiver obsoleta. Quando a Atualização automática da versão está ativada, o DMS utiliza a versão padrão atual do mecanismo quando você cria uma instância de replicação. Por exemplo, se você definir a Versão do mecanismo como um número de versão menor que a versão padrão atual, o DMS usará a versão padrão. Se a Atualização automática da versão não estiver ativada ao criar uma instância de replicação, o DMS utilizará a versão do mecanismo especificada pelo parâmetro Versão do mecanismo.
Janela de manutenção	Escolha o período semanal durante o qual pode ocorrer a manutenção do sistema, em UTC (Universal Coordinated Time). Padrão: uma janela de 30 minutos selecionada aleatoriamente a partir de um bloco de 8 horas por AWS região, ocorrendo em um dia aleatório da semana.

Opção	Ação
Apply changes immediately	Escolha essa opção para aplicar imediatamente todas as modificações feitas. Dependendo das configura ções escolhidas, escolher essa opção pode causar uma reinicialização imediata da instância de replicaçã o. Se você escolher Testar conexão enquanto o AWS DMS aplica alterações, verá uma mensagem de erro. Depois de AWS DMS aplicar as alterações à sua instância de replicação, escolha Testar conexão novamente.
Aplique as alterações durante a próxima janela de manutenção.	Escolha essa opção se quiser que o DMS espere até a próxima janela de manutenção programada para aplicar as alterações.

Reinicializar uma instância de replicação

Você pode reinicializar uma instância de AWS DMS replicação para reiniciar o mecanismo de replicação. Uma reinicialização resulta em uma interrupção momentânea da instância de replicação, durante a qual o status da instância é definido como Rebooting (Reinicializando). Se a AWS DMS instância estiver configurada para Multi-AZ, a reinicialização poderá ser realizada com um failover. Um AWS DMS evento é criado quando a reinicialização é concluída.

Se sua AWS DMS instância for uma implantação Multi-AZ, você poderá forçar um failover planejado de uma zona de AWS disponibilidade para outra ao reinicializar. Quando você força um failover planejado da sua AWS DMS instância, AWS DMS fecha as conexões ativas na instância atual antes de mudar automaticamente para uma instância em espera em outra zona de disponibilidade. A reinicialização com um failover planejado ajuda a simular um evento de failover planejado de uma AWS DMS instância, como ao escalar a classe da instância de replicação.

> **Note**
>
> Após uma reinicialização forçar um failover de uma zona de disponibilidade para outra, a alteração da zona de disponibilidade pode não ser refletida por vários minutos. Esse atraso aparece na AWS Management Console, e nas chamadas para a AWS DMS API AWS CLI e.

Se as tarefas de migração estiverem sendo executadas na instância de replicação quando ocorrer uma reinicialização, não ocorrerá nenhuma perda de dados, e o status da tarefa será alterado para um estado de erro.

Se as tabelas na tarefa de migração estiverem no meio de um carregamento em massa (fase de carga máxima) e ainda não tiverem sido iniciadas, elas entrarão em um estado de erro. Mas as tabelas concluídas naquele momento permanecem em um estado concluído. Quando ocorre uma reinicialização durante a fase de carga máxima, é recomendável executar uma das etapas abaixo.

- Remova as tabelas que estão em um estado concluído e reinicie a tarefa com as tabelas restantes.
- Crie uma nova tarefa com as tabelas em estado de erro e com as tabelas pendentes.

Se as tabelas na tarefa de migração estiverem na fase de replicação contínua, a tarefa será retomada depois que a reinicialização for concluída.

Você não pode reinicializar sua instância AWS DMS de replicação se seu status não estiver no estado Disponível. Sua AWS DMS instância pode estar indisponível por vários motivos, como uma modificação solicitada anteriormente ou uma ação na janela de manutenção. O tempo necessário para reinicializar uma instância de AWS DMS replicação geralmente é pequeno (menos de 5 minutos).

Reinicializando uma instância de replicação usando o console AWS

Para reinicializar uma instância de replicação, use o AWS console.

Para reinicializar uma instância de replicação usando o console AWS

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, selecione Replication instances.
3. Escolha a instância de replicação que você deseja reinicializar.

4. Escolha Reboot. A caixa de diálogo Reinicializar instância de replicação é aberta.
5. Selecione a caixa de diálogo Reinicializar com failover planejado? se você tiver configurado a instância de replicação para implantação multi-AZ e desejar fazer failover para outra zona de disponibilidade da AWS .
6. Escolha Reboot.

Reinicializar uma instância de replicação utilizando a CLI

Para reinicializar uma instância de replicação, use o AWS CLI `reboot-replication-instance`comando com o seguinte parâmetro:

- `--replication-instance-arn`

Example Exemplo de reinicialização simples

O AWS CLI exemplo a seguir reinicia uma instância de replicação.

```
aws dms reboot-replication-instance \
--replication-instance-arn arn of my rep instance
```

Example Exemplo de reinicialização simples com failover

O AWS CLI exemplo a seguir reinicia uma instância de replicação com failover.

```
aws dms reboot-replication-instance \
--replication-instance-arn arn of my rep instance \
--force-planned-failover
```

Reinicializar uma instância de replicação utilizando a API

Para reinicializar uma instância de replicação, use a `RebootReplicationInstance`ação da AWS DMS API com os seguintes parâmetros:

- `ReplicationInstanceArn = arn of my rep instance`

Example Exemplo de reinicialização simples

O exemplo a seguir reinicializa uma instância de replicação.

```
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 Exemplo de reinicialização simples com failover

O exemplo de código a seguir reinicia uma instância de replicação e executa o failover em outra zona de AWS disponibilidade.

```
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
```

Excluir uma instância de replicação

Você pode excluir uma instância AWS DMS de replicação quando terminar de usá-la. Se tiver tarefas de migração que usam a instância de replicação, é necessário encerrar e excluir as tarefas antes de excluir a instância de replicação.

Se você fechar sua AWS conta, todos os AWS DMS recursos e configurações associados à sua conta serão excluídos após dois dias. Esses recursos incluem todas as instâncias de replicação, configuração de endpoint de origem e de destino, tarefas de replicação e certificados SSL. Se depois de dois dias você decidir usar AWS DMS novamente, você recria os recursos de que precisa.

Se a instância de replicação atender a todos os critérios de exclusão e permanecer no status `DELETING` por um longo período, entre em contato com o suporte para solucionar o problema.

Excluindo uma instância de replicação usando o console AWS

Para excluir uma instância de replicação, use o AWS console.

Para excluir uma instância de replicação usando o console AWS

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, selecione Replication instances.
3. Selecione a instância de replicação que deseja excluir.
4. Escolha Exluir.
5. Na caixa de diálogo, escolha Excluir.

Excluir uma instância de replicação utilizando a CLI

Para excluir uma instância de replicação, use o AWS CLI `delete-replication-instance`comando com o seguinte parâmetro:

- `--replication-instance-arn`

Example Exemplo de exclusão

O AWS CLI exemplo a seguir exclui uma instância de replicação.

```
aws dms delete-replication-instance \
--replication-instance-arn arn of my rep instance
```

Excluir uma instância de replicação utilizando a API

Para excluir uma instância de replicação, use a `DeleteReplicationInstance`ação AWS DMS da API com os seguintes parâmetros:

- `ReplicationInstanceArn = arn of my rep instance`

Example Exemplo de exclusão

O exemplo de código a seguir exclui uma instância de replicação.

```
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
```

Como trabalhar com a janela de manutenção do AWS DMS

Cada instância AWS DMS de replicação tem uma janela de manutenção semanal durante a qual todas as alterações disponíveis no sistema são aplicadas. A janela de manutenção pode ser considerada uma oportunidade de controlar quando as modificações e aplicação de patches de software ocorrem.

Se AWS DMS determinar que a manutenção é necessária durante uma determinada semana, a manutenção ocorre durante a janela de manutenção de 30 minutos que você escolheu ao criar a instância de replicação. AWS DMS conclui a maior parte da manutenção durante a janela de manutenção de 30 minutos. No entanto, pode ser necessário mais tempo para maiores alterações.

Efeito da manutenção sobre tarefas de migração existentes

Quando uma tarefa de AWS DMS migração está sendo executada em uma instância, os seguintes eventos ocorrem quando um patch é aplicado:

- Se as tabelas na tarefa de migração estiverem na fase de replicação de alterações contínua (CDC), o AWS DMS pausará a tarefa por um momento e a retomará após a aplicação do patch. A migração continua do ponto em que foi interrompida quando o patch tiver sido aplicado.
- Se AWS DMS estiver migrando uma tabela como parte de uma tarefa de migrar dados existentes ou migrar dados existentes e replicar alterações em andamento, o DMS interrompe e reinicia a migração para todas as tabelas que estão em fase de carregamento total enquanto o patch

é aplicado. O DMS também interrompe e retoma todas as tabelas que estão na fase de CDC enquanto o patch é aplicado.

Alterar a definição da janela de manutenção

Você pode alterar o período da janela de manutenção usando a AWS Management Console AWS CLI, a ou a AWS DMS API.

Alterar a definição da janela de manutenção utilizando o console

Altere o período da janela de manutenção utilizando o AWS Management Console.

Como alterar a janela de manutenção preferencial utilizando o console

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, selecione Replication instances.
3. Escolha a instância de replicação a ser modificada e escolha Modify.
4. Expanda a guia Manutenção e escolha uma data e hora para a janela de manutenção.
5. Escolha Apply changes immediately.
6. Escolha Modificar.

Alterar a configuração da janela de manutenção utilizando a CLI

Para ajustar a janela de manutenção preferida, use o AWS CLI `modify-replication-instance`comando com os parâmetros a seguir.

- `--replication-instance-identifier`
- `--preferred-maintenance-window`

Example

O AWS CLI exemplo a seguir define a janela de manutenção para terças-feiras, das 4h às 4h30. UTC.

```
aws dms modify-replication-instance \
--replication-instance-identifier myrepinstance \
```

```
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Alterar a configuração da janela de manutenção utilizando a API

Para ajustar a janela de manutenção preferida, use a ModifyReplicationInstanceação AWS DMS da API com os parâmetros a seguir.

- `ReplicationInstanceIdentifier = myrepinstance`
- `PreferredMaintenanceWindow = Tue:04:00-Tue:04:30`

Example

O exemplo de código a seguir define a janela de manutenção para as terças-feiras, das 4h às 4h30. 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
```

Como trabalhar com endpoints do AWS DMS

Um endpoint fornece a conexão, o tipo de datastore e as informações de localização do datastore da AWS. O Database Migration Service utiliza essas informações para conectar-se a um datastore e para migrar dados de um endpoint de origem para um endpoint de destino. É possível especificar atributos de conexão adicionais para um endpoint utilizando as configurações do endpoint. Esses atributos podem controlar o registro em log, o tamanho do arquivo e outros parâmetros. Para obter mais informações sobre conexões a endpoints, consulte a seção da documentação do datastore.

Veja a seguir mais detalhes sobre endpoints.

Tópicos

Criar endpoints de origem e de destino

É possível criar endpoints de origem e de destino ao criar a instância de replicação ou criar os endpoints depois que a instância de replicação for criada. Os datastores de origem e de destino podem estar em uma instância do Amazon Elastic Compute Cloud (Amazon EC2), em uma instância do Amazon Relational Database Service (Amazon RDS) ou em um banco de dados on-premises. (Observe que um dos endpoints deve estar em um serviço da AWS. Não é possível utilizar o AWS DMS para migrar de um banco de dados on-premises para outro.)

O procedimento a seguir supõe que você tenha escolhido o assistente do console do AWS DMS. Observe que também é possível realizar esta etapa selecionando Endpoints no painel de navegação do console do AWS DMS e selecionando Ciar endpoint. Ao utilizar o assistente de console, os endpoints de origem e de destino são criados na mesma página. Quando o assistente de console não é usado, cada endpoint é criado separadamente.

Como especificar endpoints do banco de dados de origem ou de destino utilizando o console da AWS

1. Na página Conectar endpoints dos bancos de dados de origem e de destino, especifique as suas informações de conexão do banco de dados de origem ou de destino. A tabela a seguir descreve as configurações.

Para esta opção	Faça o seguinte
Endpoint type	Determine se esse endpoint é o endpoint de origem ou de destino.
Selecionar instância de banco de dados RDS	Escolha essa opção se o endpoint for uma instância de banco de dados Amazon RDS.
Endpoint identifier	Digite o nome que você deseja utilizar para identific ar o endpoint. Talvez você queira incluir o tipo de endpoint no nome, como **`oracle-source`** ou **`PostgreSQL-target`** . O nome deve ser exclusivo para todas as instâncias de replicação.
Mecanismo de origem e Mecanismo de destino	Escolha o tipo de mecanismo de banco de dados que é o endpoint.
Acesso ao banco de dados do endpoint	Escolha a opção desejada para especificar as credenciais do banco de dados do endpoint: • Escolher AWS Secrets Manager: utilize os segredos definidos no AWS Secrets Manager para fornecer as credenciais de forma secreta, conforme mostrado a seguir. Para obter mais informações sobre a criação desses segredos e os perfis de acesso secreto que permitem que o AWS DMS os acesse, consulte Utilizar segredos para acessar endpoints do AWS Database Migration Service. • Fornecer informações de acesso manualmen te: utilize credenciais de texto não criptografado inseridas diretamente, conforme mostrado a seguir.

Para esta opção	Faça o seguinte
Escolher AWS Secrets Manager	Defina as seguintes credenciais secretas.
ID do segredo	Digite o nome do recurso da Amazon (ARN) completo, o ARN parcial ou o nome amigável de um segredo criado no AWS Secrets Manager para acesso ao banco de dados do endpoint.
Perfil do IAM	Digite o ARN de um perfil de acesso secreto criado no IAM para fornecer acesso ao AWS DMS em seu nome ao segredo identificado pelo ID do segredo. Para obter informações sobre como criar um perfil de acesso, consulte Utilizar segredos para acessar endpoints do AWS Database Migration Service.
ID do segredo para o gerenciamento automático de armazenamento (ASM) do Oracle	(Para endpoints de origem do Oracle que usam somente o Oracle ASM) Digite o nome do recurso da Amazon (ARN) completo, o ARN parcial ou o nome amigável de um segredo criado no AWS Secrets Manager para acesso ao Oracle ASM. Esse segredo geralmente é criado para acessar o Oracle ASM no mesmo servidor do segredo identificado pelo ID do segredo.
Perfil do IAM do Oracle ASM	(Para endpoints de origem do Oracle que usam somente o Oracle ASM) Digite o ARN de um perfil de acesso secreto criado no IAM para fornecer acesso ao AWS DMS em seu nome ao segredo identificado pelo ID do segredo do gerenciamento automático de armazenamento (ASM) do Oracle.
Fornecer informações de acesso manualmente	Defina as seguintes credenciais de texto não criptografado.

Para esta opção	Faça o seguinte
Nome do servidor	Digite o nome do servidor. Para um banco de dados on-premises, esse pode ser o endereço IP ou o nome do host público. Para uma instância de banco de dados Amazon RDS, esse pode ser o endpoint (também chamado de nome do DNS) da instância de banco de dados, como **`mysqlsrvinst.abcd12345678.us-west-2.rds.amazonaws.com`** .
Porta	Digite a porta usada pelo banco de dados.
Modo Secure Socket Layer (SSL)	Escolha um modo SSL se quiser ativar a criptografia de conexão para este endpoint. Dependendo do modo selecionado, talvez você seja solicitado a fornecer informações de certificado e de certificado do servidor.
Nome do usuário	Digite o nome do usuário com as permissões necessárias para permitir a migração de dados. Para obter informações sobre as permissões necessári as, consulte a seção de segurança do mecanismo de banco de dados de origem ou de destino neste guia do usuário.
Senha	Digite a senha da conta com as permissões necessári as. As senhas de endpoints de origem e de destino do AWS DMS têm restrições de caracteres, dependend o do mecanismo do banco de dados. Para obter mais informações, consulte a tabela a seguir.
Database name	Para determinados mecanismos de banco de dados, o nome do banco de dados que você quer utilizar como o banco de dados de endpoint.

A tabela a seguir lista os caracteres não suportados nas senhas dos endpoints e nos segredos do gerenciador secreto dos mecanismos de banco de dados listados. Para utilizar vírgulas (,) em senhas de endpoint, utilize o suporte do Secrets Manager fornecido no AWS DMS para

autenticar o acesso às instâncias do AWS DMS. Para ter mais informações, consulte Utilizar segredos para acessar endpoints do AWS Database Migration Service.

Para esse mecanismo de banco de dados	Os caracteres a seguir não são suportados em uma senha de endpoint e segredos de gerenciador secreto
Todos	`{ }`
Microsoft Azure, somente como origem	`;`
Microsoft SQL Server	`, ;`
Compatível com MySQL, incluindo MySQL, MariaDB e Amazon Aurora MySQL	`;`
Oracle	`,`
PostgreSQL, Amazon Aurora Edição compatível com PostgreSQL e Amazon Aurora Sem Servidor como destino da Edição compatível com o Aurora PostgreSQL	`; + %`
Amazon Redshift, somente como destino	`, ;`

2. Escolha Configurações de Endpoint e AWS KMS key se for necessário. Para testar a conexão do endpoint, selecione Executar teste. A tabela a seguir descreve as configurações.

Para esta opção	Faça o seguinte
Configurações de endpoint	Selecione os parâmetros de conexão adicional aqui. Para obter mais informações sobre as configurações de endpoint, consulte a seção Mecanismo de origem ou Mecanismo de destino da documentação (especifi cada na etapa 1).

Para esta opção	Faça o seguinte
	Para um endpoint de origem do Oracle que utiliza o Oracle ASM, se você escolher Fornecer informações de acesso manualmente na etapa 1, talvez também precise digitar a configuração do endpoint para especificar as credenciais do usuário do Oracle ASM. Para obter mais informações sobre essas configura ções de endpoint do Oracle ASM, consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.
AWS KMS key	Escolha a chave de criptografia a ser usada para criptografar o armazenamento de replicação e as informações de conexão. Se você escolher (Padrão) aws/dms, a chave padrão do AWS Key Managemen t Service (AWS KMS) associada à sua conta e região da AWS será usada. Para obter mais informaçõ es sobre o uso da chave de criptografia, consulte Configurando uma chave de criptografia e especific ando permissões AWS KMS.
Testar a conexão do endpoint (opcional)	Adicione a VPC e o nome da instância de replicação. Para testar a conexão, selecione Executar o teste.

Origens para a migração de dados

O AWS Database Migration Service (AWS DMS) utiliza vários dos mecanismos de dados mais populares como origem para a replicação de dados. A origem do banco de dados pode ser um mecanismo autogerenciado em execução em uma instância do Amazon EC2 ou em um banco de dados on-premises. Ou pode ser uma fonte de dados em um serviço da AWS, como o Amazon RDS ou o Amazon S3.

Para obter uma lista abrangente de origens válidas, consulte Origens do AWS DMS.

Tópicos

- Utilizar um banco de dados Oracle como origem do AWS DMS

- Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS
- Utilizar um banco de dados Microsoft Azure SQL como a origem do AWS DMS
- Utilizar a instância gerenciada do Microsoft Azure SQL como origem do AWS DMS
- Utiliza o Microsoft Azure Database para PostgreSQL como origem do AWS DMS
- Utilizar um servidor flexível do Microsoft Azure Database para MySQL Server como origem do AWS DMS
- Utilizar o OCI MySQL Heatwave como origem do AWS DMS
- Utilizar o Google Cloud para MySQL como a origem do AWS DMS
- Utilizar o Google Cloud para PostgreSQL como origem do AWS DMS
- Utilizar o banco de dados PostgreSQL como origem do AWS DMS
- Utilizar um banco de dados compatível com MySQL como origem do AWS DMS
- Utilizar um banco de dados SAP ASE como origem do AWS DMS
- Utilizar o MongoDB como origem do AWS DMS
- Usando o Amazon DocumentDB (com compatibilidade com o MongoDB) como fonte para AWS DMS
- Usando o Amazon S3 como fonte para AWS DMS
- Usando o banco de dados IBM Db2 para Linux, Unix, Windows e Amazon RDS (Db2 LUW) como fonte para AWS DMS
- Utilizar o bancos de dados IBM Db2 for z/OS como origem do AWS DMS

Utilizar um banco de dados Oracle como origem do AWS DMS

Você pode migrar dados de um ou vários bancos de dados Oracle usando o. AWS DMS Com um banco de dados Oracle como origem, é possível migrar dados para qualquer um dos destinos compatíveis com o AWS DMS.

AWS DMS suporta as seguintes edições do banco de dados Oracle:

- Oracle Enterprise Edition
- Oracle Standard Edition
- Oracle Express Edition
- Oracle Personal Edition

Para obter informações sobre versões de bancos de dados Oracle que oferecem AWS DMS suporte como fonte, consulteFontes para AWS DMS.

É possível utilizar Secure Sockets Layer (SSL) para criptografar conexões entre o endpoint do Oracle e a instância de replicação. Para obter mais informações sobre a utilização de SSL com um endpoint do Oracle, consulte Suporte de SSL para um endpoint do Oracle.

AWS DMS suporta o uso da criptografia transparente de dados (TDE) da Oracle para criptografar dados em repouso no banco de dados de origem. Para obter mais informações sobre como utilizar a TDE do Oracle com um endpoint do Oracle de origem, consulte Métodos de criptografia suportados para usar o Oracle como fonte para AWS DMS.

AWS suporta o uso do TLS versão 1.2 e posterior com endpoints Oracle (e todos os outros tipos de endpoints) e recomenda o uso do TLS versão 1.3 ou posterior.

Siga estas etapas para configurar um banco de dados Oracle como um endpoint AWS DMS de origem:

1. Crie um usuário Oracle com as permissões apropriadas AWS DMS para acessar seu banco de dados de origem Oracle.
2. Crie um endpoint de origem do Oracle que esteja em conformidade com a configuração de banco de dados Oracle escolhida. Para criar uma full-load-only tarefa, nenhuma configuração adicional é necessária.
3. Para criar uma tarefa que gerencie a captura de dados de alteração (uma tarefa somente de CDC ou de carga completa e CDC), escolha Oracle LogMiner ou AWS DMS Binary Reader para capturar as alterações nos dados. A escolha LogMiner do Binary Reader determina algumas das permissões e opções de configuração posteriores. Para uma comparação entre o Binary Reader LogMiner e o Binary Reader, consulte a seção a seguir.

> **Note**
>
> Para obter mais informações sobre tarefas de carga máxima, tarefas somente CDC e tarefas de carga máxima e CDC, consulte Criar uma tarefa

Para obter detalhes adicionais sobre como trabalhar com bancos de dados de origem Oracle AWS DMS, consulte as seções a seguir.

Tópicos

Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC

Em AWS DMS, há dois métodos para ler os redo logs ao fazer a captura de dados de alteração (CDC) para Oracle como fonte: Oracle LogMiner e AWS DMS Binary Reader. LogMiner é uma API da Oracle para ler os redo logs on-line e os arquivos de redo log arquivados. O Binary Reader é um AWS DMS método que lê e analisa diretamente os arquivos brutos de redo log. Esses métodos têm os recursos a seguir.

Atributo	LogMiner	Binary Reader
Fácil de configurar	Sim	Não
Menor impacto na E/S e na CPU do sistema de origem	Não	Sim
Melhor performance da CDC	Não	Sim
Compatível com clusters de tabelas do Oracle	Sim	Não
Compatível com todos os tipos de Oracle Hybrid Columnar Compression (HCC)	Sim	Parcialmente

Atributo	LogMiner	Binary Reader
		O Binary Reader não é compatível com QUERY LOW para tarefas com CDC. Todos os outros tipos de HCC são totalmente compatíveis.
Compatível com a coluna LOB no Oracle 12c somente	Não (o LOB Support não está disponível LogMiner no Oracle 12c.)	Sim
Compatível com instruções `UPDATE` que afetam somente colunas LOB	Não	Sim
Compatível com a criptografia de dados transparente (TDE) do Oracle	Parcialmente Ao usar o Oracle LogMiner, AWS DMS não oferece suporte à criptografia TDE em nível de coluna para Amazon RDS for Oracle.	Parcialmente O Binary Reader é compatível com TDE somente em bancos de dados Oracle autogerenciados.
Compatível com todos os métodos de compactação do Oracle	Sim	Não
Compatível com transações XA	Não	Sim

Atributo	LogMiner	Binary Reader
RAC	Sim Não recomendado por motivos de desempenho e por algumas limitações internas do DMS.	Sim Altamente recomendado

> **Note**
>
> Por padrão, AWS DMS usa Oracle LogMiner for (CDC).
> AWS DMS suporta métodos transparentes de criptografia de dados (TDE) ao trabalhar com um banco de dados de origem Oracle. Se as credenciais do TDE que você especificar estiverem incorretas, a tarefa de AWS DMS migração não falhará, o que pode afetar a replicação contínua de tabelas criptografadas. Para obter mais informações sobre como especificar as credenciais da TDE, consulte Métodos de criptografia suportados para usar o Oracle como fonte para AWS DMS.

As principais vantagens de usar LogMiner com AWS DMS incluem o seguinte:

- LogMiner suporta a maioria das opções do Oracle, como opções de criptografia e opções de compactação. O Binary Reader não é compatível com todas as opções do Oracle, especialmente a compactação e a maioria das opções de criptografia.
- LogMiner oferece uma configuração mais simples, especialmente em comparação com a configuração de acesso direto do Binary Reader ou quando os redo logs são gerenciados usando o Oracle Automatic Storage Management (ASM).
- LogMiner suporta clusters de tabelas para uso por AWS DMS. O Binary Reader não oferece.

As principais vantagens de usar o Binary Reader com AWS DMS incluem o seguinte:

- Para migrações com um alto volume de alterações, LogMiner pode ter algum impacto de E/S ou CPU no computador que hospeda o banco de dados de origem Oracle. O Binary Reader tem menos chance de causar impacto na E/S ou na CPU porque os logs são minerados diretamente, em vez da execução de várias consultas ao banco de dados.
- Para migrações com um alto volume de alterações, o desempenho do CDC geralmente é muito melhor ao usar o Binary Reader em comparação ao uso do Oracle. LogMiner
- O Binary Reader suporta CDC para LOBs na versão 12c do Oracle. LogMinernão.

Em geral, use o Oracle LogMiner para migrar seu banco de dados Oracle, a menos que você tenha uma das seguintes situações:

- Você precisa executar várias tarefas de migração no banco de dados Oracle de origem.
- O volume de alterações ou o volume de redo logs no banco de dados Oracle de origem é alto ou você tem alterações e também está utilizando o Oracle ASM.

> **Note**
>
> Se você alternar entre o uso do Oracle LogMiner e do AWS DMS Binary Reader, certifique-se de reiniciar a tarefa do CDC.

Configuração da CDC em um banco de dados Oracle de origem

Para que um endpoint de origem Oracle se conecte ao banco de dados para uma tarefa de captura de dados de alteração (CDC), talvez seja necessário especificar atributos de conexão adicionais. Isso pode ser verdade tanto para uma tarefa de carga máxima e CDC quanto para uma tarefa somente de CDC. Os atributos extras de conexão que você especifica dependem do método usado para acessar os redo logs: Oracle LogMiner ou AWS DMS Binary Reader.

Você especifica atributos de conexão adicionais ao criar o endpoint de origem. Se você tiver várias configurações de atributos de conexão, separe-as umas das outras por ponto e vírgula e sem espaços em branco adicionais (por exemplo, `oneSetting;thenAnother`).

AWS DMS usa LogMiner por padrão. Não é necessário especificar atributos de conexão adicionais para utilizá-lo.

Para utilizar o Binary Reader para acessar os redo logs, inclua os seguintes atributos de conexão adicionais.

```
useLogMinerReader=N;useBfile=Y;
```

Utilize o seguinte formato para os atributos de conexão adicionais para acessar um servidor que utiliza o ASM com o Binary Reader.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_numb
+ASM;
```

Defina o parâmetro de solicitação `Password` do endpoint de origem para a senha do usuário do Oracle e a senha do ASM, separadas por uma vírgula da seguinte forma.

```
oracle_user_password,asm_user_password
```

Quando a origem Oracle utiliza o ASM, é possível trabalhar com opções de alto desempenho no Binary Reader para o processamento de transações em escala. Essas opções incluem atributos de conexão adicionais para especificar o número de threads paralelos (`parallelASMReadThreads`) e o número de buffers de leitura antecipada (`readAheadBlocks`). A configuração conjunta desses atributos pode melhorar significativamente o desempenho da tarefa de CDC. As configurações a seguir fornecem bons resultados para a maioria das configurações do ASM.

```
useLogMinerReader=N;useBfile=Y;asm_user=asm_username;asm_server=RAC_server_ip_address:port_numb
+ASM;
    parallelASMReadThreads=6;readAheadBlocks=150000;
```

Para obter mais informações sobre os valores compatíveis com atributos de conexão adicionais, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.

Além disso, o desempenho de uma tarefa de CDC com uma origem Oracle que utiliza o ASM depende das outras configurações escolhidas. Essas configurações incluem os atributos de conexão adicionais do AWS DMS e as configurações do SQL para configurar a origem Oracle. Para obter mais informações sobre atributos de conexão adicionais para uma origem Oracle que utiliza o ASM, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS

Você também precisa escolher um ponto de partida apropriado da CDC. Normalmente, ao fazer isso, você deseja identificar o ponto de processamento da transação que captura a primeira transação

aberta a partir da qual a CDC é iniciada. Caso contrário, a tarefa de CDC pode perder transações abertas anteriores. Para um banco de dados de origem Oracle, é possível escolher um ponto inicial nativo da CDC com base no número da alteração do sistema (SCN) do Oracle para identificar essa primeira transação aberta. Para ter mais informações, consulte Executar a replicação a partir de um ponto de início de CDC.

Para obter mais informações sobre como configurar a CDC para um banco de dados Oracle autogerenciado como origem, consulte Privilégios de conta necessários ao usar o Oracle LogMiner para acessar os redo logs, Privilégios de conta necessários ao usar o AWS DMS Binary Reader para acessar os redo logs e Privilégios adicionais de conta necessários ao utilizar o Binary Reader com o Oracle ASM.

Para obter mais informações sobre como configurar o CDC para um banco AWS de dados Oracle gerenciado como fonte, consulte e. Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS Utilizar um Amazon RDS Oracle Standby (réplica de leitura) como origem com o Binary Reader para CDC no AWS DMS

Fluxos de trabalho para configurar um banco de dados de origem Oracle autogerenciado ou AWS gerenciado para AWS DMS

Fluxos de trabalho para configurar um banco de dados de origem Oracle autogerenciado ou AWS gerenciado para AWS DMS

Para configurar uma instância de banco de dados de origem autogerenciado, utilize as seguintes etapas, dependendo de como você executa a CDC.

Para esta etapa do fluxo de trabalho	Se você executar o CDC usando LogMiner, faça isso	Se você executar a CDC utilizando o Binary Reader, faça isso
Conceda privilégios à conta Oracle.	Consulte Privilégios de conta de usuário necessários em uma fonte Oracle autogeren ciada para AWS DMS.	Consulte Privilégios de conta de usuário necessários em uma fonte Oracle autogeren ciada para AWS DMS.
Prepare o banco de dados de origem para replicação utilizando a CDC.	Consulte Preparando um banco de dados de origem	Consulte Preparando um banco de dados de origem

Para esta etapa do fluxo de trabalho	Se você executar o CDC usando LogMiner, faça isso	Se você executar a CDC utilizando o Binary Reader, faça isso
	autogerenciado Oracle para CDC usando AWS DMS.	autogerenciado Oracle para CDC usando AWS DMS.
Conceda privilégios adicionais ao usuário do Oracle necessários para a CDC.	Consulte Privilégios de conta necessários ao usar o Oracle LogMiner para acessar os redo logs.	Consulte Privilégios de conta necessários ao usar o AWS DMS Binary Reader para acessar os redo logs.
Para uma instância Oracle com ASM, conceda privilégios adicionais de conta de usuário necessários para acessar o ASM para CDC.	Nenhuma ação adicional. AWS DMS oferece suporte ao Oracle ASM sem privilégios adicionais de conta.	Consulte Privilégios adicionais de conta necessários ao utilizar o Binary Reader com o Oracle ASM.
Se você ainda não tiver feito isso, configure a tarefa para usar LogMiner o Binary Reader for CDC.	Consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.	Consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.
Configure o Oracle Standby como origem para a CDC.	AWS DMS não oferece suporte ao Oracle Standby como fonte.	Consulte Utilizar um Oracle Standby autogerenciado como origem com o Binary Reader para CDC no AWS DMS.

Use as seguintes etapas do fluxo de trabalho para configurar uma instância de banco AWS de dados de origem Oracle gerenciada.

Para esta etapa do fluxo de trabalho	Se você executar o CDC usando LogMiner, faça isso	Se você executar a CDC utilizando o Binary Reader, faça isso
Conceda privilégios à conta Oracle.	Para ter mais informaçõ es, consulte Privilégios de	Para ter mais informaçõ es, consulte Privilégios de

Para esta etapa do fluxo de trabalho	Se você executar o CDC usando LogMiner, faça isso	Se você executar a CDC utilizando o Binary Reader, faça isso
	conta de usuário necessári os em uma fonte Oracle AWS gerenciada para AWS DMS.	conta de usuário necessári os em uma fonte Oracle AWS gerenciada para AWS DMS.
Prepare o banco de dados de origem para replicação utilizando a CDC.	Para ter mais informações, consulte Configurando uma fonte Oracle AWS gerenciada para AWS DMS.	Para ter mais informações, consulte Configurando uma fonte Oracle AWS gerenciada para AWS DMS.
Conceda privilégios adicionai s ao usuário do Oracle necessários para a CDC.	Nenhum privilégio adicional de conta é necessário.	Para ter mais informações, consulte Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS.
Se você ainda não tiver feito isso, configure a tarefa para usar LogMiner o Binary Reader for CDC.	Para ter mais informações, consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.	Para ter mais informações, consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.
Configure o Oracle Standby como origem para a CDC.	AWS DMS não oferece suporte ao Oracle Standby como fonte.	Para ter mais informações, consulte Utilizar um Amazon RDS Oracle Standby (réplica de leitura) como origem com o Binary Reader para CDC no AWS DMS.

Trabalhando com um banco de dados Oracle autogerenciado como fonte para AWS DMS

Um banco de dados autogerenciado é um banco de dados que você configura e controla, em uma instância de banco de dados on-premises ou em um banco de dados no Amazon EC2. A seguir,

você pode descobrir os privilégios e as configurações de que precisa ao usar um banco de dados Oracle autogerenciado com o. AWS DMS

Privilégios de conta de usuário necessários em uma fonte Oracle autogerenciada para AWS DMS

Para usar um banco de dados Oracle como origem em AWS DMS, conceda os seguintes privilégios ao usuário Oracle especificado nas configurações de conexão do endpoint Oracle.

> **Note**
>
> Ao conceder privilégios, utilize o nome real dos objetos, não o sinônimo de cada objeto. Por exemplo, utilize `V_$OBJECT` incluindo o sublinhado, não `V$OBJECT` sem o sublinhado.

```
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 Métodos de
 criptografia suportados para usar o Oracle como fonte para 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.
```

Conceda o privilégio adicional a seguir para cada tabela replicada ao utilizar uma lista de tabelas específica.

```
GRANT SELECT on any-replicated-table to db_user;
```

Conceda o seguinte privilégio adicional para validar colunas LOB com o recurso de validação.

```
GRANT EXECUTE ON SYS.DBMS_CRYPTO TO db_user;
```

Conceda o seguinte privilégio adicional se você usar o leitor binário em vez de LogMiner.

```
GRANT SELECT ON SYS.DBA_DIRECTORIES TO db_user;
```

Conceda o seguinte privilégio adicional para expor visualizações.

```
GRANT SELECT on ALL_VIEWS to dms_user;
```

Para expor visualizações, adicione também o atributo de conexão adicional do `exposeViews=true` ao endpoint de origem.

Conceda o seguinte privilégio adicional ao utilizar replicações com tecnologia sem servidor.

```
GRANT SELECT on dba_segments to db_user;
```

Para obter informações sobre as replicações com tecnologia sem servidor, consulte Trabalhando com AWS DMS Serverless.

Conceda os seguintes privilégios adicionais ao utilizar avaliações de pré-migração específicas do Oracle.

```
GRANT SELECT on gv_$parameter  to dms_user;
GRANT SELECT on v_$instance to dms_user;
GRANT SELECT on v_$version to dms_user;
GRANT SELECT on gv_$ASM_DISKGROUP to dms_user;
GRANT SELECT on gv_$database to dms_user;
GRANT SELECT on dba_db_links to dms_user;
GRANT SELECT on gv_$log_History to dms_user;
GRANT SELECT on gv_$log to dms_user;
GRANT SELECT ON DBA_TYPES TO db_user;
GRANT SELECT ON DBA_USERS to dms_user;
GRANT SELECT ON DBA_DIRECTORIES to dms_user;
```

Para obter informações sobre avaliações de pré-migração específicas do Oracle, consulte. Avaliações da Oracle

Pré-requisitos para tratar transações abertas do Oracle Standby

Ao usar AWS DMS as versões 3.4.6 e superiores, execute as etapas a seguir para lidar com transações abertas para o Oracle Standby.

1. Crie um link de banco de dados nomeado AWSDMS_DBLINK no banco de dados primário. O *DMS_USER* utilizará o link de banco de dados para conectar-se ao banco de dados primário. Observe que o link de banco de dados é executado na instância em espera para consultar as transações abertas em execução no banco de dados primário. Veja o exemplo a seguir.

   ```
   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. Verifique se a conexão ao link de banco de dados que utiliza o *DMS_USER* está estabelecida conforme mostrado no exemplo a seguir.

```
select 1 from dual@AWSDMS_DBLINK
```

Preparando um banco de dados de origem autogerenciado Oracle para CDC usando AWS DMS

Prepare seu banco de dados Oracle autogerenciado como origem para executar uma tarefa de CDC fazendo o seguinte:

- Verificando se é AWS DMS compatível com a versão do banco de dados de origem.
- Verificar se o modo ARCHIVELOG está ativado.
- Configuração de registro em log suplementar.

Verificando se é AWS DMS compatível com a versão do banco de dados de origem

Execute uma consulta, como a seguinte, para verificar se a versão atual do banco de dados Oracle é compatível com o AWS DMS.

```
SELECT name, value, description FROM v$parameter WHERE name = 'compatible';
```

Aqui, `name`, `value` e `description` são colunas em algum local no banco de dados que estão sendo consultadas com base no valor de `name`. Se essa consulta for executada sem erros, AWS DMS será compatível com a versão atual do banco de dados e você poderá continuar com a migração. Se a consulta gerar um erro, AWS DMS isso não é compatível com a versão atual do banco de dados. Para continuar com a migração, primeiro converta o banco de dados Oracle em uma versão suportada pelo AWS DMS.

Verificar se o modo ARCHIVELOG está ativado

É possível executar o Oracle em dois modos diferentes: o modo `ARCHIVELOG` e o modo `NOARCHIVELOG`. Para executar uma tarefa de CDC, execute o banco de dados no modo `ARCHIVELOG`. Para saber se o banco de dados está no modo `ARCHIVELOG`, execute a consulta a seguir.

```
SQL> SELECT log_mode FROM v$database;
```

Se for retornado o modo `NOARCHIVELOG`, defina o banco de dados como `ARCHIVELOG` de acordo com as instruções do Oracle.

Configuração de registro em log suplementar

Para capturar mudanças contínuas, é AWS DMS necessário que você habilite o mínimo de registro suplementar em seu banco de dados de origem Oracle. Além disso, você precisa ativar o registro em log suplementar em cada tabela replicada no banco de dados.

Por padrão, AWS DMS adiciona registro `PRIMARY KEY` suplementar em todas as tabelas replicadas. Para permitir AWS DMS a adição de registros `PRIMARY KEY` complementares, conceda o seguinte privilégio para cada tabela replicada.

```
ALTER on any-replicated-table;
```

Você pode desativar o registro `PRIMARY KEY` suplementar padrão adicionado AWS DMS usando o atributo de conexão extra. `addSupplementalLogging` Para ter mais informações, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.

Ative o registro em log suplementar se a tarefa de replicação atualizar uma tabela utilizando uma cláusula `WHERE` que não faz referência a uma coluna de chave primária.

Como configurar o registro em log suplementar manualmente

1. Execute a consulta a seguir para verificar se o registro em log suplementar está ativado para o banco de dados.

   ```
   SELECT supplemental_log_data_min FROM v$database;
   ```

   Se o resultado retornado for `YES` ou `IMPLICIT`, o registro em log suplementar estará ativado para o banco de dados.

   Se necessário, ative o registro em log suplementar para o banco de dados executando o comando a seguir.

   ```
   ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
   ```

2. Verifique se o registro em log suplementar necessário está adicionado a cada tabela replicada.

   Considere o seguinte:

   - Se o registro em log suplementar `ALL COLUMNS` estiver adicionado à tabela, não será necessário adicionar mais registros em log.

- Se houver uma chave primária, adicione o registro em log suplementar à chave primária. É possível fazer isso utilizando o formato para adicionar registro em log suplementar à chave primária ou adicionando o registro em log suplementar às colunas de chave primária no banco de dados.

```
ALTER TABLE Tablename ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
```

- Se não houver uma chave primária e a tabela possuir um único índice exclusivo, adicione todas as colunas do índice exclusivo ao log suplementar.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG GROUP LogGroupName
 (UniqueIndexColumn1[, UniqueIndexColumn2] ...) ALWAYS;
```

  Usar `SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS` não faz com que as colunas do índice exclusivo sejam adicionadas ao log.

- Se não existir uma chave primária e a tabela tiver vários índices exclusivos, AWS DMS seleciona o primeiro índice exclusivo em uma lista ascendente ordenada alfabeticamente. Você precisa adicionar registro em log suplementar nas colunas do índice selecionado, como no item anterior.

  Usar `SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS` não faz com que as colunas do índice exclusivo sejam adicionadas ao log.

- Se não houver uma chave primária e não houver um índice exclusivo, adicione o registro em log suplementar em todas as colunas.

```
ALTER TABLE TableName ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

  Em alguns casos, a chave primária da tabela de destino ou índice exclusivo são diferentes da chave primária da tabela de origem ou índice exclusivo. Nesses casos, adicione o registro em log suplementar manualmente nas colunas da tabela de origem que compõem a chave primária da tabela ou o índice exclusivo de destino.

  Além disso, se você alterar a chave primária da tabela de destino, adicione o registro complementar às colunas do índice exclusivo de destino, em vez das colunas do índice exclusivo ou da chave primária de origem.

Se um filtro ou uma transformação for definida para uma tabela, talvez seja necessário habilitar o registro em log adicional.

Considere o seguinte:

- Se o registro em log suplementar `ALL COLUMNS` estiver adicionado à tabela, não será necessário adicionar mais registros em log.
- Se a tabela tiver um índice exclusivo ou uma chave primária, adicione registro em log suplementar a cada coluna envolvida em um filtro ou transformação. No entanto, faça isso somente se essas colunas forem diferentes da chave primária ou das colunas do índice exclusivo.
- Se uma transformação incluir apenas uma coluna, não adicione essa coluna a um grupo de registro em log suplementar. Por exemplo, para uma transformação `A+B`, adicione registro em log suplementar em ambas as colunas `A` e `B`. No entanto, para uma transformação `substring(A,10)`, não adicione registro em log suplementar à coluna `A`.
- Para configurar o registro em log suplementar em colunas de índice exclusivo ou de chave primária e outras colunas específicas filtradas ou transformadas, é possível adicionar o registro em log suplementar `USER_LOG_GROUP`. Adicione esse registro em log suplementar às colunas de chave primária ou ao índice exclusivo e às outras colunas específicas filtradas ou transformadas.

  Por exemplo, para replicar uma tabela nomeada `TEST.LOGGING` com chave primária `ID` e um filtro pela coluna `NAME`, é possível executar um comando semelhante ao seguinte para criar o registro em log suplementar do grupo de logs.

```
ALTER TABLE TEST.LOGGING ADD SUPPLEMENTAL LOG GROUP TEST_LOG_GROUP (ID, NAME) ALWAYS;
```

Privilégios de conta necessários ao usar o Oracle LogMiner para acessar os redo logs

Para acessar os redo logs usando o Oracle LogMiner, conceda os seguintes privilégios ao usuário Oracle especificado nas configurações de conexão do endpoint Oracle.

```
GRANT EXECUTE on DBMS_LOGMNR to db_user;
GRANT SELECT on V_$LOGMNR_LOGS to db_user;
GRANT SELECT on V_$LOGMNR_CONTENTS to db_user;
GRANT LOGMINING to db_user; -- Required only if the Oracle version is 12c or higher.
```

Privilégios de conta necessários ao usar o AWS DMS Binary Reader para acessar os redo logs

Para acessar os redo logs usando o AWS DMS Binary Reader, conceda os seguintes privilégios ao usuário Oracle especificado nas configurações de conexão do endpoint Oracle.

```
GRANT SELECT on v_$transportable_platform to db_user;   -- Grant this privilege if the
 redo logs are stored in Oracle Automatic Storage Management (ASM) and AWS DMS accesses
 them from ASM.
GRANT CREATE ANY DIRECTORY to db_user;                  -- Grant this privilege to
 allow AWS DMS to use Oracle BFILE read file access in certain cases. This access is
 required when the replication instance doesn't have file-level access to the redo logs
 and the redo logs are on non-ASM storage.
GRANT EXECUTE on DBMS_FILE_TRANSFER to db_user;         -- Grant this privilege to copy
 the redo log files to a temporary folder using the CopyToTempFolder method.
GRANT EXECUTE on DBMS_FILE_GROUP to db_user;
```

O Binary Reader funciona com recursos de arquivo Oracle que incluem diretórios do Oracle. Cada objeto no diretório do Oracle inclui o nome da pasta que contém os arquivos de logs redo a serem processados. Esses diretórios do Oracle não são representados no nível do sistema de arquivos. Eles são diretórios lógicos criados no nível do banco de dados Oracle. É possível exibi-los na visualização `ALL_DIRECTORIES` do Oracle.

Se você quiser AWS DMS criar esses diretórios Oracle, conceda o `CREATE ANY DIRECTORY` privilégio especificado anteriormente. AWS DMS cria os nomes dos diretórios com o `DMS_` prefixo. Se você não conceder o privilégio `CREATE ANY DIRECTORY`, crie os diretórios correspondentes manualmente. Em alguns casos, ao criar os diretórios do Oracle manualmente, o usuário do Oracle especificado no endpoint de origem Oracle não é o usuário que criou esses diretórios. Nesses casos, também conceda o privilégio `READ on DIRECTORY`.

Se o endpoint de origem da Oracle estiver no Active Dataguard Standby (ADG), consulte a publicação Como usar o Binary Reader com o ADG no Database Blog. AWS

> **Note**
>
> AWS DMS O CDC não oferece suporte ao Active Dataguard Standby que não está configurado para usar o serviço de transporte automático de redo.

Em alguns casos, é possível utilizar o Oracle Managed Files (OMF) para armazenar os logs. Ou endpoint de origem está no ADG e, portanto, o privilégio CREATE ANY DIRECTORY não pode ser

concedido. Nesses casos, crie manualmente os diretórios com todos os locais de log possíveis antes de iniciar a tarefa de AWS DMS replicação. Se o AWS DMS não encontrar um diretório pré-criado que é esperado, a tarefa será interrompida. Além disso, o AWS DMS não exclui as entradas que criou na visualização `ALL_DIRECTORIES`, portanto, exclua-as manualmente.

Privilégios adicionais de conta necessários ao utilizar o Binary Reader com o Oracle ASM

Para acessar os redo logs no Automatic Storage Management (ASM) utilizando o Binary Reader, conceda os seguintes privilégios ao usuário do Oracle especificado nas configurações de conexão de endpoint do Oracle:

```
SELECT ON v_$transportable_platform
SYSASM -- To access the ASM account with Oracle 11g Release 2 (version 11.2.0.2) and
 higher, grant the Oracle endpoint user the SYSASM privilege. For older supported
 Oracle versions, it's typically sufficient to grant the Oracle endpoint user the
 SYSDBA privilege.
```

É possível validar o acesso à conta do ASM abrindo um prompt de comando e invocando uma das instruções a seguir, dependendo da versão do Oracle, conforme especificado anteriormente.

Se o privilégio `SYSDBA` for necessário, use o seguinte.

```
sqlplus asmuser/asmpassword@+asmserver as sysdba
```

Se o privilégio `SYSASM` for necessário, use o seguinte.

```
sqlplus asmuser/asmpassword@+asmserver as sysasm
```

Utilizar um Oracle Standby autogerenciado como origem com o Binary Reader para CDC no AWS DMS

Para configurar uma instância do Oracle Standby como origem ao utilizar o Binary Reader para CDC, comece com os seguintes pré-requisitos:

- AWS DMS atualmente suporta somente o Oracle Active Data Guard Standby.
- Verifique se a configuração do Oracle Data Guard utiliza:
  - Serviços de transporte de refazer para transferências automatizadas de dados de refazer.
  - Aplique serviços para aplicar automaticamente refazer banco de dados em espera.

Para confirmar se esses requisitos foram atendidos, execute a consulta a seguir.

```
SQL> select open_mode, database_role from v$database;
```

Na saída dessa consulta, confirme se o banco de dados em espera está aberto no modo SOMENTE LEITURA e se refazer está sendo aplicado automaticamente. Por exemplo: .

```
OPEN_MODE             DATABASE_ROLE
--------------------  ----------------
READ ONLY WITH APPLY  PHYSICAL STANDBY
```

Como configurar uma instância do Oracle Standby como origem ao utilizar o Binary Reader para CDC

1. Conceda privilégios adicionais necessários para acessar arquivos de log em espera.

   ```
   GRANT SELECT ON v_$standby_log TO db_user;
   ```

2. Crie um endpoint de origem para o Oracle Standby utilizando o AWS Management Console ou a AWS CLI. Ao criar o endpoint, especifique os seguintes atributos de conexão adicionais.

   ```
   useLogminerReader=N;useBfile=Y;
   ```

   > **Note**
   >
   > Em AWS DMS, você pode usar atributos de conexão extras para especificar se deseja migrar dos registros de arquivamento em vez dos redo logs. Para ter mais informações, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.

3. Configure o destino do log arquivado.

   O Binary Reader do DMS da origem do Oracle sem ASM utiliza diretórios do Oracle para acessar os redo logs arquivados. Se o banco de dados estiver configurado para utilizar Fast Recovery Area (FRA) como destino do log de arquivamento, a localização dos arquivos de refazer de arquivamento não é constante. Cada dia em que redo logs arquivados são gerados resulta em um novo diretório criado no FRA, utilizando o formato de nome de diretório AAAA_MM_DD. Por exemplo: .

```
DB_RECOVERY_FILE_DEST/SID/archivelog/YYYY_MM_DD
```

Quando o DMS precisa acessar arquivos de refazer arquivados no diretório FRA recém-criado, e o banco de dados primário de leitura e gravação está sendo utilizado como origem, o DMS cria um novo diretório Oracle ou substitui um existente, da seguinte forma.

```
CREATE OR REPLACE DIRECTORY dmsrep_taskid AS 'DB_RECOVERY_FILE_DEST/SID/archivelog/
YYYY_MM_DD';
```

Quando o banco de dados em espera está sendo utilizado como origem, o DMS não pode criar ou substituir o diretório Oracle porque o banco de dados está no modo somente leitura. Porém, é possível optar por executar uma dessas etapas adicionais:

a. Modificar `log_archive_dest_id_1` para utilizar um caminho real em vez do FRA em uma configuração que o Oracle não crie subdiretórios diariamente:

   ```
   ALTER SYSTEM SET log_archive_dest_1='LOCATION=full directory path'
   ```

   Criar um objeto no diretório Oracle para ser utilizado pelo DMS:

   ```
   CREATE OR REPLACE DIRECTORY dms_archived_logs AS 'full directory path';
   ```

b. Criar um destino adicional de log de arquivamento e um objeto no diretório Oracle apontando para esse destino. Por exemplo: .

   ```
   ALTER SYSTEM SET log_archive_dest_3='LOCATION=full directory path';
   CREATE DIRECTORY dms_archived_log AS 'full directory path';
   ```

   Adicionar um atributo de conexão adicional ao endpoint da origem da tarefa:

   ```
   archivedLogDestId=3
   ```

c. Pré-crie manualmente objetos no diretório Oracle para serem utilizados pelo DMS.

   ```
   CREATE DIRECTORY dms_archived_log_20210301 AS 'DB_RECOVERY_FILE_DEST/SID/
   archivelog/2021_03_01';
   CREATE DIRECTORY dms_archived_log_20210302 AS 'DB_RECOVERY_FILE_DEST>/SID>/
   archivelog/2021_03_02';
   ```

```
...
```

d. Criar uma tarefa do programador do Oracle que seja executada diariamente e criar o diretório necessário.

Utilizar um banco de dados gerenciado pelo usuário no Oracle Cloud Infrastructure (OCI) como origem da CDC no AWS DMS

Um banco de dados gerenciado pelo usuário é um banco de dados que você configura e controla, como um banco de dados Oracle criado em uma máquina virtual (VM), bare metal ou servidor Exadata. Ou bancos de dados que você configura e controla que são executados em uma infraestrutura dedicada, como o Oracle Cloud Infrastructure (OCI). As informações a seguir descrevem os privilégios e as configurações necessárias ao utilizar um banco de dados Oracle gerenciado pelo usuário no OCI como origem para a captura de dados de alteração (CDC) no AWS DMS.

Para configurar um banco de dados Oracle gerenciado pelo usuário hospedado pelo OCI como origem para a captura de dados de alteração

1. Conceda os privilégios da conta de usuário necessários para utilizar um banco de dados de origem do Oracle gerenciado pelo usuário no OCI. Para obter mais informações, consulte Privilégios de conta para um endpoint de origem do Oracle autogerenciado.
2. Conceda os privilégios da conta necessários ao utilizar o Binary Reader para acessar os redo logs. Para obter mais informações, consulte Privilégios da conta necessários ao utilizar o Binary Reader.
3. Adicione os privilégios da conta necessários ao utilizar o Binary Reader com o Oracle Automatic Storage Management (ASM). Para obter mais informações, consulte Privilégios adicionais da conta necessários ao utilizar o Binary Reader com o Oracle ASM.
4. Configure o registro em log suplementar. Para obter mais informações, consulte Configuração do registro em log suplementar.
5. Configure a criptografia de TDE. Para obter mais informações, consulte Métodos de criptografia ao utilizar um banco de dados Oracle como um endpoint de origem.

As limitações a seguir se aplicam ao replicar dados de um banco de dados de origem do Oracle no Oracle Cloud Infrastructure (OCI).

Limitações

- O DMS não suporta o uso do Oracle LogMiner para acessar os redo logs.
- O DMS não é compatível com o banco de dados autônomo.

Trabalhando com um banco AWS de dados Oracle gerenciado como fonte para AWS DMS

Um banco de dados AWS gerenciado é um banco de dados que está em um serviço da Amazon, como Amazon RDS, Amazon Aurora ou Amazon S3. A seguir, você pode encontrar os privilégios e configurações que você precisa definir ao usar um banco de dados Oracle AWS gerenciado com o. AWS DMS

Privilégios de conta de usuário necessários em uma fonte Oracle AWS gerenciada para AWS DMS

Conceda os seguintes privilégios à conta de usuário do Oracle especificada na definição do endpoint de origem do Oracle.

> **Important**
>
> Para todos os valores de parâmetros, como *db_user* e *any-replicated-table*, o Oracle pressupõe que o valor está todo em letras maiúsculas, a menos que você especifique o valor com um identificador que diferencia letras maiúsculas de minúsculas. Por exemplo, suponha que você crie um valor de *db_user* sem utilizar aspas, como em `CREATE USER` *myuser* ou `CREATE USER MYUSER`. Nesse caso, o Oracle identifica e armazena o valor como todo em maiúsculas (`MYUSER`). Se você utilizar aspas, como em `CREATE USER "MyUser"` ou`CREATE USER 'MyUser'`, o Oracle identificará e armazenará o valor com distinção entre maiúsculas e minúsculas que você especificar (`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.
```

Além disso, conceda as permissões SELECT e EXECUTE em objetos SYS utilizando o procedimento do Amazon RDS rdsadmin.rdsadmin_util.grant_sys_object conforme mostrado. Para obter mais informações, consulte Conceder privilégios SELECT ou EXECUTE a objetos SYS.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_VIEWS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_TAB_PARTITIONS', 'db_user',
 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_INDEXES', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_OBJECTS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_TABLES', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_USERS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_CATALOG', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_CONSTRAINTS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_CONS_COLUMNS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_TAB_COLS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_IND_COLUMNS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_LOG_GROUPS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$ARCHIVED_LOG', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOG', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOGFILE', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$DATABASE', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$THREAD', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$PARAMETER', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$NLS_PARAMETERS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$TIMEZONE_NAMES', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$TRANSACTION', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$CONTAINERS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('OBJ$', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('ALL_ENCRYPTED_COLUMNS', 'db_user',
 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOGMNR_LOGS', 'db_user', 'SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$LOGMNR_CONTENTS','db_user','SELECT');
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_LOGMNR', 'db_user', 'EXECUTE');

-- (as of Oracle versions 12.1 and higher)
exec rdsadmin.rdsadmin_util.grant_sys_object('REGISTRY$SQLPATCH', 'db_user', 'SELECT');

-- (for Amazon RDS Active Dataguard Standby (ADG))
exec rdsadmin.rdsadmin_util.grant_sys_object('V_$STANDBY_LOG', 'db_user', 'SELECT');

-- (for transparent data encryption (TDE))
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('ENC$', 'db_user', 'SELECT');

-- (for validation with LOB columns)
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_CRYPTO', 'db_user', 'EXECUTE');

-- (for binary reader)
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_DIRECTORIES','db_user','SELECT');

-- Required when the source database is Oracle Data guard, and Oracle Standby is used
 in the latest release of DMS version 3.4.6, version 3.4.7, and higher.

exec rdsadmin.rdsadmin_util.grant_sys_object('V_$DATAGUARD_STATS', 'db_user',
 'SELECT');
```

Para obter mais informações sobre como utilizar o Amazon RDS Active Dataguard Standby (ADG) com o AWS DMS , consulte Utilizar um Amazon RDS Oracle Standby (réplica de leitura) como origem com o Binary Reader para CDC no AWS DMS.

Para obter mais informações sobre como usar o Oracle TDE com AWS DMS, consulteMétodos de criptografia suportados para usar o Oracle como fonte para AWS DMS.

Pré-requisitos para tratar transações abertas do Oracle Standby

Ao usar AWS DMS as versões 3.4.6 e superiores, execute as etapas a seguir para lidar com transações abertas para o Oracle Standby.

1. Crie um link de banco de dados nomeado AWSDMS_DBLINK no banco de dados primário. O *DMS_USER* utilizará o link de banco de dados para conectar-se ao banco de dados primário. Observe que o link de banco de dados é executado na instância em espera para consultar as transações abertas em execução no banco de dados primário. Veja o exemplo a seguir.

   ```
   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. Verifique se a conexão ao link de banco de dados que utiliza o *DMS_USER* está estabelecida conforme mostrado no exemplo a seguir.

```
select 1 from dual@AWSDMS_DBLINK
```

Configurando uma fonte Oracle AWS gerenciada para AWS DMS

Antes de usar um banco AWS de dados Oracle gerenciado como fonte para AWS DMS, execute as seguintes tarefas para o banco de dados Oracle:

- Ative backups automáticos. Para obter mais informações sobre como ativar backups automáticos, consulte Ativar backups automáticos no Guia do usuário do Amazon RDS.
- Configure o registro em log suplementar.
- Configure o arquivamento. O arquivamento dos redo logs da sua instância de banco de dados Amazon RDS for Oracle AWS DMS permite recuperar as informações de log usando o LogMiner Oracle ou o Binary Reader.

Como configurar o arquivamento

1. Execute o comando `rdsadmin.rdsadmin_util.set_configuration` para configurar o arquivamento.

   Por exemplo, para reter os redo logs arquivados por 24 horas, execute o comando a seguir.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
commit;
```

> **Note**
>
> A confirmação é necessária para que uma alteração entre em vigor.

2. Verifique se o armazenamento tem espaço suficiente para os redo logs arquivados durante o período de retenção especificado. Por exemplo, se o período de retenção for de 24 horas, calcule o tamanho total dos redo logs arquivados acumulados em uma hora típica de processamento de transações e multiplique esse total por 24. Compare esse total calculado de 24 horas com o espaço de armazenamento disponível e decida se você tem espaço de armazenamento suficiente para tratar o processamento de transações de um total 24 horas.

Como configurar o registro em log suplementar

1. Execute o comando a seguir para ativar o registro em log suplementar no nível de banco de dados.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');
```

2. O exemplo a seguir ativa o registro em log suplementar de chave primária.

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD','PRIMARY KEY');
```

3. (Opcional) Ative o registro em log suplementar em nível de chave no nível da tabela.

   O banco de dados de origem incorre em pequenos custos quando o registro em log suplementar em nível de chave está ativado. Portanto, se estiver migrando apenas um subconjunto das tabelas, ative o registro em log suplementar de nível de chave no nível da tabela. Para ativar o registro em log suplementar de nível de chave no nível da tabela, execute o seguinte comando.

```
alter table table_name add supplemental log data (PRIMARY KEY) columns;
```

Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS

Você pode configurar AWS DMS para acessar os redo logs de origem da instância Amazon RDS for Oracle usando o Binary Reader for CDC.

> **Note**
>
> Para usar o Oracle LogMiner, os privilégios mínimos de conta de usuário necessários são suficientes. Para ter mais informações, consulte Privilégios de conta de usuário necessários em uma fonte Oracle AWS gerenciada para AWS DMS.

Para usar o AWS DMS Binary Reader, especifique configurações adicionais e atributos de conexão extras para o endpoint de origem Oracle, dependendo da sua AWS DMS versão.

A compatibilidade com o Binary Reader está disponível nas seguintes versões do Amazon RDS para Oracle:

- Oracle 11.2, versões 11.2.0.4V11 e superior
- Oracle 12.1, versões 12.1.0.2.V7 e superior
- Oracle 12.2, todas as versões
- Oracle 18.0, todas as versões
- Oracle 19.0, todas as versões

Como configurar a CDC utilizando o Binary Reader

1. Faça login no banco de dados de origem do Amazon RDS para Oracle como o usuário mestre e execute os seguintes procedimentos armazenados para criar os diretórios em nível de servidor.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

2. Conceda os seguintes privilégios à conta de usuário do Oracle utilizada para acessar o endpoint de origem do Oracle:

```
GRANT READ ON DIRECTORY ONLINELOG_DIR TO db_user;
GRANT READ ON DIRECTORY ARCHIVELOG_DIR TO db_user;
```

3. Defina os atributos de conexão adicionais a seguir no endpoint de origem Amazon RDS Oracle.

   - Para as versões 11.2 e 12.1 do RDS Oracle, defina o seguinte.

```
useLogminerReader=N;useBfile=Y;accessAlternateDirectly=false;useAlternateFolderForOnline=
oraclePathPrefix=/rdsdbdata/db/{$DATABASE_NAME}_A/;usePathPrefix=/rdsdbdata/
log/;replacePathPrefix=true;
```

   - Para as versões 12.2, 18.0 e 19.0 do RDS Oracle, defina o seguinte.

```
useLogminerReader=N;useBfile=Y;
```

> **Note**
>
> Verifique se não há nenhum espaço em branco após o separador de ponto e vírgula (;) em várias configurações de atributo, por exemplo, `oneSetting;thenAnother`.

Para obter mais informações para configurar uma tarefa de CDC, consulte Configuração da CDC em um banco de dados Oracle de origem.

Utilizar um Amazon RDS Oracle Standby (réplica de leitura) como origem com o Binary Reader para CDC no AWS DMS

Verifique os seguintes pré-requisitos para utilizar o Amazon RDS para Oracle Standby como origem ao utilizar o Binary Reader para CDC no AWS DMS:

- Utilize o usuário mestre do Oracle para configurar o Binary Reader.
- Certifique-se de que AWS DMS atualmente suporta o uso somente do Oracle Active Data Guard Standby.

Depois de fazer isso, utilize o procedimento a seguir para utilizar o RDS para Oracle Standby como origem ao utilizar o Binary Reader para CDC.

Como configurar um RDS para Oracle Standby como origem ao utilizar o Binary Reader para CDC

1. Faça login na instância primária do RDS para Oracle como usuário mestre.
2. Execute os seguintes procedimentos armazenados conforme documentado no Guia do usuário do Amazon RDS para criar os diretórios no nível do servidor.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

3. Identifique os diretórios criados na etapa 2.

```
SELECT directory_name, directory_path FROM all_directories
WHERE directory_name LIKE ( 'ARCHIVELOG_DIR_%' )
        OR directory_name LIKE ( 'ONLINELOG_DIR_%' )
```

   Por exemplo, o código anterior exibe uma lista de diretórios como a seguinte.

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. Conceda o privilégio `Read` nos diretórios anteriores à conta de usuário Oracle utilizada para acessar o Oracle Standby.

```
GRANT READ ON DIRECTORY ARCHIVELOG_DIR_A TO db_user;
GRANT READ ON DIRECTORY ARCHIVELOG_DIR_B TO db_user;
GRANT READ ON DIRECTORY ONLINELOG_DIR_A TO db_user;
GRANT READ ON DIRECTORY ONLINELOG_DIR_B TO db_user;
```

5. Execute uma troca de log de arquivamento na instância primária. Isso garante que as alterações de `ALL_DIRECTORIES` também sejam transferidas para o Oracle Standby.
6. Execute uma consulta `ALL_DIRECTORIES` no Oracle Standby para confirmar se as alterações foram aplicadas.
7. Crie um endpoint de origem para o Oracle Standby usando o AWS DMS Management Console ou AWS Command Line Interface ()AWS CLI. Ao criar o endpoint, especifique os seguintes atributos de conexão adicionais.

```
useLogminerReader=N;useBfile=Y;archivedLogDestId=1;additionalArchivedLogDestId=2
```

8. Depois de criar o endpoint, use Testar conexão de endpoint na página Criar endpoint do console ou o AWS CLI `test-connection` comando para verificar se a conectividade foi estabelecida.

Limitações no uso da Oracle como fonte para AWS DMS

As seguintes limitações se aplicam quando um banco de dados Oracle é utilizado como origem do AWS DMS:

- AWS DMS oferece suporte aos tipos de dados Oracle Extended na AWS DMS versão 3.5.0 e superior.
- AWS DMS não suporta nomes de objetos longos (mais de 30 bytes).
- AWS DMS não oferece suporte a índices baseados em funções.

- Se você gerenciar o registro em log suplementar e realizar transformações em qualquer uma das colunas, verifique se o registro em log suplementar está ativado para todos os campos e colunas. Para obter mais informações sobre como configurar o registro em log suplementar, consulte os seguintes tópicos.
  - Para um banco de dados Oracle autogerenciado como origem, consulte Configuração de registro em log suplementar.
  - Para um banco AWS de dados de origem Oracle gerenciado, consulte Configurando uma fonte Oracle AWS gerenciada para AWS DMS.
- AWS DMS não oferece suporte ao banco de dados raiz de contêineres multilocatários (CDB $ROOT). Ele é compatível com um PDB utilizando o Binary Reader.
- AWS DMS não suporta restrições diferidas.
- Na AWS DMS versão 3.5.1 e superior, os LOBs seguros são suportados somente por meio da realização de uma pesquisa de LOB.
- AWS DMS suporta a `rename table` *`table-name`* `to` *`new-table-name`* sintaxe de todas as versões 11 e superiores do Oracle suportadas. Essa sintaxe não é compatível para nenhum banco de dados de origem Oracle versão 10.
- AWS DMS não replica os resultados da instrução DDL. `ALTER TABLE ADD` *`column data_type`* `DEFAULT` *`default_value`* Em vez de replicar *`default_value`* para o destino, ele define a nova coluna como `NULL`.
- Ao usar a AWS DMS versão 3.4.7 ou superior, para replicar as alterações resultantes das operações de partição ou subpartição, faça o seguinte antes de iniciar uma tarefa do DMS.
  - Crie manualmente a estrutura da tabela particionada (DDL);
  - Verifique se a DDL é a mesma na origem e no destino do Oracle;
  - Defina o atributo de conexão adicional `enableHomogenousPartitionOps=true`.

  Para obter mais informações sobre `enableHomogenousPartitionOps`, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS. Além disso, observe que em tarefas FULL+CDC, o DMS não replica as alterações de dados capturadas como parte das alterações em cache. Nesse caso de uso, recrie a estrutura da tabela no destino do Oracle e recarregue as tabelas em questão.

  Antes da AWS DMS versão 3.4.7:

O DMS não replica alterações de dados resultantes de operações de partição ou subpartição (`ADD`, `DROP`, `EXCHANGE` e `TRUNCATE`). Essas atualizações podem causar os seguintes erros durante a replicação:

- Para operações `ADD`, atualizações e exclusões nos dados adicionados podem gerar um aviso de "0 linhas afetadas".
- Para operações `DROP` e `TRUNCATE`, novas inserções podem gerar erros de "duplicatas".
- Operações `EXCHANGE` podem gerar um aviso de "0 linhas afetadas" e erros de "duplicatas".

Para replicar alterações resultantes de operações de partição ou subpartição, recarregue as tabelas em questão. Depois de adicionar uma nova partição vazia, as operações na partição adicionada são replicadas para o destino como normais.

- AWS DMS versões anteriores à 3.4 não suportam alterações de dados no destino que resultam da execução da `CREATE TABLE AS` instrução na fonte. No entanto, a nova tabela é criada no destino.
- AWS DMS não captura as alterações feitas pelo `DBMS_REDEFINITION` pacote Oracle, por exemplo, os metadados da tabela e o `OBJECT_ID` campo.
- AWS DMS mapeia colunas BLOB e CLOB vazias para o `NULL` alvo.
- Ao capturar alterações com o Oracle 11 LogMiner, uma atualização em uma coluna CLOB com um comprimento de string maior que 1982 é perdida e o destino não é atualizado.
- Durante a captura de dados de alteração (CDC), AWS DMS não oferece suporte a atualizações em lote em colunas numéricas definidas como chave primária.
- AWS DMS não suporta determinados `UPDATE` comandos. O exemplo a seguir é um comando `UPDATE` não compatível.

```
UPDATE TEST_TABLE SET KEY=KEY+1;
```

Aqui, `TEST_TABLE` é o nome da tabela e `KEY` é uma coluna numérica definida como chave primária.

- AWS DMS não suporta o modo LOB completo para carregar colunas LONG e LONG RAW. Em vez disso, é possível utilizar o modo LOB limitado para migrar esses tipos de dados para um destino do Oracle. No modo LOB limitado, AWS DMS trunca todos os dados de 64 KB definidos como colunas LONG ou LONG RAW com mais de 64 KB.
- AWS DMS não suporta o modo LOB completo para carregar colunas XMLTYPE. Em vez disso, é possível utilizar o modo LOB limitado para migrar colunas XMLTYPE para um destino do Oracle.

No modo LOB limitado, o DMS trunca todos os dados maiores que a variável “Tamanho máximo de LOB” definida pelo usuário. O valor máximo recomendado para “Tamanho máximo de LOB” é 100 MB.

- AWS DMS não replica tabelas cujos nomes contêm apóstrofos.
- AWS DMS suporta CDC a partir de visualizações materializadas. Mas o DMS não é compatível com a CDC de nenhuma outra visualização.
- AWS DMS não oferece suporte ao CDC para tabelas organizadas por índice com um segmento de estouro.
- AWS DMS não suporta a `Drop Partition` operação para tabelas particionadas por referência com `enableHomogenousPartitionOps` definido como. `true`
- Quando você usa o Oracle LogMiner para acessar os redo logs, AWS DMS tem as seguintes limitações:
  - Somente para o Oracle 12, AWS DMS não replica nenhuma alteração nas colunas LOB.
  - Para todas as versões do Oracle, AWS DMS não replica o resultado das `UPDATE` operações em colunas `XMLTYPE` LOB.
  - AWS DMS não suporta transações XA na replicação ao usar o Oracle LogMiner.
  - O Oracle LogMiner não suporta conexões com um banco de dados conectável (PDB). Para conectar-se a um PDB, acesse os redo logs utilizando o Binary Reader.
  - As operações SHRINK SPACE não são compatíveis.
- Quando você usa o Binary Reader, AWS DMS tem estas limitações:
  - Ele não é compatível com clusters de tabela.
  - Ele é compatível somente com operações `SHRINK SPACE` em nível da tabela. Esse nível inclui a tabela completa, as partições e as subpartições.
  - Ele não é compatível com alterações em tabelas organizadas por índice com compactação de chaves.
  - Ele não é compatível com a implementação de redo logs on-line em dispositivos brutos.
  - O Binary Reader é compatível com a TDE somente para bancos de dados Oracle autogerenciados, uma vez que o RDS para Oracle não é compatível com a recuperação de senha de carteira para chaves de criptografia da TDE.
- AWS DMS não suporta conexões com uma fonte Oracle do Amazon RDS usando um proxy Oracle Automatic Storage Management (ASM).
- AWS DMS não suporta colunas virtuais.

- AWS DMS não suporta o tipo de `ROWID` dados ou visualizações materializadas com base em uma coluna ROWID.

  AWS DMS tem suporte parcial para Oracle Materialized Views. Para cargas máximas, o DMS pode fazer uma cópia da carga máxima de uma visão materializada do Oracle. O DMS copia a visão materializada como uma tabela base para o sistema de destino e ignora todas as colunas ROWID na visão materializada. Para a replicação contínua (CDC), o DMS tenta replicar as alterações nos dados da visão materializada, mas os resultados podem não ser ideais. Especificamente, se a visão materializada estiver completamente atualizada, o DMS replica exclusões individuais de todas as linhas, seguidas por inserções individuais em todas as linhas. Esse é um exercício que consome muitos recursos e pode ter um desempenho inadequado em visões materializadas com um grande número de linhas. Para a replicação contínua em que as visões materializadas fazem uma atualização rápida, o DMS tenta processar e replicar as alterações de dados de atualização rápida. Em ambos os casos, o DMS ignora qualquer coluna ROWID na visão materializada.
- AWS DMS não carrega nem captura tabelas temporárias globais.
- Para destinos do S3 que utilizam a replicação, ative o registro em log suplementar em cada coluna para que as atualizações da linha de origem possam capturar cada valor da coluna. Veja a seguir um exemplo: `alter table yourtablename add supplemental log data (all) columns;`.
- Uma atualização de uma linha com uma chave exclusiva composta que contém `null` não pode ser replicada no destino.
- AWS DMS não suporta o uso de várias chaves de criptografia Oracle TDE no mesmo endpoint de origem. Cada endpoint pode ter somente um atributo para o nome da chave de criptografia da TDE "`securityDbEncryptionName`" e uma senha da TDE para essa chave.
- Ao replicar do Amazon RDS for Oracle, o TDE é suportado somente com tablespace criptografado e usando Oracle. LogMiner
- AWS DMS não suporta várias operações de renomeação de tabelas em rápida sucessão.
- Ao usar o Oracle 19.0 como fonte, AWS DMS não oferece suporte aos seguintes recursos:
  - Redirecionamento de DML do Data-guard
  - Tabelas particionadas híbridas
  - Contas Oracle somente de esquemas
- AWS DMS não suporta a migração de tabelas ou visualizações do tipo `BIN$` ou`DR$`.
- A partir do Oracle 18.x, AWS DMS não oferece suporte à captura de dados de alteração (CDC) do Oracle Express Edition (Oracle Database XE).

- Ao migrar dados de uma coluna CHAR, o DMS trunca todos os espaços à direita.
- AWS DMS não oferece suporte à replicação de contêineres de aplicativos.
- AWS DMS não suporta a execução do Oracle Flashback Database e dos pontos de restauração, pois essas operações afetam a consistência dos arquivos Oracle Redo Log.
- O procedimento de carga direta `INSERT` com a opção de execução paralela não é compatível nos seguintes casos:
  - Tabelas não compactadas com mais de 255 colunas
  - O tamanho da linha excede 8K
  - Tabelas do Exadata HCC
  - Banco de dados em execução na plataforma Big Endian
- Uma tabela de origem sem chave primária ou exclusiva exige que o registro em log suplementar ALL COLUMN esteja ativado. Ele cria mais atividades no redo log e pode aumentar a latência da CDC do DMS.
- AWS DMS não migra dados de colunas invisíveis em seu banco de dados de origem. Para incluir essas colunas no escopo da migração, utilize a instrução `ALTER TABLE` para tornar essas colunas visíveis.

Suporte de SSL para um endpoint do Oracle

AWS DMS Os endpoints Oracle oferecem suporte a SSL V3 para os modos `none` e `verify-ca` SSL. Para utilizar SSL com um endpoint do Oracle, carregue o Oracle Wallet para o endpoint, em vez de nos arquivos de certificado .pem.

Tópicos

- Utilizar um certificado existente para SSL no Oracle
- Utilizar um certificado autoassinado para SSL no Oracle

Utilizar um certificado existente para SSL no Oracle

Para utilizar uma instalação cliente existente do Oracle e criar o arquivo Oracle Wallet a partir de um arquivo de certificado CA, siga as seguintes etapas.

Como utilizar uma instalação cliente existente do Oracle para SSL no Oracle com o AWS DMS

1. Defina a variável do sistema `ORACLE_HOME` como local do seu diretório dbhome_1, executando o comando a seguir.

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

2. Anexe `$ORACLE_HOME/lib` ao sistema variável `LD_LIBRARY_PATH`.

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. Crie um diretório para o Oracle Wallet em `$ORACLE_HOME/ssl_wallet`.

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

4. Coloque o arquivo `.pem` de certificado da CA no diretório `ssl_wallet`. Se você utilizar o Amazon RDS, poderá baixar o arquivo raiz do certificado a CA `rds-ca-2015-root.pem` hospedado pelo Amazon RDS. Para obter mais informações sobre como baixar esse arquivo, consulte Utilizar o SSL/TLS para criptografar uma conexão com uma instância de banco de dados no Guia do usuário do Amazon RDS.
5. Execute os seguintes comandos para criar o 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
```

Ao concluir as etapas anteriores, é possível importar o arquivo wallet com a chamada de API `ImportCertificate`, especificando o parâmetro certificate-wallet. Utilize o certificado wallet importado ao selecionar `verify-ca` como modo SSL ao criar ou modificar o seu endpoint do Oracle.

> **Note**
>
> As carteiras Oracle são arquivos binários. AWS O DMS aceita esses arquivos como estão.

Utilizar um certificado autoassinado para SSL no Oracle

Para utilizar um certificado autoassinado para SSL no Oracle, siga as etapas a seguir, pressupondo uma senha de carteira Oracle de `oracle123`.

Para usar um certificado autoassinado para Oracle SSL com AWS DMS

1. Crie um diretório que será utilizado para trabalhar com o certificado autoassinado.

   ```
   mkdir -p /u01/app/oracle/self_signed_cert
   ```

2. Altere para o diretório que você criou na etapa anterior.

   ```
   cd /u01/app/oracle/self_signed_cert
   ```

3. Crie uma chave raiz.

   ```
   openssl genrsa -out self-rootCA.key 2048
   ```

4. Autoassine um certificado raiz utilizando a chave criada na etapa anterior.

   ```
   openssl req -x509 -new -nodes -key self-rootCA.key
           -sha256 -days 3650 -out self-rootCA.pem
   ```

   Utilize os parâmetros de entrada, como os seguintes:

   - `Country Name (2 letter code) [XX]`, por exemplo: `AU`
   - `State or Province Name (full name) []`, por exemplo: `NSW`
   - `Locality Name (e.g., city) [Default City]`, por exemplo: `Sydney`
   - `Organization Name (e.g., company) [Default Company Ltd]`, por exemplo: `AmazonWebService`
   - `Organizational Unit Name (e.g., section) []`, por exemplo: `DBeng`
   - `Common Name (e.g., your name or your server's hostname) []`, por exemplo: `aws`

- `Email Address []`, por exemplo: abcd.efgh@amazonwebservice.com

5. Crie um diretório do Oracle Wallet para o banco de dados Oracle.

```
mkdir -p /u01/app/oracle/wallet
```

6. Crie um novo Oracle Wallet.

```
orapki wallet create -wallet "/u01/app/oracle/wallet" -pwd oracle123 -
auto_login_local
```

7. Adicione o certificado raiz ao 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. Liste os conteúdos do Oracle Wallet. A lista deve incluir o certificado raiz.

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

Por exemplo, isso pode ser exibido de forma semelhante à seguinte.

```
Requested Certificates:
User Certificates:
Trusted Certificates:
Subject:        CN=aws,OU=DBeng,O= AmazonWebService,L=Sydney,ST=NSW,C=AU
```

9. Gere o Certificate Signing Request (CSR - Solicitação de assinatura de certificado) utilizando o utilitário ORAPKI.

```
orapki wallet add -wallet "/u01/app/oracle/wallet" -pwd oracle123
-dn "CN=aws" -keysize 2048 -sign_alg sha256
```

10. Execute o seguinte comando .

```
openssl pkcs12 -in /u01/app/oracle/wallet/ewallet.p12 -nodes -out /u01/app/oracle/
wallet/nonoracle_wallet.pem
```

Isso produz uma saída semelhante à seguinte.

```
Enter Import Password:
```

```
MAC verified OK
Warning unsupported bag type: secretBag
```

11. Coloque "dms" como o nome comum.

```
openssl req -new -key /u01/app/oracle/wallet/nonoracle_wallet.pem -out certdms.csr
```

Utilize os parâmetros de entrada, como os seguintes:

- `Country Name (2 letter code) [XX]`, por exemplo: `AU`
- `State or Province Name (full name) []`, por exemplo: `NSW`
- `Locality Name (e.g., city) [Default City]`, por exemplo: `Sydney`
- `Organization Name (e.g., company) [Default Company Ltd]`, por exemplo: `AmazonWebService`
- `Organizational Unit Name (e.g., section) []`, por exemplo: `aws`
- `Common Name (e.g., your name or your server's hostname) []`, por exemplo: `aws`
- `Email Address []`, por exemplo: abcd.efgh@amazonwebservice.com

Verifique se isso não é o mesmo que a etapa 4. É possível fazer isso, por exemplo, alterando o nome da unidade organizacional para um nome diferente, conforme mostrado.

Insira os atributos adicionais a seguir para serem enviados com a solicitação de certificado.

- `A challenge password []`, por exemplo: `oracle123`
- `An optional company name []`, por exemplo: `aws`

12. Obtenha a assinatura do certificado.

```
openssl req -noout -text -in certdms.csr | grep -i signature
```

A chave de assinatura desta postagem é `sha256WithRSAEncryption`.

13. Utilize o seguinte comando para gerar o arquivo de certificado (`.crt`):

```
openssl x509 -req -in certdms.csr -CA self-rootCA.pem -CAkey self-rootCA.key
-CAcreateserial -out certdms.crt -days 365 -sha256
```

Isso exibe uma saída semelhante à seguinte.

```
Signature ok
subject=/C=AU/ST=NSW/L=Sydney/O=awsweb/OU=DBeng/CN=aws
Getting CA Private Key
```

14. Adicione o certificado à carteira.

```
orapki wallet add -wallet /u01/app/oracle/wallet -pwd oracle123 -user_cert -cert
 certdms.crt
```

15. Visualize a carteira. Deve haver duas entradas. Consulte o seguinte código:

```
orapki wallet display -wallet /u01/app/oracle/wallet -pwd oracle123
```

16. Configure o arquivo `sqlnet.ora` (`$ORACLE_HOME/network/admin/sqlnet.ora`).

```
WALLET_LOCATION =
   (SOURCE =
     (METHOD = FILE)
     (METHOD_DATA =
       (DIRECTORY = /u01/app/oracle/wallet/)
     )
   )

SQLNET.AUTHENTICATION_SERVICES = (NONE)
SSL_VERSION = 1.0
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
```

17. Interrompa o receptor do Oracle.

```
lsnrctl stop
```

18. Adicione entradas para SSL no arquivo `listener.ora` (`$ORACLE_HOME/network/admin/listener.ora`).

```
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
```

```
    (METHOD_DATA =
      (DIRECTORY = /u01/app/oracle/wallet/)
    )
  )

SID_LIST_LISTENER =
 (SID_LIST =
  (SID_DESC =
   (GLOBAL_DBNAME = SID)
   (ORACLE_HOME = ORACLE_HOME)
   (SID_NAME = SID)
  )
 )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCPS)(HOST = localhost.localdomain)(PORT = 1522))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
```

19. Configure o arquivo `tnsnames.ora` (`$ORACLE_HOME/network/admin/tnsnames.ora`).

```
<SID>=
(DESCRIPTION=
        (ADDRESS_LIST =
                (ADDRESS=(PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT =
 1521))
        )
        (CONNECT_DATA =
                (SERVER = DEDICATED)
                (SERVICE_NAME = <SID>)
        )
)

<SID>_ssl=
(DESCRIPTION=
        (ADDRESS_LIST =
                (ADDRESS=(PROTOCOL = TCPS)(HOST = localhost.localdomain)(PORT =
 1522))
        )
        (CONNECT_DATA =
```

```
                (SERVER = DEDICATED)
                (SERVICE_NAME = <SID>)
        )
)
```

20. Reinicie o receptor do Oracle.

```
lsnrctl start
```

21. Mostre o status do receptor do Oracle.

```
lsnrctl status
```

22. Teste a conexão SSL ao banco de dados no host local, utilizando sqlplus e a a entrada SSL tnsnames.

```
sqlplus -L ORACLE_USER@SID_ssl
```

23. Verifique se você se conectou com êxito utilizando SSL.

```
SELECT SYS_CONTEXT('USERENV', 'network_protocol') FROM DUAL;

SYS_CONTEXT('USERENV','NETWORK_PROTOCOL')
--------------------------------------------------------------------------------
tcps
```

24. Altere o diretório para o diretório com o certificado autoassinado.

```
cd /u01/app/oracle/self_signed_cert
```

25. Crie uma nova carteira Oracle de cliente AWS DMS para usar.

```
orapki wallet create -wallet ./ -auto_login_only
```

26. Adicione o certificado raiz autoassinado ao Oracle Wallet.

```
orapki wallet add -wallet ./ -trusted_cert -cert self-rootCA.pem -auto_login_only
```

27. Liste o conteúdo da carteira Oracle AWS DMS para uso. A lista deve incluir o certificado raiz autoassinado.

```
orapki wallet display -wallet ./
```

Isso produz uma saída semelhante à seguinte.

```
Trusted Certificates:
Subject:        CN=aws,OU=DBeng,O=AmazonWebService,L=Sydney,ST=NSW,C=AU
```

28. Faça upload da carteira Oracle que você acabou de criar AWS DMS.

Métodos de criptografia suportados para usar o Oracle como fonte para AWS DMS

Na tabela a seguir, você pode encontrar os métodos de criptografia de dados transparente (TDE) que são AWS DMS compatíveis ao trabalhar com um banco de dados de origem Oracle.

Método de acesso de logs redo	Espaço de tabela de TDE	Coluna de TDE
Oráculo LogMiner	Sim	Sim
Binary Reader	Sim	Sim

AWS DMS oferece suporte ao Oracle TDE ao usar o Binary Reader, tanto no nível da coluna quanto no nível do espaço de tabela. Para usar a criptografia TDE AWS DMS, primeiro identifique o local da carteira Oracle em que a chave de criptografia e a senha do TDE estão armazenadas. Identifique a chave de criptografia e a senha corretas da TDE para o endpoint de origem do Oracle.

Para identificar e especificar a chave de criptografia e a senha para a criptografia da TDE

1. Execute a consulta a seguir para encontrar a carteira de criptografia do Oracle no host do banco de dados Oracle.

```
SQL> SELECT WRL_PARAMETER FROM V$ENCRYPTION_WALLET;

WRL_PARAMETER
--------------------------------------------------------------------------------
/u01/oracle/product/12.2.0/dbhome_1/data/wallet/
```

Aqui, `/u01/oracle/product/12.2.0/dbhome_1/data/wallet/` é a localização da carteira.

2. Obtenha o ID da chave mestra utilizando uma das seguintes opções de criptografia, dependendo de qual delas retorna esse valor.

   a. Para a criptografia em nível de tabela ou de coluna, execute as consultas a seguir.

```
SQL> SELECT OBJECT_ID FROM ALL_OBJECTS
WHERE OWNER='DMS_USER' AND OBJECT_NAME='TEST_TDE_COLUMN' AND
 OBJECT_TYPE='TABLE';

OBJECT_ID
---------------
81046
SQL> SELECT MKEYID FROM SYS.ENC$ WHERE OBJ#=81046;

MKEYID
------------
AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

   Aqui, `AWGDC9glSk8Xv+3bVveiVSg` é o ID da chave mestra (`MKEYID`). Se você obtiver um valor para `MKEYID`, poderá continuar com a Etapa 3. Caso contrário, continue na Etapa 2.2.

> **Note**
>
> Os caracteres à direita da string `'A'` (`AAA...`) não fazem parte do valor.

   b. Para a criptografia em nível de espaço para tabela, execute as consultas a seguir.

```
SQL> SELECT TABLESPACE_NAME, ENCRYPTED FROM dba_tablespaces;
TABLESPACE_NAME                ENC
------------------------------ ---
SYSTEM                         NO
SYSAUX                         NO
UNDOTBS1                       NO
TEMP                           NO
USERS                          NO
TEST_ENCRYT                    YES
SQL> SELECT name,utl_raw.cast_to_varchar2( utl_encode.base64_encode('01'||
substr(mkeyid,1,4))) ||
```

```
 utl_raw.cast_to_varchar2( utl_encode.base64_encode(substr(mkeyid,5,length(mkeyid))))
 masterkeyid_base64
FROM (SELECT t.name, RAWTOHEX(x.mkid) mkeyid FROM v$tablespace t, x$kcbtek x
 WHERE t.ts#=x.ts#)
WHERE name = 'TEST_ENCRYT';

NAME                           MASTERKEYID_BASE64
------------------------------ ----------------------------------
TEST_ENCRYT                    AWGDC9glSk8Xv+3bVveiVSg=
```

Aqui, `AWGDC9glSk8Xv+3bVveiVSg` é o ID da chave mestra (`TEST_ENCRYT`). Se as etapas 2.1 e 2.2 retornarem um valor, elas serão sempre idênticas.

O caractere à direita de `'='` não faz parte do valor.

3. Na linha de comando, liste as entradas da carteira de criptografia no host do banco de dados Oracle de origem.

```
$ mkstore -wrl /u01/oracle/product/12.2.0/dbhome_1/data/wallet/ -list
Oracle Secret Store entries:
ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.AY1mRA8OXU9Qvzo3idU4OH4AAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY
ORACLE.SECURITY.ID.ENCRYPTION.
ORACLE.SECURITY.KB.ENCRYPTION.
ORACLE.SECURITY.KM.ENCRYPTION.AY1mRA8OXU9Qvzo3idU4OH4AAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Encontre a entrada que contém o ID da chave mestra que você encontrou na etapa 2 (`AWGDC9glSk8Xv+3bVveiVSg`). Essa entrada é o nome da chave de criptografia da TDE.

4. Visualize os detalhes da entrada que você localizou na etapa anterior.

```
$ mkstore -wrl /u01/oracle/product/12.2.0/dbhome_1/data/wallet/ -viewEntry
 ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Oracle Secret Store Tool : Version 12.2.0.1.0
Copyright (c) 2004, 2016, Oracle and/or its affiliates. All rights reserved.
Enter wallet password:
ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 = AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3pJC+UGD/
PdtE2jLIcBQcAeHgJChQGLA==
```

Digite a senha da carteira para ver o resultado.

Aqui, o valor à direita de `'='` é a senha da TDE.

5. Especifique o nome da chave de criptografia da TDE para o endpoint de origem do Oracle definindo o atributo de conexão adicional `securityDbEncryptionName`.

```
securityDbEncryptionName=ORACLE.SECURITY.DB.ENCRYPTION.AWGDC9glSk8Xv
+3bVveiVSgAAAAAAAAAAAAAAAAAAAAAAAAAA
```

6. Forneça a senha da TDE associada para essa chave no console como parte do valor da Senha da origem do Oracle. Utilize a ordem a seguir para formatar os valores de senha separados por vírgula, finalizados pelo valor da senha da TDE.

```
Oracle_db_password,ASM_Password,AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Especifique os valores de senha nessa ordem, independentemente da configuração do banco de dados Oracle. Por exemplo, se estiver utilizando a TDE, mas o banco de dados Oracle não estiver utilizando o ASM, especifique os valores de senha relevantes na ordem separada por vírgulas a seguir.

```
Oracle_db_password,,AEMAASAASGYs0phWHfNt9J5mEMkkegGFiD4LLfQszDojgDzbfoYDEACv0x3pJC
+UGD/PdtE2jLIcBQcAeHgJChQGLA==
```

Se as credenciais do TDE que você especificar estiverem incorretas, a tarefa de AWS DMS migração não falhará. No entanto, a tarefa também não lê nem aplica alterações contínuas de replicação ao banco de dados de destino. Depois de iniciar a tarefa, monitore as Estatísticas da tabela na página de tarefas de migração no console para garantir que as alterações sejam replicadas.

Se um DBA alterar os valores das credenciais da TDE para o banco de dados Oracle enquanto a tarefa estiver em execução, a tarefa falhará. A mensagem de erro contém o novo nome da chave de criptografia da TDE. Para especificar novos valores e reiniciar a tarefa, utilize o procedimento anterior.

> **Important**
>
> Não é possível manipular uma carteira da TDE criada em um local do Oracle Automatic Storage Management (ASM) porque comandos em nível do sistema operacional, como `cp`, `mv`, `orapki` e `mkstore`, corrompem os arquivos da carteira armazenados em um local do

ASM. Essa restrição é específica de arquivos de carteira da TDE armazenados somente em um local do ASM, mas não para arquivos de carteira da TDE armazenados em um diretório local do sistema operacional.
Para manipular uma carteira da TDE armazenada no ASM com comandos em nível do sistema operacional, crie um keystore local e mescle o keystore do ASM no keystore local da seguinte forma:

1. Crie um keystore local.

```
ADMINISTER KEY MANAGEMENT create keystore file system wallet location
 identified by wallet password;
```

2. Mescle o keystore do ASM com o keystore local.

```
ADMINISTER KEY MANAGEMENT merge keystore ASM wallet location identified
 by wallet password into existing keystore file system wallet location
 identified by wallet password with backup;
```

Para listar as entradas da carteira de criptografia e a senha da TDE, execute as etapas 3 e 4 no keystore local.

Métodos de compactação suportados para usar o Oracle como fonte para AWS DMS

Na tabela a seguir, você pode descobrir quais métodos de compactação são AWS DMS compatíveis ao trabalhar com um banco de dados de origem Oracle. Como mostra a tabela, o suporte à compactação depende da versão do seu banco de dados Oracle e se o DMS está configurado para usar o Oracle LogMiner para acessar os redo logs.

Version (Versão)	Basic	OLTP	HCC (do Oracle 11g R2 ou mais recente)	Outros
Oracle 10	Não	N/D	N/D	Não

Version (Versão)	Basic	OLTP	HCC (do Oracle 11g R2 ou mais recente)	Outros
Oracle 11 ou mais recente — Oracle LogMiner	Sim	Sim	Sim	Sim — Qualquer método de compactação suportado pela Oracle LogMiner.
Oracle 11 ou mais recente: Binary Reader	Sim	Sim	Sim, para obter mais informaçõ es, consulte a observação a seguir.	Sim

> **Note**
>
> Quando o endpoint de origem Oracle é configurado para utilizar o Binary Reader, o nível Query Low do método de compactação HCC tem suporte somente para tarefas de carga máxima.

Replicando tabelas aninhadas usando o Oracle como fonte para AWS DMS

AWS DMS suporta a replicação de tabelas Oracle contendo colunas que são tabelas aninhadas ou tipos definidos. Para ativar essa funcionalidade, adicione a seguinte configuração do atributo de conexão adicional ao endpoint de origem Oracle.

```
allowSelectNestedTables=true;
```

AWS DMS cria as tabelas de destino a partir de tabelas aninhadas da Oracle como tabelas pai e filho regulares no destino sem uma restrição exclusiva. Para acessar os dados corretos no destino, junte as tabelas pai e filho. Para fazer isso, primeiro crie manualmente um índice não exclusivo na coluna `NESTED_TABLE_ID` na tabela filho de destino. É possível utilizar a coluna `NESTED_TABLE_ID` na cláusula de junção `ON` juntamente com a coluna pai que corresponde ao nome da tabela filho. Além disso, a criação desse índice melhora o desempenho quando os dados da tabela secundária de

destino são atualizados ou excluídos pelo AWS DMS. Para ver um exemplo, consulte Exemplo de junção para tabelas pai e filho no destino.

É recomendável configurar a tarefa para ser interrompida após a conclusão de uma carga máxima. Crie esses índices não exclusivos para todas as tabelas filho replicadas no destino e retome a tarefa.

Se uma tabela aninhada capturada for adicionada a uma tabela principal existente (capturada ou não capturada), ela AWS DMS será tratada corretamente. No entanto, o índice não exclusivo para a tabela de destino correspondente não é criado. Nesse caso, se a tabela filho de destino se tornar extremamente grande, o desempenho pode ser afetado. Nesse caso, é recomendável interromper a tarefa, criar o índice e retomar a tarefa.

Depois que as tabelas aninhadas forem replicadas para o destino, solicite que o administrador de banco de dados execute uma junção nas tabelas pai e filho correspondentes para nivelar os dados.

Pré-requisitos para replicação de tabelas aninhadas Oracle como origem

Replique tabelas pai para todas as tabelas aninhadas replicadas. Inclua as tabelas principais (as tabelas que contêm a coluna da tabela aninhada) e as tabelas secundárias (ou seja, aninhadas) nos mapeamentos da AWS DMS tabela.

Tipos de tabela aninhada Oracle com suporte como origem

AWS DMS suporta os seguintes tipos de tabela aninhada Oracle como fonte:

- Tipo de dados
- Objeto definido pelo usuário

Limitações de suporte do AWS DMS para tabelas aninhadas Oracle como origem

AWS DMS tem as seguintes limitações em seu suporte às tabelas aninhadas da Oracle como fonte:

- AWS DMS suporta somente um nível de aninhamento de tabelas.
- AWS DMS o mapeamento de tabelas não verifica se a tabela ou tabelas principal e secundária estão selecionadas para replicação. Ou seja, é possível selecionar uma tabela pai sem uma tabela filho ou uma tabela filho sem pai.

Como o AWS DMS replica tabelas aninhadas Oracle como origem

AWS DMS replica tabelas principais e aninhadas para o destino da seguinte forma:

- AWS DMS cria a tabela principal idêntica à fonte. Ele define a coluna aninhada no pai como `RAW(16)` e inclui uma referência às tabelas aninhadas do pai em sua coluna `NESTED_TABLE_ID`.
- AWS DMS cria a tabela secundária idêntica à fonte aninhada, mas com uma coluna adicional chamada`NESTED_TABLE_ID`. Essa coluna tem o mesmo tipo e valor que a coluna aninhada pai correspondente e tem o mesmo significado.

Exemplo de junção para tabelas pai e filho no destino

Para nivelar a tabela pai, execute uma junção entre as tabelas pai e filho, conforme mostrado no exemplo a seguir:

1. Crie a tabela de `Type`.

```
CREATE OR REPLACE TYPE NESTED_TEST_T AS TABLE OF VARCHAR(50);
```

2. Crie a tabela pai com uma coluna do tipo `NESTED_TEST_T`, conforme definido anteriormente.

```
CREATE TABLE NESTED_PARENT_TEST (ID NUMBER(10,0) PRIMARY KEY, NAME NESTED_TEST_T)
 NESTED TABLE NAME STORE AS NAME_KEY;
```

3. Nivele a tabela `NESTED_PARENT_TEST` utilizando uma junção com a tabela filho `NAME_KEY` em que `CHILD.NESTED_TABLE_ID` corresponde a `PARENT.NAME`.

```
SELECT … FROM NESTED_PARENT_TEST PARENT, NAME_KEY CHILD WHERE CHILD.NESTED_
TABLE_ID = PARENT.NAME;
```

Armazenando REDO no Oracle ASM ao usar o Oracle como fonte para AWS DMS

Para origens Oracle com alta geração de REDO, o armazenamento de REDO no Oracle ASM pode beneficiar o desempenho, especialmente em uma configuração RAC, pois é possível configurar o DMS para distribuir leituras de REDO do ASM em todos os nós do ASM.

Para utilizar essa configuração, utilize o atributo de conexão `asmServer`. Por exemplo, a seguinte string de conexão distribui leituras de REDO do DMS em três nós do ASM:

```
asmServer=(DESCRIPTION=(CONNECT_TIMEOUT=8)(ENABLE=BROKEN)(LOAD_BALANCE=ON)(FAILOVER=ON)
(ADDRESS_LIST=
(ADDRESS=(PROTOCOL=tcp)(HOST=asm_node1_ip_address)(PORT=asm_node1_port_number))
(ADDRESS=(PROTOCOL=tcp)(HOST=asm_node2_ip_address)(PORT=asm_node2_port_number))
```

```
(ADDRESS=(PROTOCOL=tcp)(HOST=asm_node3_ip_address)(PORT=asm_node3_port_number)))
(CONNECT_DATA=(SERVICE_NAME=+ASM)))
```

Ao utilizar o NFS para armazenar o REDO do Oracle, é importante garantir que os patches de cliente DNFS (Direct NFS) aplicáveis sejam aplicados, especificamente qualquer patch que resolva o erro 25224242 do Oracle. Para obter informações adicionais, analise a seguinte publicação do Oracle sobre os patches relacionados ao cliente Direct NFS, Patches recomendados para o cliente Direct NFS.

Além disso, para melhorar o desempenho de leitura do NFS, é recomendável aumentar o valor de `rsize` e `wsize` em `fstab` para o volume NFS, conforme mostrado no exemplo a seguir.

```
NAS_name_here:/ora_DATA1_archive /u09/oradata/DATA1 nfs
 rw,bg,hard,nointr,tcp,nfsvers=3,_netdev,
timeo=600,rsize=262144,wsize=262144
```

Além disso, ajuste o valor de `tcp-max-xfer-size` da seguinte forma:

```
vserver nfs modify -vserver vserver -tcp-max-xfer-size 262144
```

Configurações de endpoint ao usar o Oracle como fonte para AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados de origem Oracle de forma semelhante à utilização de atributos de conexão adicionais. Você especifica as configurações ao criar o endpoint de origem usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint que podem ser utilizadas com o Oracle como origem.

Nome	Descrição
`AccessAlternateDirectly`	Defina esse atributo como falso para utilizar o Binary Reader para a captura de dados de alteração para um Amazon RDS para Oracle como origem. Isso informa a instância do DMS para não acessar logs de redo por meio de qualquer substituição de prefixo de caminho especificado utilizando o acesso direto a arquivos. Para ter mais informações, consulte

Nome	Descrição
	Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS. Valor padrão: verdadeiro Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"AccessAlternateDirectly": false}'`
`AdditionalArchivedLogDestId`	Defina esse atributo com `ArchivedLogDestId` em uma configuração primária em espera. Essa configuração é útil em uma transição quando o banco de dados Oracle Data Guard é utilizado como origem. Nesse caso, AWS DMS precisa saber de qual destino obter os redo logs de arquivamento para ler as alterações. Isso é porque a primária anterior agora é uma instância em espera depois da transição. Embora AWS DMS ofereça suporte ao uso da `RESETLOGS` opção Oracle para abrir o banco de dados, nunca use `RESETLOGS` a menos que seja necessário. Para obter informações adicionais sobre `RESETLOGS`, consulte Conceitos de reparo do RMAN no Guia do usuário de backup e recuperação do banco de dados Oracle®. Valores válidos: Ids de destino de arquivamento Exemplo: `--oracle-settings '{"AdditionalArchivedLogDestId": 2}'`

Nome	Descrição
`AddSupplementalLogging`	Defina este atributo para configurar a criação de logs complementares no nível da tabela para o banco de dados Oracle. Esse atributo habilita uma das seguintes opções em todas as tabelas selecionadas para uma tarefa de migração, dependendo dos respectivos metadados: • Registro em log suplementar de COLUNAS DE CHAVE PRIMÁRIA • Registro em log suplementar de COLUNAS DE CHAVE EXCLUSIVA • Registro em log suplementar de TODAS AS COLUNAS Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"AddSupplementalLogging": false}'` **Note** Se você utilizar essa opção, ainda precisará ativar a criação de registro em log suplementar no nível do banco de dados, conforme discutido anteriormente.
`AllowSelectNestedTables`	Defina esse atributo como true para permitir a replicação de tabelas Oracle com colunas que são tabelas aninhadas ou tipos definidos. Para ter mais informações, consulte Replicando tabelas aninhadas usando o Oracle como fonte para AWS DMS. Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"AllowSelectNestedTables": true}'`

Nome	Descrição
`ArchivedLogDestId`	Especifica o ID do destino para os logs redo de restauração arquivados. Esse valor deve ser o mesmo que um número na coluna dest_id da visualização v$archived_log. Se você trabalhar com um destino de redo log adicional, é recomendável utilizar o atributo `AdditionalArchivedLogDestId` para especificar o ID de destino adicional. Fazer isso aprimora o desempenho ao garantir que os logs corretos sejam acessados no início. Valor padrão: 1 Valores válidos: número Exemplo: `--oracle-settings '{"ArchivedLogDestId": 1}'`
`ArchivedLogsOnly`	Quando esse campo é definido como Y, AWS DMS só acessa os redo logs arquivados. Se os redo logs arquivados forem armazenados somente no Oracle ASM, a conta do AWS DMS usuário precisará receber privilégios de ASM. Valor padrão: N Valores válidos: Y/N Exemplo: `--oracle-settings '{"ArchivedLogsOnly": Y}'`

Nome	Descrição
asmUsePLSQLArray (Somente ECA)	Use esse atributo de conexão extra (ECA) ao capturar alterações na fonte com BinaryReader. Essa configuração permite que o DMS armazene 50 leituras em nível do ASM por thread de leitura único ao controlar o número de threads utilizando o atributo parallelASMReadThread . Quando você define esse atributo, o leitor AWS DMS binário usa um bloco PL/SQL anônimo para capturar dados de redo e enviá-los de volta para a instância de replicação como um grande buffer. Isso reduz o número de viagens de ida e volta até a origem. Isso pode melhorar significativamente o desempenho da captura de origem, mas resulta em consumo de memória mais alto de PGA na instância do ASM. Poderão surgir problemas de estabilidade se o destino da memória não for suficiente. É possível utilizar a fórmula a seguir para estimar a utilização total da memória PGA da instância do ASM por uma única tarefa do DMS: number_of_redo_threads * parallelASMReadThreads * 7 MB Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo de ECA: asmUsePLSQLArray=true;
ConvertTimestampWithZoneToUTC	Defina esse atributo como true para converter o valor do timestamp das colunas 'TIMESTAMP WITH TIME ZONE' e 'TIMESTAMP WITH LOCAL TIME ZONE' em UTC. Por padrão, o valor desse atributo é "falso" e os dados serão replicados utilizando o fuso horário do banco de dados de origem. Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: --oracle-settings '{"ConvertTimestampWithZoneToUTC": true}'

Nome	Descrição
`EnableHomogenousPartitionOps`	Defina esse atributo como `true` ativar a replicação das operações DDL de partição e subpartição do Oracle para migração homogênea do Oracle. Observe que esse recurso e aprimoramento foram introduzidos na AWS DMS versão 3.4.7. Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"EnableHomogenousPartitionOps": true}'`
`EnableHomogenousTablespace`	Defina esse atributo para habilitar a replicação homogênea de tablespace e criar tabelas ou índices existentes no mesmo tablespace no destino. Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"EnableHomogenousTablespace": true}'`

Nome	Descrição
`EscapeCharacter`	Defina esse atributo como um caractere de escape. Esse caractere de escape permite que um único caractere curinga se comporte como um caractere normal em expressões de mapeamento de tabela. Para ter mais informações, consulte Curingas no mapeamento de tabela. Valor padrão: nulo Valores válidos: qualquer caractere que não seja um caractere curinga Exemplo: `--oracle-settings '{"EscapeCharacter": "#"}'` **Note** É possível utilizar `escapeCharacter` somente em nomes de tabelas. Ele não escapa caracteres dos nomes dos esquemas ou dos nomes das colunas.
`ExposeViews`	É possível extrair dados uma vez de uma visualização, mas não é possível utilizá-los para replicação contínua. Ao extrair dados de uma visualização, a visualização aparece como uma tabela no esquema de destino. Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"ExposeViews": true}'`

Nome	Descrição
`ExtraArchivedLogDestIds`	Especifica os IDs de mais um destino para um ou mais logs redo arquivados. Esses IDs são os valores da coluna dest_id na visualização v$archived_log. Use essa configuração com o atributo de conexão ArchivedLogDestId extra em uma primary-to-single configuração ou primary-to-multiple-standby configuração. Essa configuração é útil em uma transição quando você utiliza um banco de dados Oracle Data Guard como origem. Nesse caso, AWS DMS precisa de informações sobre de qual destino obter os redo logs arquivados para ler as alterações. AWS DMS precisa disso porque, após a transição, a primária anterior é uma instância em espera. Valores válidos: Ids de destino de arquivamento Exemplo: `--oracle-settings '{"ExtraArchivedLogDestIds": 1}'`
`FailTasksOnLobTruncation`	Quando definido como `true`, esse atributo faz com que a tarefa falhe, caso o tamanho real de uma coluna LOB seja superior ao `LobMaxSize` especificado. Se a tarefa for definida como modo LOB limitado e essa opção estiver definida como `true`, a tarefa falhará em vez de truncar os dados de LOB. Valor padrão: falso Valores válidos: booleano Exemplo: `--oracle-settings '{"FailTasksOnLobTruncation": true}'`

Nome	Descrição
`filterTransactionsOfUser` (Somente ECA)	Use esse atributo de conexão extra (ECA) para permitir que o DMS ignore transações de um usuário especificado ao replicar dados do Oracle durante o uso. LogMiner É possível passar valores de nome de usuário separados por vírgula, mas eles devem estar em letras MAIÚSCULAS. Exemplo de ECA: `filterTransactionsOfUser=` *`USERNAME`*`;`
`NumberDataTypeScale`	Especifica a escala de números. É possível selecionar um aumento da escala verticalmente para 38 ou selecionar -1 para FLOAT ou -2 para VARCHAR. Por padrão, o tipo de dados NUMBER é convertido para um valor com precisão 38 e escala 10. Valor padrão: 10 Valores válidos: de -2 a 38 (–2 para VARCHAR, -1 para FLOAT) Exemplo: `--oracle-settings '{"NumberDataTypeScale": 12}'` **Note** Selecione uma combinação de escala de precisão, -1 (FLOAT) ou -2 (VARCHAR). O DMS é compatível com qualquer combinação de escala de precisão compatível com o Oracle. Se a precisão for 39 ou superior, selecione -2 (VARCHAR). A NumberDataTypeScale configuração do banco de dados Oracle é usada somente para o tipo de dados NUMBER (sem a precisão explícita e a definição de escala).

Nome	Descrição
`OpenTransactionWindow`	Fornece o período em minutos para verificar se há transações abertas apenas para tarefas da CDC. **Note** Quando você define como `OpenTransactionWindow` 1 ou superior, o DMS usa `SCN_TO_TIMESTAMP` para converter valores de SCN em valores de timestamp. Devido às limitações do banco de dados Oracle, se você especificar um SCN muito antigo como ponto inicial do CDC, o SCN_TO_TIMESTAMP falhará com um `ORA-08181` erro e você não poderá iniciar tarefas somente do CDC. Valor padrão: 0 Valores válidos: um número inteiro de 0 a 240 Exemplo: `openTransactionWindow=15;`
`OraclePathPrefix`	Defina esse atributo de string como o valor exigido para usar o Binary Reader para capturar dados de alteração para um Amazon RDS for Oracle como origem. Esse valor especifica a raiz padrão do Oracle utilizada para acessar os logs redo. Para ter mais informações, consulte Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS. Valor padrão: nenhum Valor válido: /rdsdbdata/db/ORCL_A/ Exemplo: `--oracle-settings '{"OraclePathPrefix": " /rdsdbdata/db/ORCL_A/ "}'`

Nome	Descrição
`ParallelASMReadThreads`	Defina esse atributo para alterar o número de threads que o DMS configura para executar uma captura de dados de alteração (CDC) utilizando o Oracle Automatic Storage Management (ASM). É possível especificar um valor inteiro entre 2 (o padrão) e 8 (o máximo). Use esse atributo junto com o atributo `ReadAheadBlocks` . Para ter mais informações, consulte Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS. Valor padrão: 2 Valores válidos: um número inteiro de 2 a 8 Exemplo: `--oracle-settings '{"ParallelASMReadThreads": 6;}'`
`ReadAheadBlocks`	Defina esse atributo para alterar o número de blocos de leitura antecipad a que o DMS configura para executar a CDC utilizando o Oracle Automatic Storage Management (ASM) o armazenamento não ASM NAS. É possível especificar um valor inteiro entre 1000 (o padrão) e 200.000 (o máximo). Use esse atributo junto com o atributo `ParallelASMReadThreads` . Para ter mais informações, consulte Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS. Valor padrão: 1000 Valores válidos: um número inteiro de 1.000 a 200.000 Exemplo: `--oracle-settings '{"ReadAheadBlocks": 150000}'`

Nome	Descrição
`ReadTableSpaceName`	Quando definido como `true`, esse atributo é compatível com a replicação de espaço para tabela. Valor padrão: falso Valores válidos: booleano Exemplo: `--oracle-settings '{"ReadTableSpaceName": true}'`
`ReplacePathPrefix`	Defina esse atributo como true para usar o Binary Reader para capturar dados de alteração em um Amazon RDS for Oracle como a origem. Essa configuração informa a instância do DMS para substituir raiz padrão do Oracle pela configuração `UsePathPrefix` especificada para acessar os logs redo. Para ter mais informações, consulte Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS. Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"ReplacePathPrefix": true}'`
`RetryInterval`	Especifica o número de segundos que o sistema espera antes de enviar novamente uma consulta. Valor padrão: 5 Valores válidos: número a partir de 1 Exemplo: `--oracle-settings '{"RetryInterval": 6}'`

Nome	Descrição
`SecurityDbEncryptionName`	Especifica o nome de uma chave utilizada para a criptografia de dados transparente (TDE) das colunas e do espaço para tabela no banco de dados de origem Oracle. Para obter mais informações sobre como definir esse atributo e sua senha associada no endpoint de origem Oracle, consulte Métodos de criptografia suportados para usar o Oracle como fonte para AWS DMS. Valor padrão: "" Valores válidos: string Exemplo: `--oracle-settings '{"SecurityDbEncryptionName": "ORACLE.SECURITY.DB.ENCRYPTION.Adg8m2dhkU/0v/m5QUaaNJEAAAAAAAAAAAAAAAAAAAAAAAAAAA"}'`
`SpatialSdo2GeoJsonFunctionName`	Para origens Oracle versão 12.1 ou anterior migrando para destinos PostgreSQL, utilize esse atributo para converter SDO_GEOMETRY para o formato GEOJSON. Por padrão, AWS DMS chama a função `SDO2GEOJSON` personalizada que deve estar presente e acessível ao AWS DMS usuário. Ou é possível criar seu próprio perfil personalizado que imite a operação `SDOGEOJSON` e definir `SpatialSdo2GeoJsonFunctionName` para chamá-la. Valor padrão: SDO2GEOJSON Valores válidos: string Exemplo: `--oracle-settings '{"SpatialSdo2GeoJsonFunctionName": "myCustomSDO2GEOJSONFunction"}'`

Nome	Descrição
`StandbyDelayTime`	Utilize esse atributo para especificar um tempo em minutos para o atraso na sincronização de espera. Se a origem for um banco de dados em espera do Active Data Guard, utilize esse atributo para especificar o atraso de tempo entre os bancos de dados primário e em espera. Em AWS DMS, você pode criar uma tarefa do Oracle CDC que usa uma instância standby do Active Data Guard como fonte para replicar as alterações em andamento. Isso elimina a necessidade de se conectar a um banco de dados ativo que pode estar em produção. Valor padrão: 0 Valores válidos: número Exemplo: `--oracle-settings '{"StandbyDelayTime": 1}'` Observação: ao utilizar o DMS 3.4.6, 3.4.7 e superior, a utilização dessa configuração de conexão é opcional. Na versão mais recente do DMS 3.4.6 e na versão 3.4.7, *`dms_user`* deve ter a permissão `select` ativada em `V_$DATAGUARD_STATS` , permitindo que o DMS calcule o tempo de atraso em espera.
`UseAlternateFolderForOnline`	Defina esse atributo como true para usar o Binary Reader para capturar dados de alteração em um Amazon RDS for Oracle como a origem. Isso informa a instância do DMS para utilizar qualquer substituição do prefixo especificado para acessar todos os logs redo online, Para ter mais informações, consulte Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS. Valor padrão: falso Valores válidos: verdadeiro/falso Exemplo: `--oracle-settings '{"UseAlternateFolderForOnline": true}'`

Nome	Descrição
`UseBfile`	Defina esse atributo como Y para capturar dados de alterações utilizando o utilitário Binary Reader. Defina `UseLogminerReader` como N para definir esse atributo como Y. Para utilizar o Binary Reader com o Amazon RDS para Oracle como a origem, defina atributos adicionais. Para obter mais informações sobre essa configuração e sobre como utilizar o Oracle Automatic Storage Management (ASM), consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC. Observação: ao definir esse valor como um atributo de conexão adicional (ECA), os valores válidos são "S" e "N". Ao definir esse valor como uma configuração de endpoint, os valores válidos são `true` e `false`. Valor padrão: N Valores válidos: S/N (ao definir esse valor como ECA); verdadeiro/falso (ao definir esse valor como uma configuração de endpoint). Exemplo: `--oracle-settings '{"UseBfile": Y}'`
`UseLogminerReader`	Defina esse atributo como Y para capturar dados de alteração usando o LogMiner utilitário (o padrão). Defina essa opção como N para que o AWS DMS acesse os logs redo como arquivos binários. Ao definir essa opção como N, adicione também a configuração useBfile=Y. Para obter mais informações sobre essa configuração e sobre a utilização do Oracle Automatic Storage Management (ASM), consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC. Observação: ao definir esse valor como um atributo de conexão adicional (ECA), os valores válidos são "S" e "N". Ao definir esse valor como uma configuração de endpoint, os valores válidos são `true` e `false`. Valor padrão: Y Valores válidos: S/N (ao definir esse valor como ECA); verdadeiro/falso (ao definir esse valor como uma configuração de endpoint). Exemplo: `--oracle-settings '{"UseLogminerReader": Y}'`

Nome	Descrição
`UsePathPrefix`	Defina esse atributo de string como o valor exigido para usar o Binary Reader para capturar dados de alteração para um Amazon RDS for Oracle como origem. Esse valor especifica o prefixo do caminho utilizado para substituir a raiz padrão do Oracle para acessar os redo logs. Para ter mais informações, consulte Configurando uma tarefa do CDC para usar o Binary Reader com uma fonte do RDS for Oracle para AWS DMS. Valor padrão: nenhum Valor válido: /rdsdbdata/log/ Exemplo: `--oracle-settings '{"UsePathPrefix": "  /rdsdbdata/log/ "}'`

Tipos de dados de origem do Oracle

O endpoint Oracle para AWS DMS suporta a maioria dos tipos de dados Oracle. A tabela a seguir mostra os tipos de dados de origem Oracle que são suportados durante o uso AWS DMS e o mapeamento padrão para AWS DMS os tipos de dados.

> **Note**
>
> Com exceção dos tipos de dados LONG e LONG RAW, ao replicar de uma origem Oracle para um destino Oracle (uma replicação homogênea), todos os tipos de dados de origem e de destino serão idênticos. Mas o tipo de dados LONG será mapeado para CLOB e o tipo de dados LONG RAW será mapeado para BLOB.

Para obter informações sobre como visualizar o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que você está usando.

Para obter informações adicionais sobre AWS DMS os tipos de dados, consulteTipos de dados do AWS Database Migration Service.

Tipo de dados do Oracle	AWS DMS tipo de dados
BINARY_FLOAT	REAL4
BINARY_DOUBLE	REAL8
BINARY	BYTES
FLOAT (P)	Se a precisão é menor ou igual a 24, utilize REAL4. Se a precisão for maior que 24, utilize REAL8.
NUMBER (P,S)	Quando a escala for maior que 0, utilize NUMERIC. Quando a escala for 0: • E a precisão for menor ou igual a 2, utilize INT1. • E a precisão for maior do que 2 e menor ou igual a 4, utilize INT2. • E a precisão for maior do que 4 e menor ou igual a 9, utilize INT4. • E a precisão for maior do que 9, utilize NUMERIC. • E a precisão for maior ou igual à escala, utilize NUMERIC. Quando a escala for menor que 0, utilize REAL8.
DATA	DATETIME
INTERVAL YEAR TO MONTH	STRING (com indicação de intervalo de tempo em anos e meses)
INTERVAL DAY TO SECOND	STRING (com indicação de intervalo de tempo em dias e segundos)
TIMESTAMP	DATETIME
TIMESTAMP WITH TIME ZONE	STRING (com indicação de time stamp com fuso horário)
TIMESTAMP WITH LOCAL TIME ZONE	STRING (com indicação de time stamp com fuso horário local)

Tipo de dados do Oracle	AWS DMS tipo de dados
CHAR	STRING
VARCHAR2	STRING
NCHAR	WSTRING
NVARCHAR2	WSTRING
RAW	BYTES
REAL	REAL8
BLOB	BLOB Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de tipos de dados BLOB para uma tarefa específica. AWS DMS suporta tipos de dados BLOB somente em tabelas que incluem uma chave primária.
CLOB	CLOB Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de tipos de dados CLOB para uma tarefa específica. Durante o CDC, AWS DMS suporta tipos de dados CLOB somente em tabelas que incluem uma chave primária.
NCLOB	NCLOB Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso dos tipos de dados NCLOB para uma tarefa específic a. Durante o CDC, AWS DMS suporta tipos de dados NCLOB somente em tabelas que incluem uma chave primária.

Tipo de dados do Oracle	AWS DMS tipo de dados
LONG	CLOB O tipo de dados LONG não é suportado no modo de aplicação otimizado em lote (modo TurboStream CDC). Para usar esse tipo de dados com AWS DMS, habilite o uso de LOBs para uma tarefa específica. Durante o CDC ou com carga total, AWS DMS suporta tipos de dados LOB somente em tabelas que tenham uma chave primária. Além disso, AWS DMS não suporta o modo LOB completo para carregar colunas LONG. Em vez disso, é possível utilizar o modo LOB limitado para migrar colunas LONG para um destino Oracle. No modo LOB limitado, AWS DMS trunca todos os dados de 64 KB definidos como colunas LONG maiores que 64 KB. Para obter mais informações sobre o suporte a LOB em AWS DMS, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS

Tipo de dados do Oracle	AWS DMS tipo de dados
LONG RAW	BLOB O tipo de dados LONG RAW não é suportado no modo de aplicação otimizado em lote (modo TurboStream CDC). Para usar esse tipo de dados com AWS DMS, habilite o uso de LOBs para uma tarefa específica. Durante o CDC ou com carga total, AWS DMS suporta tipos de dados LOB somente em tabelas que tenham uma chave primária. Além disso, AWS DMS não suporta o modo LOB completo para carregar colunas LONG RAW. Em vez disso, é possível utilizar o modo LOB limitado para migrar colunas LONG RAW para um destino Oracle. No modo LOB limitado, AWS DMS trunca todos os dados de 64 KB definidos como colunas LONG RAW com mais de 64 KB. Para obter mais informações sobre o suporte a LOB em AWS DMS, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS
XMLTYPE	CLOB
SDO_GEOMETRY	BLOB (quando é uma migração de Oracle para Oracle) CLOB (quando é uma migração de Oracle para PostgreSQL)

As tabelas do Oracle utilizado como origem com colunas dos tipos de dados a seguir não são compatíveis e não podem ser replicadas. A replicação de colunas com esses tipos de dados resultará em colunas nulas.

- BFILE
- ROWID
- REF
- UROWID
- Tipos de dados definidos pelo usuário

- ANYDATA
- VARRAY

> **Note**
>
> Colunas virtuais não são compatíveis.

Migrar tipos de dados espaciais do Oracle

Dados espaciais identificam as informações de geometria de um objeto ou local no espaço. Em um banco de dados Oracle, a descrição geométrica de um objeto geográfico é armazenada em um objeto do tipo SDO_GEOMETRY. Dentro desse objeto, a descrição geométrica é armazenada em uma única linha em uma única coluna de uma tabela definida pelo usuário.

AWS DMS suporta a migração do tipo Oracle SDO_GEOMETRY de uma fonte Oracle para um destino Oracle ou PostgreSQL.

Ao migrar tipos de dados espaciais Oracle usando AWS DMS, esteja ciente destas considerações:

- Ao migrar para um destino Oracle, transfira manualmente as entradas USER_SDO_GEOM_METADATA que incluem informações de tipo.
- Ao migrar de um endpoint de origem Oracle para um endpoint de destino do PostgreSQL, cria colunas de destino. AWS DMS Essas colunas têm geometria padrão e informações de tipo geográfico com uma dimensão 2D e um identificador de referência espacial (SRID) igual a zero (0). Um exemplo é `GEOMETRY, 2, 0`.
- Para origens Oracle versão 12.1 ou anterior migrando para destinos PostgreSQL, converta os objetos `SDO_GEOMETRY` para o formato `GEOJSON` utilizando o perfil `SDO2GEOJSON` ou o atributo de conexão adicional `spatialSdo2GeoJsonFunctionName`. Para ter mais informações, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.
- AWS DMS suporta migrações de colunas espaciais da Oracle somente para o modo LOB completo. AWS DMS não suporta os modos LOB limitado ou LOB embutido. Para obter mais informações sobre o modo LOB, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS.
- Como AWS DMS só oferece suporte ao modo LOB completo para migrar colunas espaciais do Oracle, a tabela das colunas precisa de uma chave primária e uma chave exclusiva. Se a tabela não tiver uma chave primária e uma chave exclusiva, a tabela será ignorada na migração.

Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS

Migre dados de um ou vários bancos de dados do Microsoft SQL Server usando o. AWS DMS Com um banco de dados do SQL Server como fonte, você pode migrar dados para outro banco de dados do SQL Server ou para um dos outros bancos de dados AWS DMS compatíveis.

Para obter informações sobre as versões do SQL Server que oferecem AWS DMS suporte como fonte, consulteFontes para AWS DMS.

O banco de dados de origem SQL Server pode ser instalado em qualquer computador na rede. É necessário ter uma conta do SQL Server com os privilégios de acesso ao banco de dados de origem adequados ao tipo de tarefa escolhido para utilizar com o AWS DMS. Essa conta deve ter as permissões `view definition` e `view server state`. Adicione essa permissão utilizando o seguinte comando:

```
grant view definition to [user]
grant view server state to [user]
```

AWS DMS oferece suporte à migração de dados de instâncias nomeadas do SQL Server. É possível utilizar a seguinte notação no nome do servidor ao criar o endpoint de origem.

```
IPAddress\InstanceName
```

Por exemplo, o seguinte é um nome do servidor do endpoint de origem correto. Aqui, a primeira parte do nome é o endereço IP do servidor, e a segunda parte é o nome da instância do SQL Server (neste exemplo, SQLTest).

```
10.0.0.25\SQLTest
```

Além disso, obtenha o número da porta que sua instância nomeada do SQL Server escuta e use-o para configurar seu endpoint de AWS DMS origem.

> **Note**
>
> A porta 1433 é o padrão para o Microsoft SQL Server. No entanto, as portas dinâmicas que são alteradas a cada vez que o SQL Server é iniciado e os números de portas estáticas

específicos utilizados para conexão ao SQL Server por meio de um firewall também são utilizados com frequência. Então, você quer saber o número real da porta da sua instância nomeada do SQL Server ao criar o endpoint de AWS DMS origem.

É possível utilizar SSL para criptografar conexões entre o endpoint do SQL Server e a instância de replicação. Para obter mais informações sobre como utilizar o SSL com um endpoint do SQL Server, consulte Usando SSL com AWS Database Migration Service.

Para obter detalhes adicionais sobre como trabalhar com bancos de dados de origem do SQL Server e AWS DMS, consulte o seguinte.

Tópicos

- Limitações no uso do SQL Server como fonte para AWS DMS
- Permissões para tarefas somente de carga máxima
- Utilizar replicação contínua (CDC) a partir de uma origem do SQL Server
- Capturar dados alterados no SQL Server autogerenciado on-premises ou no Amazon EC2
- Configurar a replicação contínua em uma instância de banco de dados SQL Server na nuvem
- Configurações recomendadas ao usar o Amazon RDS for SQL Server como fonte para AWS DMS
- Métodos de compactação compatíveis com o SQL Server
- Trabalhando com grupos de AlwaysOn disponibilidade autogerenciados do SQL Server
- Requisitos de segurança ao usar o SQL Server como fonte para AWS Database Migration Service
- Configurações de endpoint ao usar o SQL Server como fonte para AWS DMS
- Tipos de dados de origem no SQL Server

Limitações no uso do SQL Server como fonte para AWS DMS

As seguintes limitações se aplicam ao utilizar um banco de dados SQL como origem do AWS DMS:

- A propriedade identity de uma coluna não é migrada para uma coluna de banco de dados de destino.
- O endpoint do SQL Server não oferece suporte ao uso de tabelas com colunas esparsas.
- A Autenticação do Windows não é compatível.
- As alterações em campos computados em um SQL Server não são replicadas.

- Tabelas temporais não são compatíveis.
- A alternância de partições do SQL Server não é compatível.
- Ao usar os utilitários WRITETEXT e UPDATETEXT, AWS DMS não captura eventos aplicados no banco de dados de origem.
- O seguinte padrão da linguagem de manipulação de dados (DML) não é compatível:

```
SELECT * INTO new_table FROM existing_table
```

- Ao utilizar o SQL Server como uma origem, a criptografia em nível de colunas não é compatível.
- AWS DMS não oferece suporte a auditorias em nível de servidor no SQL Server 2008 ou no SQL Server 2008 R2 como fontes. Isso ocorre devido a um problema conhecido com o SQL Server 2008 e 2008 R2. Por exemplo, executar o comando AWS DMS a seguir causa falha.

```
USE [master]
GO
ALTER SERVER AUDIT [my_audit_test-20140710] WITH (STATE=on)
GO
```

- As colunas de geometria não são compatíveis no modo LOB completo ao utilizar o SQL Server como origem. Em vez disso, utilize o modo LOB limitado ou defina a configuração da tarefa `InlineLobMaxSize` para utilizar o modo LOB em linha.
- Ao utilizar um banco de dados de origem Microsoft SQL Server em uma tarefa de replicação, as definições do publicador de replicação do SQL Server não serão removidas se você remover a tarefa. Um administrador de sistema do Microsoft SQL Server deve excluir essas definições do Microsoft SQL Server.
- A migração de dados de non-schema-bound visualizações e vinculados a esquemas é suportada somente para tarefas de carga total.
- A renomeação de tabelas não é compatível com a utilização de sp_rename (por exemplo, `sp_rename 'Sales.SalesRegion', 'SalesReg;`))
- A renomeação de colunas não é compatível com a utilização de sp_rename (por exemplo, `sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';`)
- AWS DMS não oferece suporte ao processamento de alterações para definir e desdefinir valores padrão da coluna (usando a `ALTER COLUMN SET DEFAULT` cláusula com `ALTER TABLE` instruções).
- AWS DMS não oferece suporte ao processamento de alterações para definir a nulidade da coluna (usando a `ALTER COLUMN [SET|DROP] NOT NULL` cláusula com instruções). `ALTER TABLE`

- Com o SQL Server 2012 e o SQL Server 2014, ao utilizar a replicação do DMS com grupos de disponibilidade, o banco de dados de distribuição não pode ser colocado em um grupo de disponibilidade. O SQL 2016 oferece suporte à colocação do banco de dados de distribuição em um grupo de disponibilidade, exceto para bancos de dados de distribuição usados em topologias de mesclagem, bidirecional ou peer-to-peer replicação.
- Para tabelas particionadas, AWS DMS não oferece suporte a diferentes configurações de compactação de dados para cada partição.
- Ao inserir um valor em tipos de dados espaciais (GEOGRAPHY e GEOMETRY) no SQL Server, é possível ignorar a propriedade de identificador de sistema de referência espacial (SRID) ou especificar outro número. Ao replicar tabelas com tipos de dados espaciais, AWS DMS substitui o SRID pelo SRID padrão (0 para GEOMETRIA e 4326 para GEOGRAFIA).
- Se o banco de dados não estiver configurado para MS-REPLICATION ou MS-CDC, ainda será possível capturar tabelas que não tenham uma chave primária, mas somente eventos INSERT/DELETE DML serão capturados. Os eventos UPDATE e TRUNCATE TABLE são ignorados.
- Os índices Columnstore não são compatíveis.
- Tabelas otimizadas para memória (utilizando OLTP na memória) não são compatíveis.
- Ao replicar uma tabela com uma chave primária que consiste em várias colunas, a atualização das colunas de chave primária durante a carga máxima não é compatível.
- A durabilidade atrasada não é compatível.
- A configuração do endpoint `readBackup0nly=Y` (atributo de conexão adicional) não funciona em instâncias de origem do RDS para SQL Server devido à forma como o RDS executa backups.
- O `EXCLUSIVE_AUTOMATIC_TRUNCATION` não funciona em instâncias de origem do Amazon RDS SQL Server porque os usuários do RDS não têm acesso para executar o procedimento armazenado do SQL Server, `sp_repldone`.
- AWS DMS não captura comandos truncados.
- AWS DMS não oferece suporte à replicação de bancos de dados com a recuperação acelerada de banco de dados (ADR) ativada.
- AWS DMS não suporta a captura de instruções de linguagem de definição de dados (DDL) e linguagem de manipulação de dados (DML) em uma única transação.
- AWS DMS não oferece suporte à replicação de pacotes de aplicativos de camada de dados (DACPAC).
- As instruções UPDATE que envolvem chaves primárias ou índices exclusivos e atualizam várias linhas de dados podem causar conflitos ao aplicar alterações no banco de dados de destino.

Isso pode acontecer, por exemplo, quando o banco de dados de destino aplica atualizações, como instruções INSERT e DELETE, em vez de uma única instrução UPDATE. Com o modo de aplicação otimizado em lote, a tabela pode ser ignorada. Com o modo de aplicação transacional, a operação UPDATE pode resultar em violações de restrições. Para evitar esse problema, recarregue a tabela relevante. Como alternativa, localize os registros problemáticos na tabela de controle Exceções da aplicação (`dmslogs.awsdms_apply_exceptions`) e edite-os manualmente no banco de dados de destino. Para ter mais informações, consulte Configurações de ajuste de processamento de alterações.

- AWS DMS não suporta a replicação de tabelas e esquemas, em que o nome inclui um caractere especial do conjunto a seguir.

  ```
  \\ -- \n \" \b \r ' \t ;
  ```

- O mascaramento de dados não é suportado. AWS DMS migra dados mascarados sem mascarar.
- AWS DMS replica até 32.767 tabelas com chaves primárias e até 1.000 colunas para cada tabela. Isso ocorre porque AWS DMS cria um artigo de replicação do SQL Server para cada tabela replicada, e os artigos de replicação do SQL Server têm essas limitações.
- Ao utilizar a captura de dados de alteração (CDC), defina todas as colunas que compõem um índice exclusivo como `NOT NULL`. Se esse requisito não for atendido, ocorrerá o erro 22838 do sistema do SQL Server.

As seguintes limitações se aplicam ao acessar os logs de transação de backup:

- Backups criptografados não são compatíveis.
- Backups armazenados em um URL ou no Windows Azure não são compatíveis.
- AWS DMS não oferece suporte ao processamento direto de backups de registros de transações no nível do arquivo a partir de pastas compartilhadas alternativas.

Permissões para tarefas somente de carga máxima

As permissões a seguir são necessárias para realizar tarefas somente de carga máxima. Observe que AWS DMS isso não cria o `dms_user` login. Para obter informações sobre como criar um login para o SQL Server, consulte Criar um usuário de banco de dados com o Microsoft SQL Server.

```
USE db_name;

                CREATE USER dms_user FOR LOGIN dms_user;
```

```
ALTER ROLE [db_datareader] ADD MEMBER dms_user;
GRANT VIEW DATABASE STATE to dms_user ;

USE master;

GRANT VIEW SERVER STATE TO dms_user;
```

Utilizar replicação contínua (CDC) a partir de uma origem do SQL Server

É possível utilizar a replicação contínua (captura de dados de alteração, ou CDC) para um banco de dados SQL Server autogerenciado on-premises ou no Amazon EC2, ou um banco de dados de nuvem, como o Amazon RDS ou uma instância gerenciada pelo Microsoft Azure SQL.

Os seguintes requisitos se aplicam especificamente ao utilizar a replicação contínua com um banco de dados SQL Server como uma origem para o AWS DMS:

- O SQL Server deve ser configurado para fazer backups completos e um backup deve ser feito antes do início da replicação dos dados.
- O modelo de recuperação deve ser definido como Bulk-logged ou Full.
- O backup do SQL Server para múltiplos discos não é compatível. Se o backup estiver definido para gravar o backup do banco de dados em vários arquivos em discos diferentes, não será AWS DMS possível ler os dados e a AWS DMS tarefa falhará.
- Para origens do SQL Server autogerenciadas, as definições do SQL Server Replication Publisher para a origem utilizada em uma tarefa de CDC do DMS não são removidas quando você remove a tarefa. Um administrador de sistema do SQL Server deve excluir essas definições do SQL Server para origens autogerenciadas.
- Durante o CDC, é AWS DMS necessário consultar os backups do log de transações do SQL Server para ler as alterações. AWS DMS não oferece suporte a backups de log de transações do SQL Server criados usando software de backup de terceiros que não estejam em formato nativo. Para compatibilidade com backups de logs de transações que estão em formato nativo e foram criados utilizando software de backup de terceiros, adicione o atributo de conexão `use3rdPartyBackupDevice=Y` ao endpoint de origem.
- Para origens autogerenciadas do SQL Server, lembre-se de que o SQL Server não captura alterações em tabelas recém-criadas até que elas sejam publicadas. Quando as tabelas são adicionadas a uma fonte do SQL Server, AWS DMS gerencia a criação da publicação. No entanto, esse processo pode demorar alguns minutos. As operações feitas durante esse intervalo nas tabelas recentemente criadas não são capturadas ou replicadas no destino.

- AWS DMS a captura de dados de alteração exige que o registro completo de transações seja ativado no SQL Server. Para ativar o registro em log de transações completo no SQL Server, ative MS-REPLICATION ou CHANGE DATA CAPTURE (CDC).
- As entradas tlog do SQL Server não serão marcadas para reutilização até que o trabalho de captura de MS CDC processe essas alterações.
- As operações de CDC não são compatíveis com tabelas com otimização de memória. Essa limitação se aplica ao SQL Server 2014 (quando o recurso foi introduzido pela primeira vez) e posterior.
- AWS DMS a captura de dados de alteração requer, por padrão, um banco de dados de distribuição no Amazon EC2 ou no servidor SQL On-Prem como fonte. Portanto, ative o distribuidor ao configurar a replicação de MS para tabelas com chaves primárias.

Capturar dados alterados no SQL Server autogerenciado on-premises ou no Amazon EC2

Para capturar as alterações de um banco de dados de origem do Microsoft SQL Server, verifique se o banco de dados está configurado para backups completos. Configure o banco de dados no modo de recuperação total ou no modo de registro em log em massa.

Para uma fonte autogerenciada do SQL Server, AWS DMS use o seguinte:

Replicação de MS

Para capturar alterações em tabelas com chaves primárias. Você pode configurar isso automaticamente concedendo privilégios de administrador de sistema ao usuário do AWS DMS endpoint na instância de origem do SQL Server. Ou você pode seguir as etapas desta seção para preparar a fonte e usar um usuário que não tenha privilégios de administrador de sistema para o endpoint. AWS DMS

MS-CDC

Para capturar alterações em tabelas sem chaves primárias. Ative a MS-CDC no nível do banco de dados e em todas as tabelas individualmente.

Ao configurar um banco de dados SQL Server para replicação contínua (CDC), é possível seguir um destes procedimentos:

- Configurar a replicação contínua utilizando o perfil sysadmin.

- Configurar a replicação contínua para não utilizar o perfil sysadmin.

Configurar a replicação contínua em um SQL Server autogerenciado

Esta seção contém informações sobre como configurar a replicação contínua em um servidor SQL autogerenciado com ou sem a utilização do perfil sysadmin.

Tópicos

- Configurar a replicação contínua em um SQL Server autogerenciado: utilizando o perfil sysadmin
- Configurar a replicação contínua em um SQL Server autônomo: sem o perfil sysadmin

Configurar a replicação contínua em um SQL Server autogerenciado: utilizando o perfil sysadmin

AWS DMS a replicação contínua para o SQL Server usa a replicação nativa do SQL Server para tabelas com chaves primárias e a captura de dados alterados (CDC) para tabelas sem chaves primárias.

Antes de configurar a replicação contínua, consulte Utilizar replicação contínua (CDC) a partir de uma origem do SQL Server.

Para tabelas com chaves primárias, geralmente é AWS DMS possível configurar os artefatos necessários na fonte. No entanto, para instâncias de origem do SQL Server que são autogerenciadas, configure primeiro a distribuição do SQL Server manualmente. Depois de fazer isso, os usuários de AWS DMS origem com permissão de administrador de sistema podem criar automaticamente a publicação para tabelas com chaves primárias.

Para verificar se a distribuição já foi configurada, execute o comando a seguir.

```
sp_get_distributor
```

Se o resultado for `NULL` para a distribuição de colunas, a distribuição não estará configurada. É possível utilizar o procedimento a seguir para configurar a distribuição.

Como configurar a distribuição

1. Conecte-se ao banco de dados de origem do SQL Server utilizando a ferramenta SQL Server Management Studio (SSMS).
2. Abra o menu de contexto (clique com o botão direito do mouse) da pasta Replicação e escolha Configurar distribuição. O assistente de configuração da distribuição é exibido.

3. Siga o assistente para inserir os valores padrão e criar a distribuição.

Como configurar a CDC

AWS DMS a versão 3.4.7 e superior pode configurar o MS CDC para seu banco de dados e todas as suas tabelas automaticamente se você não estiver usando uma réplica somente para leitura. Para utilizar esse recurso, defina o ECA `SetUpMsCdcForTables` como verdadeiro. Para obter mais informações sobre ECAs, consulte Configurações de endpoint.

Para versões AWS DMS anteriores à 3.4.7 ou para uma réplica somente leitura como fonte, execute as seguintes etapas:

1. Para tabelas sem chaves primárias, configure a MS-CDC para o banco de dados. Para fazer isso, utilize uma conta que tenha o perfil sysadmin atribuído a ela e execute o comando a seguir.

```
use [DBname]
EXEC sys.sp_cdc_enable_db
```

2. Configure a MS-CDC para cada uma das tabelas de origem. Para cada tabela com chaves exclusivas, mas sem chave primária, execute a consulta a seguir para configurar a 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. Para cada tabela sem chave primária nem chaves exclusivas, execute a consulta a seguir para configurar a MS-CDC.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@role_name = NULL
GO
```

Para obter mais informações sobre como configurar a MS-CDC para tabelas específicas, consulte a Documentação do SQL Server.

Configurar a replicação contínua em um SQL Server autônomo: sem o perfil sysadmin

Para obter informações sobre como configurar a replicação contínua em um SQL Server autônomo sem o perfil sysadmin, consulte Configurar a replicação contínua em um SQL Server autônomo: sem o perfil sysadmin.

Configurar a replicação contínua em uma instância de banco de dados SQL Server na nuvem

Esta seção descreve como configurar a CDC em uma instância de banco de dados SQL Server hospedada na nuvem. Uma instância do SQL Server hospedada na nuvem é uma instância em execução no Amazon RDS para SQL Server, uma instância gerenciada do Azure SQL ou qualquer outra instância gerenciada do SQL Server na nuvem. Para obter informações sobre as limitações da replicação contínua para cada tipo de banco de dados, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Antes de configurar a replicação contínua, consulte Utilizar replicação contínua (CDC) a partir de uma origem do SQL Server.

Ao contrário das origens autogerenciadas do SQL Server, o Amazon RDS para SQL Server não é compatível com a MS-Replication. Portanto, AWS DMS precisa usar o MS-CDC para tabelas com ou sem chaves primárias.

O Amazon RDS não concede privilégios de administrador de sistema para definir artefatos de replicação AWS DMS usados para alterações contínuas em uma instância de origem do SQL Server. Ative a MS-CDC na instância do Amazon RDS (utilizando privilégios de usuário mestre) como no procedimento a seguir.

Para ativar a MS-CDC em uma instância de banco de dados do SQL Server na nuvem

1. Execute as consultas a seguir no nível do banco de dados.

   Para uma instância de banco de dados RDS para SQL Server, utilize essa consulta.

   ```
   exec msdb.dbo.rds_cdc_enable_db 'DB_name'
   ```

   Para uma instância de banco de dados gerenciada do Azure SQL, utilize essa consulta.

   ```
   USE DB_name
   GO
   ```

```
EXEC sys.sp_cdc_enable_db
GO
```

2. Para cada tabela com uma chave primária, execute a consulta a seguir para ativar a 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
```

Para cada tabela com chaves exclusivas, mas sem chave primária, execute a consulta a seguir para ativar a 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
```

Para cada tabela sem chave primária e sem chaves exclusivas, execute a consulta a seguir para habilitar a MS-CDC.

```
exec sys.sp_cdc_enable_table
@source_schema = N'schema_name',
@source_name = N'table_name',
@role_name = NULL
GO
```

3. Defina o período de retenção para que as alterações estejam disponíveis na origem utilizando o comando a seguir.

```
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'
```

O parâmetro `@pollinginterval` é medido em segundos com um valor recomendado definido como 86399. Isso significa que o log de transações retém as alterações por 86.399 segundos (um dia) quando `@pollinginterval = 86399`. O procedimento `exec sp_cdc_start_job 'capture'` inicia as configurações.

> **Note**
>
> Com algumas versões do SQL Server, se o valor de `pollinginterval` for definido como mais de 3599 segundos, o valor será redefinido para os cinco segundos padrão. Quando isso acontece, as entradas do T-Log são removidas antes que AWS DMS você possa lê-las. Para determinar quais versões do SQL Server são afetadas por esse problema conhecido, consulte este artigo da Microsoft KB.

Se estiver utilizando o Amazon RDS com multi-AZ, defina também o secundário para ter os valores corretos em caso de failover.

```
exec rdsadmin..rds_set_configuration 'cdc_capture_pollinginterval' , 86399
```

Se uma tarefa de AWS DMS replicação que captura alterações contínuas na origem do SQL Server parar por mais de uma hora, use o procedimento a seguir.

Para manter o período de retenção durante uma tarefa de AWS DMS replicação

1. Interrompa o trabalho que está truncando os logs de transações utilizando este comando:

   ```
   exec sp_cdc_stop_job 'capture'
   ```

2. Encontre sua tarefa no AWS DMS console e continue a tarefa.
3. Escolha a guia Monitoramento e marque a métrica `CDCLatencySource`.
4. Quando a métrica `CDCLatencySource` for igual a 0 (zero) e permanecer nesse valor, reinicie o trabalho truncando os logs de transações com o seguinte comando:

   ```
   exec sp_cdc_start_job 'capture'
   ```

Lembre-se de iniciar o trabalho que trunca os logs de transações do SQL Server. Caso contrário, o armazenamento na instância do SQL Server poderá ficar cheio.

Limitações da replicação contínua em uma instância do banco de dados SQL Server na nuvem

- AWS DMS suporta replicação contínua (CDC) somente com o log de transações ativo. Não é possível utilizar o log de backup com a CDC.
- É possível perder eventos se você os mover do log de transações ativo para o log de backup ou truncá-los no log de transações ativo.

Configurações recomendadas ao usar o Amazon RDS for SQL Server como fonte para AWS DMS

Quando você trabalha com o Amazon RDS para SQL Server como origem, o trabalho de captura depende dos parâmetros `maxscans` e `maxtrans`. Esses parâmetros governam o número máximo de verificações que a captura faz no log de transações e o número de transações que são processadas para cada verificação.

Para bancos de dados, em que o número de transações é maior que `maxtrans*maxscans`, aumentar o valor de `polling_interval` pode causar um acúmulo de registros no log de transações. Por sua vez, esse acúmulo pode levar a um aumento no tamanho do log de transações.

Observe que AWS DMS não depende da tarefa de captura do MS-CDC. A tarefa de captura da MS-CDC marca as entradas do log de transações como processadas. Isso permite que a tarefa de backup do log de transações remova as entradas do log de transações.

É recomendável monitorar o tamanho do log de transações e o sucesso das tarefas da MS-CDC. Se as tarefas do MS-CDC falharem, o registro de transações poderá crescer excessivamente e causar falhas na replicação. AWS DMS É possível monitorar erros do trabalho de captura da MS-CDC utilizando a visualização de gerenciamento dinâmico `sys.dm_cdc_errors` no banco de dados de origem. É possível monitorar o tamanho do log de transações utilizando o comando de gerenciamento `DBCC SQLPERF(LOGSPACE)`.

Como abordar o aumento do log de transações causado pela MS-CDC

1. Verifique de onde `Log Space Used %` o banco de dados AWS DMS está sendo replicado e valide se ele aumenta continuamente.

   ```
   DBCC SQLPERF(LOGSPACE)
   ```

2. Identifique o que está bloqueando o processo de backup do log de transações.

```
Select log_reuse_wait, log_reuse_wait_desc, name from sys.databases where name =
 db_name();
```

   Se o valor de `log_reuse_wait_desc` for igual a `REPLICATION`, a retenção do backup do log será causada pela latência na MS-CDC.

3. Aumente o número de eventos processados pelo trabalho de captura aumentando os valores dos parâmetros `maxtrans` e `maxscans`.

```
EXEC sys.sp_cdc_change_job @job_type = 'capture' ,@maxtrans = 5000, @maxscans = 20
exec sp_cdc_stop_job 'capture'
exec sp_cdc_start_job 'capture'
```

Para resolver esse problema, defina os valores de `maxscans` e para que `maxtrans` `maxtrans*maxscans` sejam iguais ao número médio de eventos gerados para tabelas que são AWS DMS replicadas do banco de dados de origem para cada dia.

Se você definir esses parâmetros acima do valor recomendado, os trabalhos de captura processarão todos os eventos nos logs de transações. Se você definir esses parâmetros abaixo do valor recomendado, a latência da MS-CDC aumentará e o log de transações também.

A identificação dos valores apropriados para `maxscans` e `maxtrans` pode ser difícil porque as alterações na workload produzem um número variável de eventos. Nesse caso, é recomendável configurar o monitoramento na latência da MS-CDC. Para obter mais informações, consulte Monitorar o processo na documentação do SQL Server. Configure `maxtrans` e `maxscans` dinamicamente com base nos resultados do monitoramento.

Se a AWS DMS tarefa não conseguir encontrar os números de sequência de log (LSNs) necessários para retomar ou continuar a tarefa, a tarefa poderá falhar e exigir uma recarga completa.

> **Note**
>
> Ao usar AWS DMS para replicar dados de uma fonte do RDS para SQL Server, você pode encontrar erros ao tentar retomar a replicação após um evento de stop-start da instância do Amazon RDS. Isso ocorre porque o processo do SQL Server Agent reinicia o processo da tarefa de captura quando ele é reiniciado após o evento de interrupção-inicialização. Isso ignora o intervalo de pesquisa da MS-CDC.

Por esse motivo, em bancos de dados com volumes de transações menores do que o processamento da tarefa de captura do MS-CDC, isso pode fazer com que os dados sejam processados ou marcados como replicados e copiados antes AWS DMS de serem retomados de onde pararam, resultando no seguinte erro:

```
[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)
```

Para mitigar esse problema, defina os valores de `maxtrans` e de `maxscans` conforme recomendado anteriormente.

Métodos de compactação compatíveis com o SQL Server

Observe o seguinte sobre os métodos de compactação compatíveis com o SQL Server no AWS DMS:

- AWS DMS oferece suporte à compactação de linha/página no SQL Server versão 2008 e posterior.
- AWS DMS não suporta o formato de armazenamento Vardecimal.
- AWS DMS não suporta colunas esparsas e compressão de estrutura colunar.

Trabalhando com grupos de AlwaysOn disponibilidade autogerenciados do SQL Server

A disponibilidade dos grupos de disponibilidade AlwaysOn do SQL Server fornece alta disponibilidade e recuperação de desastres como alternativa ao espelhamento do banco de dados no nível empresarial.

Em AWS DMS, você pode migrar as alterações de uma única réplica primária ou secundária do grupo de disponibilidade.

Como trabalhar com a réplica primária do grupo de disponibilidade

Para usar o grupo de disponibilidade principal como fonte em AWS DMS, faça o seguinte:

1. Ative a opção de distribuição em todas as instâncias do SQL Server em suas réplicas de disponibilidade. Para ter mais informações, consulte Configurar a replicação contínua em um SQL Server autogerenciado.
2. No AWS DMS console, abra as configurações do banco de dados de origem do SQL Server. Em Nome do servidor, especifique o nome do serviço de nomes de domínio (DNS) ou o endereço IP configurado para o receptor do grupo de disponibilidade.

Quando você inicia uma AWS DMS tarefa pela primeira vez, ela pode levar mais tempo do que o normal para começar. Essa lentidão ocorre porque a criação dos artigos da tabela está sendo duplicada pelo servidor de grupos de disponibilidade.

Como trabalhar com uma réplica do grupo de disponibilidade secundário

Para usar um grupo de disponibilidade secundário como origem AWS DMS, faça o seguinte:

1. Use as mesmas credenciais usadas pelo usuário do endpoint de AWS DMS origem para se conectar a réplicas individuais.
2. Certifique-se de que sua instância de AWS DMS replicação possa resolver os nomes DNS de todas as réplicas existentes e se conectar a elas. É possível utilizar a consulta SQL a seguir para obter os nomes DNS de todas as réplicas.

   ```
   select ar.replica_server_name, ar.endpoint_url from sys.availability_replicas ar
   JOIN sys.availability_databases_cluster adc
   ON adc.group_id = ar.group_id AND adc.database_name = '<source_database_name>';
   ```

3. Ao criar o endpoint de origem, especifique o nome DNS do receptor do grupo de disponibilidade do Nome do servidor do endpoint ou o Endereço do servidor do segredo do endpoint. Para obter mais informações sobre receptores de grupos de disponibilidade, consulte O que é um receptor de grupos de disponibilidade? na documentação do SQL Server.

   É possível utilizar um servidor DNS público ou um servidor DNS on-premises para resolver o receptor do grupo de disponibilidade, a réplica primária e as réplicas secundárias. Para utilizar um servidor DNS on-premises, configure o Amazon Route 53 Resolver. Para ter mais informações, consulte Utilização do seu próprio servidor de nomes on-premises.
4. Adicione os seguintes atributos de conexão adicional ao endpoint de origem.

Atributos de conexão adicional	Valor	Observações
`applicationIntent`	`ReadOnly`	Sem essa configuração de ODBC, a tarefa de replicação é roteada para a réplica primária do grupo de disponibilidade. Para obter mais informações, consulte Compatibilidade do SQL Server Native Client com alta disponibilidade e recuperação de desastres na documentação do SQL Server.
`multiSubnetFailover`	`yes`	Para obter mais informações, consulte Compatibilidade do SQL Server Native Client com alta disponibilidade e recuperação de desastres na documentação do SQL Server.
`alwaysOnSharedSynchedBackupIsEnabled`	`false`	Para ter mais informações, consulte Configurações de endpoint ao usar o SQL Server como fonte para AWS DMS.
`activateSafeguard`	`false`	Para obter mais informações, consulte Limitações a seguir.
`setUpMsCdcForTables`	`false`	Para obter mais informações, consulte Limitações a seguir.

5. Ative a opção de distribuição em todas as réplicas no grupo de disponibilidade. Adicione todos os nós à lista de distribuidores. Para ter mais informações, consulte Como configurar a distribuição.
6. Execute a consulta a seguir na réplica primária de leitura e gravação para ativar a publicação do banco de dados. Você executa essa consulta somente uma vez para o banco de dados.

```
sp_replicationdboption @dbname = N'<source DB name>', @optname = N'publish', @value
 = N'true';
```

Limitações

Veja a seguir as limitações ao trabalhar com uma réplica secundária do grupo de disponibilidade:

- AWS DMS não oferece suporte ao Safeguard ao usar uma réplica de grupo de disponibilidade somente para leitura como fonte. Para ter mais informações, consulte Configurações de endpoint ao usar o SQL Server como fonte para AWS DMS.
- AWS DMS não oferece suporte ao atributo de conexão `setUpMsCdcForTables` extra ao usar uma réplica de grupo de disponibilidade somente para leitura como fonte. Para ter mais informações, consulte Configurações de endpoint ao usar o SQL Server como fonte para AWS DMS.
- AWS DMS pode usar uma réplica autogerenciada do grupo de disponibilidade secundário como banco de dados de origem para replicação contínua (captura de dados de alteração ou CDC) a partir da versão 3.4.7. As réplicas de leitura do Cloud SQL Server Multi-AZ não são compatíveis. Se você usa versões anteriores do AWS DMS, certifique-se de usar a réplica principal do grupo de disponibilidade como banco de dados de origem para o CDC.

Failover para outros nós

Se você definir o atributo de conexão `ApplicationIntent` extra para seu endpoint`ReadOnly`, sua AWS DMS tarefa se conectará ao nó somente leitura com a maior prioridade de roteamento somente para leitura. Ele executa failover para outros nós somente leitura no grupo de disponibilidade quando o nó somente leitura de prioridade mais alta não está disponível. Se você não definir`ApplicationIntent`, sua AWS DMS tarefa se conectará somente ao nó primário (leitura/gravação) em seu grupo de disponibilidade.

Requisitos de segurança ao usar o SQL Server como fonte para AWS Database Migration Service

A conta de AWS DMS usuário deve ter pelo menos a função de `db_owner` usuário no banco de dados SQL Server de origem ao qual você está se conectando.

Configurações de endpoint ao usar o SQL Server como fonte para AWS DMS

É possível utilizar as configurações de endpoint para configurar a origem do SQL Server de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de origem usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o SQL Server como origem.

Nome	Descrição
`ActivateSafeguard`	Esse atributo ativa ou desativa o Safeguard. Para obter mais informações sobre o Safeguard, consulte a `SafeguardPolicy` a seguir: Valor padrão: `true` Valores válidos: {`false`, `true`} Exemplo: `'{"ActivateSafeguard": true}'`
`AlwaysOnSharedSynchedBackupIsEnabled`	Esse atributo ajusta o comportamento da migração de AWS DMS um banco de dados de origem do SQL Server hospedado como parte de um cluster de grupos de disponibilidade Always On. AWS DMS tem suporte aprimorado para bancos de dados de origem do SQL Server que estão configura dos para serem executados em um cluster Always On. Nesse caso, AWS DMS tenta rastrear se os backups de transações estão acontecendo em nós no cluster Always On diferentes do nó em que a instância do banco de dados de origem está hospedada. Na inicialização da tarefa de migração, AWS DMS tenta se conectar a cada nó no cluster, mas falha se não conseguir se conectar a nenhum dos nós.

Nome	Descrição
	Se você AWS DMS precisar pesquisar todos os nós no cluster Always On para backups de transações, defina esse atributo `false` como. Valor padrão: `true` Valores válidos: `true` ou `false` Exemplo: `'{"AlwaysOnSharedSynchedBackupIsEnabled": false}'`
`"ApplicationIntent": "readonly"`	Essa configuração de atributo de driver ODBC faz com que o SQL Server roteie a tarefa de replicação para o nó somente leitura de prioridade mais alta. Sem essa configuração, o SQL Server roteia a tarefa de replicação para o nó primário de leitura e gravação.
`EnableNonSysadminWrapper`	Utilize essa configuração de endpoint ao configurar a replicação contínua em um servidor SQL autônomo sem um usuário sysadmin. Esse parâmetro é suportado na AWS DMS versão 3.4.7 e superior. Para obter informações sobre como configurar a replicação contínua em um SQL Server autônomo, consulte Configurar a replicação contínua em um SQL Server autônomo: sem o perfil sysadmin. Valor padrão: `false` Valores válidos: `true`, `false` Exemplo: `'{"EnableNonSysadminWrapper": true}'`

Nome	Descrição
`ExecuteTimeout`	Utilize esse atributo de conexão adicional (ECA) para definir o tempo limite da instrução do cliente para a instância do SQL Server, em segundos. O valor padrão é de 60 segundos. Exemplo: `'{"ExecuteTimeout": 100}'`
`FatalOnSimpleModel`	Quando definida como `true`, essa configuração gera um erro fatal quando o modelo de recuperação de banco de dados SQL Server está definido como `simple`. Valor padrão: `false` Valores válidos: `true` ou `false` Exemplo: `'{"FatalOnSimpleModel": true}'`
`ForceLobLookup`	Força a pesquisa de LOB em LOB em linha. Valor padrão: `false` Valores válidos: `true`, `false` Exemplo: `'{"ForceLobLookup": false}'`
`"MultiSubnetFailover": "Yes"`	Esse atributo de driver ODBC ajuda o DMS a se conectar ao novo primário no caso de um failover do grupo de disponibilidade. Esse atributo foi criado para situações em que a conexão é interrompida ou o endereço IP do receptor está incorreto. Nessas situações, AWS DMS tenta se conectar a todos os endereços IP associados ao ouvinte do Grupo de Disponibilidade.

Nome	Descrição
`ReadBackupOnly`	A utilização desse atributo requer privilégios de sysadmin. Quando esse atributo é definido como`Y`, durante a replicação contínua, AWS DMS lê as alteraçõe s somente dos backups do log de transações e não lê do arquivo de log de transações ativo. A definição desse parâmetro como `Y` permite controlar o crescimento do log de transações ativas durante tarefas de carga máxima e de replicação contínua. No entanto, ele pode adicionar alguma latência de origem à replicação contínua. Valores válidos: `N` ou `Y`. O padrão é `N`. Exemplo: `'{"ReadBackupOnly": Y}'` Observação: esse parâmetro não funciona nas instâncias de origem do Amazon RDS SQL Server devido à maneira como o RDS executa backups.

Nome	Descrição
`SafeguardPolicy`	Para um desempenho ideal, AWS DMS tenta capturar todas as alterações não lidas do registro de transações ativo (TLOG). Contudo, às vezes devido a um truncamento, o TLOG ativo talvez não contenha todas as alterações não lidas. Quando isso ocorre, AWS DMS acessa o backup do log para capturar as alterações ausentes. Para minimizar a necessidade de acessar o backup do log, AWS DMS evita o truncamento usando um dos seguintes métodos: 1. `RELY_ON_SQL_SERVER_REPLICATION_AGENT` (Iniciar transações no banco de dados): Esse é o padrão para AWS DMS. Ao utilizar essa configuração, o AWS DMS requer que o agente de leitura do SQL Server esteja em execução, para que o AWS DMS possa transferir as transações marcadas para replicação no TLOG ativo. Observe que, se o agente de leitura de log não estiver em execução, o TLOG ativo poderá ficar cheio, fazendo com que o banco de dados de origem mude para o modo somente leitura até que você possa resolver o problema. Se você precisar habilitar a Replicação Microsoft em seu banco de dados para uma finalidade diferente de AWS DMS, escolha essa configuração. Ao usar essa configuração, AWS DMS minimiza as leituras de backup de log criando uma tabela chamada `awsdms_truncation_safeguard` e evita o truncamento do TLOG imitando uma transação aberta no banco de dados. Isso impede que o banco de dados trunque eventos e os transfira para o log de backup por cinco minutos (por padrão). Verifique se a tabela não está incluída em nenhum plano de

Nome	Descrição
	manutenção, pois isso pode causar falhas no trabalho de manutenção. É possível excluir a tabela com segurança se não houver tarefas configuradas com a opção de banco de dados `Start Transactions` . 2. `EXCLUSIVE_AUTOMATIC_TRUNCATION` (Uso exclusivo **`sp_repldone`** com uma única tarefa): Ao usar essa configuração, AWS DMS tem controle total do processo do agente de replicação que marca as entradas de registro como sendo `ready for truncation` usadas. `sp_repldone` Com essa configuração, AWS DMS não usa uma transação fictícia como na configuração `RELY_ON_SQL_SERVER_REPLICATION_AGENT` (padrão). Você só pode usar essa configuração quando o MS Replication não é usado para nenhuma outra finalidade que não seja AWS DMS no banco de dados de origem. Além disso, ao usar essa configuração, somente uma AWS DMS tarefa pode acessar o banco de dados. Se você precisar executar AWS DMS tarefas paralelas no mesmo banco de dados, use`RELY_ON_SQL_SERVER_REPLICATION_AGENT` . • Essa configuração requer que o agente de leitura de log seja interrompido no banco de dados. Se o Log Reader Agent estiver em execução quando a tarefa for iniciada, a AWS DMS tarefa a forçará a parar. Como alternativa, é possível interromper o agente de leitura de log manualmente antes de iniciar a tarefa. • Ao utilizar esse método com a MS-CDC, interrompa e desative os trabalhos de Captura da MS-CDC e de Limpeza da MS-CDC. • Você não pode usar essa configuração quando o trabalho de migração do Microsoft SQL Server é

Nome	Descrição
	executado em uma máquina remota do Distribui dor, porque AWS DMS não tem acesso à máquina remota. • EXCLUSIVE_AUTOMATIC_TRUNCATION não funciona nas instâncias de origem do Amazon RDS para SQL Server porque os usuários do Amazon RDS não têm acesso para executar o procedimento armazenado sp_repldone . • Se você definir SafeguardPolicy como EXCLUSIVE_AUTOMATIC_TRUNCATION sem utilizar o perfil sysadmin, deverá conceder permissões nos objetos dbo.syscategories e dbo.sysjobs ao usuário dmsuser. Valor padrão: RELY_ON_SQL_SERVER_REPLICAT ION_AGENT Valores válidos: {EXCLUSIVE_AUTOMATIC_TRUNCAT ION , RELY_ON_SQL_SERVER_REPLICAT ION_AGENT } Exemplo: '{"SafeguardPolicy": "EXCLUSIV E_AUTOMATIC_TRUNCATION"}'

Nome	Descrição
`SetUpMsCdcForTables`	Esse atributo ativa a MS-CDC para o banco de dados de origem e para tabelas no mapeamento de tarefa que não têm a MS Replication ativada. A definição desse valor como `true` executa o procedimento armazenado `sp_cdc_enable_db` no banco de dados de origem e o procedimento armazenado `sp_cdc_enable_table` em cada tabela na tarefa que não tem a MS Replication ativada no banco de dados de origem. Para obter mais informações sobre como ativar a distribuição, consulte Configurar a replicação contínua em um SQL Server autogerenciado. Valores válidos: {`true`, `false`} Exemplo: `'{"SetUpMsCdcForTables": true}'`
`TlogAccessMode`	Indica o modo utilizado para buscar dados da CDC. Valor padrão: `PreferTlog` Valores válidos: `BackupOnly` , `PreferBackup` , `PreferTlog` , `TlogOnly` Exemplo: `'{"TlogAccessMode": "PreferTlog"}'`
`Use3rdPartyBackupDevice`	Quando esse atributo for definido como Y, o AWS DMS processará backups de logs de transações de terceiros se eles forem criados no formato nativo.

Tipos de dados de origem no SQL Server

A migração de dados que usa o SQL Server como fonte de AWS DMS suporte à maioria dos tipos de dados do SQL Server. A tabela a seguir mostra os tipos de dados de origem do SQL Server que são suportados durante o uso AWS DMS e o mapeamento padrão dos tipos de AWS DMS dados.

Para obter informações sobre como visualizar o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que você está usando.

Para obter informações adicionais sobre AWS DMS os tipos de dados, consulteTipos de dados do AWS Database Migration Service.

Tipos de dados do SQL Server	AWS DMS tipos de dados
BIGINT	INT8
BIT	BOOLEAN
DECIMAL	NUMERIC
INT	INT4
MONEY	NUMERIC
NUMERIC (p,s)	NUMERIC
SMALLINT	INT2
SMALLMONEY	NUMERIC
TINYINT	UINT1
REAL	REAL4
FLOAT	REAL8
DATETIME	DATETIME
DATETIME2 (SQL Server 2008 e superior)	DATETIME
SMALLDATETIME	DATETIME
DATA	DATA
TIME	TIME
DATETIMEOFFSET	WSTRING

Tipos de dados do SQL Server	AWS DMS tipos de dados
CHAR	STRING
VARCHAR	STRING
VARCHAR (máximo)	CLOB TEXT Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de tipos de dados CLOB para uma tarefa específica. Para tabelas do SQL Server, AWS DMS atualiza as colunas LOB no destino até mesmo para instruções UPDATE que não alteram o valor da coluna LOB no SQL Server. Durante o CDC, AWS DMS suporta tipos de dados CLOB somente em tabelas que incluem uma chave primária.
NCHAR	WSTRING
NVARCHAR (tamanho)	WSTRING

Tipos de dados do SQL Server	AWS DMS tipos de dados
NVARCHAR (máximo)	NCLOB NTEXT Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de SupportLobs para uma tarefa específica. Para obter mais informações sobre como ativar a compatibi lidade com o LOB, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS. Para tabelas do SQL Server, AWS DMS atualiza as colunas LOB no destino até mesmo para instruções UPDATE que não alteram o valor da coluna LOB no SQL Server. Durante o CDC, AWS DMS suporta tipos de dados CLOB somente em tabelas que incluem uma chave primária.
BINARY	BYTES
VARBINARY	BYTES

Tipos de dados do SQL Server	AWS DMS tipos de dados
VARBINARY (máximo)	BLOB IMAGE Para tabelas do SQL Server, AWS DMS atualiza as colunas LOB no destino até mesmo para instruções UPDATE que não alteram o valor da coluna LOB no SQL Server. Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de tipos de dados BLOB para uma tarefa específica. AWS DMS suporta tipos de dados BLOB somente em tabelas que incluem uma chave primária.
TIMESTAMP	BYTES
UNIQUEIDENTIFIER	STRING
HIERARCHYID	Utilize o tipo HIERARCHYID ao replicar para um endpoint de destino do SQL Server. Utilize WSTRING (250) ao replicar para os demais endpoints de destino.

Tipos de dados do SQL Server	AWS DMS tipos de dados
XML	NCLOB Para tabelas do SQL Server, AWS DMS atualiza as colunas LOB no destino até mesmo para instruções UPDATE que não alteram o valor da coluna LOB no SQL Server. Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso dos tipos de dados NCLOB para uma tarefa específica. Durante o CDC, AWS DMS suporta tipos de dados NCLOB somente em tabelas que incluem uma chave primária.
GEOMETRY	Utilize o tipo GEOMETRY ao replicar para endpoints de destino compatíveis com esse tipo de dados. Use o tipo CLOB ao replicar para endpoints de destino que não são compatíveis com esse tipo de dados.
GEOGRAPHY	Utilize o tipo GEOGRAPHY ao replicar para endpoints de destino compatíveis com esse tipo de dados. Use o tipo CLOB ao replicar para endpoints de destino que não são compatíveis com esse tipo de dados.

AWS DMS não oferece suporte a tabelas que incluam campos com os seguintes tipos de dados.

- CURSOR
- SQL_VARIANT
- TABLE

> **Note**
>
> A existência de suporte para um tipo de dados definido pelo usuário vai depender do tipo base utilizado. Por exemplo, um tipo de dados definido pelo usuário baseado no tipo DATETIME é tratado como o tipo de dados DATETIME.

Utilizar um banco de dados Microsoft Azure SQL como a origem do AWS DMS

Com o AWS DMS, é possível utilizar o banco de dados Microsoft Azure SQL como origem da mesma maneira como você utiliza o SQL Server. O AWS DMS é compatível, como origem, com a mesma lista de versões de bancos de dados que são compatíveis com o SQL Server em execução on-premises ou em uma instância do Amazon EC2.

Para obter mais informações, consulte Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS.

> **Note**
>
> O AWS DMS não oferece suporte a operações de captura de dados de alterações (CDC) com bancos de dados Azure SQL.

Utilizar a instância gerenciada do Microsoft Azure SQL como origem do AWS DMS

Com o AWS DMS, é possível utilizar o Microsoft Azure SQL como origem da mesma maneira como utiliza o SQL Server. O AWS DMS é compatível, como origem, com a mesma lista de versões de bancos de dados que são compatíveis com o SQL Server em execução on-premises ou em uma instância do Amazon EC2.

Para obter mais informações, consulte Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS.

Utiliza o Microsoft Azure Database para PostgreSQL como origem do AWS DMS

Com o AWS DMS, é possível utilizar o servidor flexível do Microsoft Azure Database para PostgreSQL como origem da mesma maneira como você utiliza o PostgreSQL.

Para obter informações sobre as versões do servidor flexível do Microsoft Azure Database para PostgreSQL como origem com as quais o AWS DMS é compatível, consulte Fontes para AWS DMS.

Configurar o servidor flexível do Microsoft Azure para PostgreSQL para replicação lógica e decodificação

É possível utilizar os recursos de replicação lógica e decodificação no servidor flexível do Microsoft Azure Database para PostgreSQL durante a migração do banco de dados.

Para decodificação lógica, o DMS utiliza o plug-in `test_decoding` ou `pglogical`. Se o plug-in `pglogical` estiver disponível em um banco de dados PostgreSQL de origem, o DMS criará um slot de replicação utilizando o `pglogical`, caso contrário, o plug-in `test_decoding` será utilizado.

Para configurar o servidor flexível do Microsoft Azure para PostgreSQL como um endpoint de origem para o DMS, execute as seguintes etapas:

1. Abra a página Parâmetros do servidor no portal.
2. Defina o parâmetro `wal_level` do servidor como `LOGICAL`.
3. Se quiser utilizar a extensão `pglogical`, defina os parâmetros `shared_preload_libraries` e `azure.extensions` como`pglogical`.
4. Defina o parâmetro `max_replication_slots` como o número máximo de tarefas do DMS que você planeja executar simultaneamente. No Microsoft Azure, o valor padrão desse parâmetro é 10. O valor máximo desse parâmetro depende da memória disponível na instância do PostgreSQL, permitindo entre 2 e 8 slots de replicação por GB de memória.
5. Defina o parâmetro `max_wal_senders` como um valor maior que 1. O parâmetro `max_wal_senders` define o número de tarefas simultâneas que podem ser executadas. O valor padrão é 10.
6. Defina o valor do parâmetro `max_worker_processes` como pelo menos 16. Caso contrário, você poderá ver erros como os seguintes:

```
WARNING: out of background worker slots.
```

7. Salve as alterações. Reinicie o servidor para aplicar as alterações.
8. Confirme se a instância do PostgreSQL permite tráfego de rede no recurso de conexão.
9. Conceda permissões de replicação a um usuário existente ou crie um novo usuário com permissões de replicação utilizando os comandos a seguir.

   - Conceda as permissões de replicação a um usuário existente utilizando o seguinte comando:

     ```
     ALTER USER <existing_user> WITH REPLICATION;
     ```

   - Crie um novo usuário com permissões de replicação utilizando o seguinte comando:

     ```
     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;
     ```

Para obter mais informações sobre a replicação lógica com o PostgreSQL, consulte os tópicos a seguir:

- Ativar a captura de dados de alteração (CDC) utilizando replicação lógica
- Utilizar pontos de início nativos da CDC para configurar uma carga de CDC de uma origem PostgreSQL
- Replicação lógica e decodificação lógica no Azure Database for PostgreSQL: servidor flexível na Documentação do Azure Database para PostgreSQL.

Utilizar um servidor flexível do Microsoft Azure Database para MySQL Server como origem do AWS DMS

Com o AWS DMS, é possível utilizar o Microsoft Azure Database para MySQL como origem da mesma maneira como utiliza o MySQL.

Para obter informações sobre as versões do servidor flexível do Microsoft Azure Database para MySQL como origem com as quais o AWS DMS é compatível, consulte Fontes para AWS DMS.

Para obter mais informações sobre como utilizar um banco de dados compatível com o MySQL gerenciado pelo cliente com o AWS DMS, consulte Usando um banco de dados autogerenciado compatível com MySQL como fonte para AWS DMS.

Limitações ao utilizar o Azure MySQL como origem do AWS Database Migration Service

- O valor padrão da variável `sql_generate_invisible_primary_key` do sistema do servidor flexível do Azure MySQL é `ON`, e o servidor adiciona automaticamente uma chave primária invisível gerada (GIPK) a qualquer tabela criada sem uma chave primária explícita. O AWS DMS não é compatível com a replicação contínua para tabelas do MySQL com restrições de GIPK.

Utilizar o OCI MySQL Heatwave como origem do AWS DMS

Com o AWS DMS, é possível utilizar o OCI MySQL Heatwave como origem da mesma forma como você utiliza o MySQL. A utilização do OCI MySQL Heatwave como origem requer algumas mudanças adicionais na configuração.

Para obter informações sobre as versões do OCI MySQL Heatwave como origem compatíveis com o AWS DMS, consulte Fontes para AWS DMS.

Configurar o OCI MySQL Heatwave para replicação lógica

Para configurar a instância do OCI MySQL Heatwave como um endpoint de origem do DMS, faça o seguinte:

1. Faça login no console do OCI e abra o menu principal de hambúrguer (≡) no canto superior esquerdo.
2. Escolha Bancos de dados, Sistemas de banco de dados.
3. Abra o menu Configurações.
4. Escolha Criar configuração.
5. Insira um nome da configuração, como **`dms_configuration`**.
6. Escolha a forma da instância do OCI MySQL Heatwave atual. É possível encontrar a forma da Configuração do sistema de banco de dados da instância, na seção Configuração do sistema de banco de dados: forma.
7. Na seção Variáveis do usuário, escolha a variável `binlog_row_value_options` do sistema. O valor padrão é `PARTIAL_JSON`. Limpe o valor.
8. Escolha o botão Criar.
9. Abra a instância do OCI MySQLHeatwave e escolha o botão Editar.

10. Na seção Configuração, escolha o botão Alterar configuração e escolha a configuração da forma que você criou na etapa 4.
11. Depois que as alterações estiverem em vigor, a instância estará pronta para a replicação lógica.

Utilizar o Google Cloud para MySQL como a origem do AWS DMS

Com o AWS DMS, é possível utilizar o Google Cloud para MySQL como a origem da mesma forma como utiliza o MySQL.

Para obter informações sobre as versões do GCP MySQL como origem compatíveis com o AWS DMS, consulte Fontes para AWS DMS.

Para obter mais informações, consulte Utilizar um banco de dados compatível com MySQL como origem do AWS DMS.

> **Note**
>
> O suporte para o GCP MySQL 8.0 como origem está disponível no AWS DMS versão 3.4.6.
> O AWS DMS não é compatível com o modo SSL `verify-full` de instâncias do GCP para MySQL.
> A configuração de segurança `Allow only SSL connections` do GCP MySQL não é compatível porque exige a verificação do certificado do servidor e do cliente. O AWS DMS só é compatível com a verificação do certificado do servidor.
> O AWS DMS é compatível com o valor padrão do GCP CloudSQL para MySQL do `CRC32` para o sinalizador de banco de dados `binlog_checksum`.

Utilizar o Google Cloud para PostgreSQL como origem do AWS DMS

Com o AWS DMS, é possível utilizar o Google Cloud para PostgreSQL como origem da mesma forma como você utiliza bancos de dados PostgreSQL autogerenciados.

Para obter informações sobre as versões do GCP PostgreSQL como origem compatíveis com o AWS DMS, consulte Fontes para AWS DMS.

Para obter mais informações, consulte Utilizar o banco de dados PostgreSQL como origem do AWS DMS.

Configurar o Google Cloud para PostgreSQL para replicação lógica e decodificação

É possível utilizar os recursos lógicos de replicação e de decodificação no Google Cloud SQL para PostgreSQL durante a migração do banco de dados.

Para decodificação lógica, o DMS utiliza um dos seguintes plug-ins:

- `test_decoding`
- `pglogical`

Se o plug-in `pglogical` estiver disponível em um banco de dados PostgreSQL de origem, o DMS criará um slot de replicação utilizando o `pglogical`, caso contrário, o plug-in `test_decoding` será utilizado.

Observe o seguinte a respeito da utilização da decodificação lógica com o AWS DMS:

1. Com o Google Cloud SQL para PostgreSQL, ative a decodificação lógica definindo a sinalização `cloudsql.logical_decoding` como `on`.
2. Para ativar o `pglogical`, defina o sinalizador `cloudsql.enable_pglogical` como `on` e reinicie o banco de dados.
3. Para utilizar os recursos de decodificação lógica, crie um usuário do PostgreSQL com o atributo `REPLICATION`. Ao utilizar a extensão do `pglogical`, o usuário deve ter o perfil `cloudsqlsuperuser`. Para criar um usurário com o perfil `cloudsqlsuperuser`, faça o seguinte:

   ```
   CREATE USER new_aws_dms_user WITH REPLICATION
   IN ROLE cloudsqlsuperuser LOGIN PASSWORD 'new_aws_dms_user_password';
   ```

   Para definir esse atributo em um usuário existente, faça o seguinte:

   ```
   ALTER USER existing_user WITH REPLICATION;
   ```

4. Defina o parâmetro `max_replication_slots` como o número máximo de tarefas do DMS que você planeja executar simultaneamente. No Google Cloud SQL, o valor padrão desse parâmetro é 10. O valor máximo desse parâmetro depende da memória disponível na instância do PostgreSQL, permitindo entre 2 e 8 slots de replicação por GB de memória.

Para obter mais informações sobre a replicação lógica com o PostgreSQL, consulte os tópicos a seguir:

- Ativar a captura de dados de alteração (CDC) utilizando replicação lógica
- Utilizar pontos de início nativos da CDC para configurar uma carga de CDC de uma origem PostgreSQL
- Configurar a replicação lógica e a decodificação na Documentação do Cloud SQL para PostgreSQL.

Utilizar o banco de dados PostgreSQL como origem do AWS DMS

Você pode migrar dados de um ou vários bancos de dados PostgreSQL usando o. AWS DMS Com um banco de dados PostgreSQL como origem, é possível migrar dados para outro banco de dados PostgreSQL ou para um dos outros bancos de dados compatíveis.

Para obter informações sobre as versões do PostgreSQL AWS DMS que oferecem suporte como fonte, consulte. Fontes para AWS DMS

AWS DMS oferece suporte ao PostgreSQL para esses tipos de bancos de dados:

- Bancos de dados on-premises
- Bancos de dados em uma instância do Amazon EC2
- Bancos de dados em uma instância de banco de dados Amazon RDS
- Bancos de dados em uma instância de banco de dados com base na edição do Amazon Aurora compatível com PostgreSQL
- Bancos de dados em uma instância de banco de dados com base na edição Tecnologia Sem Servidor do Amazon Aurora compatível com o PostgreSQL

> **Note**
>
> O DMS é compatível com o Amazon Aurora PostgreSQL—com Tecnologia Sem Servidor V1 como origem somente para carga máxima. Mas é possível utilizar o Amazon Aurora PostgreSQL—com Tecnologia Sem Servidor V2 como origem para tarefas de carga máxima, carga máxima + CDC e CDC somente.

AWS DMS versão a ser usada
Use qualquer AWS DMS versão disponível.
Use a AWS DMS versão 3.4.3 e superior.
Use a AWS DMS versão 3.4.7 e superior.
Use a AWS DMS versão 3.5.1 e superior.
Use a AWS DMS versão 3.5.3 e superior.

É possível utilizar o Secure Socket Layers (SSL) para criptografar conexões entre o endpoint do PostgreSQL e a instância de replicação. Para obter mais informações sobre como utilizar o SSL com um endpoint do PostgreSQL, consulte Usando SSL com AWS Database Migration Service.

O único requisito de segurança ao utilizar o PostgreSQL como origem é que a conta de usuário especificada deve ser de um usuário registrado no banco de dados PostgreSQL.

Para configurar um banco de dados PostgreSQL como AWS DMS um endpoint de origem, faça o seguinte:

- Crie um usuário do PostgreSQL com as permissões apropriadas para AWS DMS fornecer acesso ao seu banco de dados de origem do PostgreSQL.

> **Note**
>
> - Se o banco de dados de origem PostgreSQL for autogerenciado, consulte Trabalhando com bancos de dados PostgreSQL autogerenciados como fonte em AWS DMS para obter mais informações.

- Se o banco de dados de origem PostgreSQL for gerenciado pelo Amazon RDS, consulte Trabalhando com bancos AWS de dados PostgreSQL gerenciados como fonte de DMS para obter mais informações.

- Crie um endpoint de origem do PostgreSQL que esteja em conformidade com a configuração do banco do dados PostgreSQL escolhida.
- Crie uma tarefa ou um conjunto de tarefas para migrar as tabelas.

  Para criar uma full-load-only tarefa, nenhuma configuração adicional de endpoint é necessária.

  Antes de criar uma tarefa de captura de dados de alteração (uma tarefa de CDC somente ou de carga máxima e CDC), consulte Habilitando o CDC usando um banco de dados PostgreSQL autogerenciado como fonte AWS DMS ou Habilitando o CDC com uma instância de banco AWS de dados PostgreSQL gerenciada com AWS DMS.

Tópicos

- Trabalhando com bancos de dados PostgreSQL autogerenciados como fonte em AWS DMS
- Trabalhando com bancos AWS de dados PostgreSQL gerenciados como fonte de DMS
- Ativar a captura de dados de alteração (CDC) utilizando replicação lógica
- Utilizar pontos de início nativos da CDC para configurar uma carga de CDC de uma origem PostgreSQL
- Migrando do PostgreSQL para o PostgreSQL usando AWS DMS
- Migração do Babelfish para o Amazon Aurora PostgreSQL usando AWS DMS
- Removendo AWS DMS artefatos de um banco de dados de origem do PostgreSQL
- Definições de configuração adicionais ao utilizar um banco de dados PostgreSQL como origem do DMS
- Usando a configuração de MapBooleanAsBoolean endpoint do PostgreSQL
- Configurações de endpoint e atributos extras de conexão (ECAs) ao usar o PostgreSQL como fonte de DMS
- Limitações ao utilizar um banco de dados PostgreSQL como origem do DMS
- Tipos de dados de origem para o PostgreSQL

Trabalhando com bancos de dados PostgreSQL autogerenciados como fonte em AWS DMS

Com um banco de dados PostgreSQL autogerenciado como fonte, você pode migrar dados para outro banco de dados PostgreSQL ou para um dos outros bancos de dados de destino suportados pelo. AWS DMS A origem do banco de dados pode ser um banco de dados on-premises ou um mecanismo autogerenciado em execução em uma instância do Amazon EC2. É possível utilizar uma instância de banco de dados para tarefas de carga máxima e de captura de dados de alteração (CDC).

Pré-requisitos para usar um banco de dados PostgreSQL autogerenciado como fonte AWS DMS

Antes de migrar dados de um banco de dados PostgreSQL de origem autogerenciado, faça o seguinte:

- Utilize um banco de dados PostgreSQL com a versão 9.4.x ou superior.
- Para tarefas de carga máxima mais CDC ou tarefas de somente CDC, conceda permissões de superusuário para a conta de usuário especificada para o banco de dados de origem do PostgreSQL. A conta de usuário precisa de permissões de superusuário para acessar perfis específicos de replicação na origem. Para tarefas somente de carga máxima, a conta de usuário precisa das permissões SELECT nas tabelas para migrá-las.
- Adicione o endereço IP do servidor de AWS DMS replicação ao arquivo de `pg_hba.conf` configuração e habilite a replicação e as conexões de soquete. Veja a seguir um exemplo.

  ```
  # 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
  ```

  O arquivo de configuração `pg_hba.conf` do PostgreSQL controla a autenticação do cliente. (HBA significa autenticação baseada em host.) O arquivo é tradicionalmente armazenado no diretório de dados do cluster de banco de dados.
- Se você estiver configurando um banco de dados como fonte para replicação lógica usando consulte AWS DMS Habilitando o CDC usando um banco de dados PostgreSQL autogerenciado como fonte AWS DMS

> **Note**
>
> Algumas AWS DMS transações ficam inativas por algum tempo antes que o mecanismo do DMS as use novamente. Com a utilização do parâmetro `idle_in_transaction_session_timeout` no PostgreSQL versões 9.6 e superior é possível fazer com que as transações ociosas atinjam o tempo limite e falhem. Não encerre transações ociosas quando você usar o AWS DMS.

Habilitando o CDC usando um banco de dados PostgreSQL autogerenciado como fonte AWS DMS

AWS DMS suporta captura de dados de alteração (CDC) usando replicação lógica. Para ativar a replicação lógica de um banco de dados de origem do PostgreSQL autogerenciado, defina os seguintes parâmetros e valores no arquivo de configuração `postgresql.conf`:

- Defina `wal_level = logical`.
- Defina `max_replication_slots` como um valor maior que 1.

  Defina o valor de `max_replication_slots` de acordo com o número de tarefas a serem executadas. Por exemplo, para executar cinco tarefas, defina no mínimo cinco slots. Os slots são abertos automaticamente assim que uma tarefa é iniciada e permanecem abertos até mesmo quando a tarefa não está mais em execução. Exclua manualmente os slots abertos. Observe que o DMS descarta automaticamente os slots de replicação quando a tarefa é excluída, se o DMS tiver criado o slot.
- Defina `max_wal_senders` como um valor maior que 1.

  O parâmetro `max_wal_senders` define o número de tarefas simultâneas que podem ser executadas.
- O parâmetro `wal_sender_timeout` encerra as conexões de replicação que estão inativas por mais tempo do que o número de milissegundos especificado. O padrão para um banco de dados PostgreSQL on-premises é 60000 milissegundos (60 segundos). A definição do valor como 0 (zero) desativa o mecanismo de tempo limite e é uma configuração válida para o DMS.

  Ao definir `wal_sender_timeout` como um valor diferente de zero, uma tarefa do DMS com CDC requer um mínimo de 10000 milissegundos (10 segundos) e falha se o valor for menor que 10000. Mantenha o valor em menos de 5 minutos para evitar provocar um atraso durante um failover de multi-AZ de uma instância de replicação do DMS.

Alguns parâmetros são estáticos, e você só pode defini-los na inicialização do servidor. Quaisquer alterações nas entradas do arquivo de configuração (para um banco de dados autogerenciado) ou no grupo de parâmetros do banco de dados (para um banco de dados RDS para PostgreSQL) são ignoradas até que o servidor seja reiniciado. Para obter mais informações, consulte a Documentação do PostgreSQL.

Para obter mais informações sobre como ativar a CDC, consulte Ativar a captura de dados de alteração (CDC) utilizando replicação lógica.

Trabalhando com bancos AWS de dados PostgreSQL gerenciados como fonte de DMS

Você pode usar uma instância AWS de banco de dados PostgreSQL gerenciada como fonte para. AWS DMSÉ possível executar tarefas de carga máxima e tarefas de captura de dados de alteração (CDC) utilizando uma origem PostgreSQL gerenciado pela AWS.

Pré-requisitos para usar um banco de dados AWS PostgreSQL gerenciado como fonte de DMS

Antes de migrar dados de um banco de dados de origem AWS PostgreSQL gerenciado, faça o seguinte:

- Recomendamos que você use uma conta de AWS usuário com as permissões mínimas necessárias para a instância de banco de dados PostgreSQL como a conta de usuário para o endpoint de origem do PostgreSQL. AWS DMS A utilização de uma conta mestre não é recomendada. A conta deve ter o perfil `rds_superuser` e o perfil `rds_replication`. O perfil `rds_replication` concede permissões para gerenciar slots lógicos e transmitir dados utilizando slots lógicos.

  Crie vários objetos da conta do usuário mestre para a conta que você utiliza. Para obter mais informações sobre como criar esses objetos, consulte Migrar um banco de dados Amazon RDS para PostgreSQL sem utilizar a conta de usuário mestre.
- Se o banco de dados de origem estiver em uma nuvem privada virtual (VPC), selecione o grupo de segurança da VPC que fornece acesso à instância de banco de dados em que o banco de dados reside. Isso é necessário para que a instância de replicação do DMS se conecte com êxito à instância de banco de dados de origem. Quando o banco de dados e a instância de replicação do DMS estiverem na mesma VPC, adicione o grupo de segurança apropriado às suas próprias regras de entrada.

> **Note**
>
> Algumas AWS DMS transações ficam inativas por algum tempo antes que o mecanismo do DMS as use novamente. Com a utilização do parâmetro `idle_in_transaction_session_timeout` no PostgreSQL versões 9.6 e superior é possível fazer com que as transações ociosas atinjam o tempo limite e falhem. Não encerre transações ociosas quando você usar o AWS DMS.

Habilitando o CDC com uma instância de banco AWS de dados PostgreSQL gerenciada com AWS DMS

AWS DMS oferece suporte ao CDC nos bancos de dados PostgreSQL do Amazon RDS quando a instância de banco de dados está configurada para usar a replicação lógica. A tabela a seguir resume a compatibilidade de replicação lógica de cada versão AWS gerenciada do PostgreSQL.

Não é possível utilizar as réplicas de leitura RDS PostgreSQL para CDC (replicação contínua).

Versão do PostgreSQL	AWS DMS suporte de carga total	AWS DMS Suporte CDC
Aurora PostgreSQL versão 2.1 compatível com o PostgreSQL 10.5 (ou inferior)	Sim	Não
Compatibilidade do Aurora PostgreSQL versão 2.2 com o PostgreSQL 10.6 (ou superior)	Sim	Sim
Compatibilidade do RDS para PostgreSQL com o PostgreSQL 10.21 (ou superior)	Sim	Sim

Como ativar a replicação lógica para uma instância de banco de dados RDS PostgreSQL

1. Use a conta de usuário AWS principal para a instância de banco de dados PostgreSQL como a conta de usuário para o endpoint de origem do PostgreSQL. A conta de usuário mestra tem as funções necessárias para permitir a configuração da captura de dados de alteração (CDC).

Se você utilizar uma conta diferente da conta de usuário mestre, deverá criar vários objetos da conta mestre para a conta utilizada. Para ter mais informações, consulte Migrar um banco de dados Amazon RDS para PostgreSQL sem utilizar a conta de usuário mestre.

2. Defina o parâmetro `rds.logical_replication` no grupo de parâmetros do CLUSTER do banco de dados como 1. Esse parâmetro estático requer uma reinicialização da instância de banco de dados para entrar em vigor. Como parte da aplicação desse parâmetro, o AWS DMS define os parâmetros `wal_level`, `max_wal_senders`, `max_replication_slots` e `max_connections`. Essas alterações de parâmetros podem aumentar o log de gravação antecipada (WAL), portanto, só defina `rds.logical_replication` ao utilizar slots de replicação lógica.
3. O parâmetro `wal_sender_timeout` encerra as conexões de replicação que estão inativas por mais tempo do que o número de milissegundos especificado. O padrão para um banco AWS de dados PostgreSQL gerenciado é 30.000 milissegundos (30 segundos). A definição do valor como 0 (zero) desativa o mecanismo de tempo limite e é uma configuração válida para o DMS.

   Ao definir `wal_sender_timeout` como um valor diferente de zero, uma tarefa do DMS com CDC requer um mínimo de 10000 milissegundos (10 segundos) e falhará se o valor estiver entre 0 e 10000. Mantenha o valor em menos de 5 minutos para evitar provocar um atraso durante um failover de multi-AZ de uma instância de replicação do DMS.
4. Verifique se o valor do parâmetro `max_worker_processes` no grupo de parâmetros do cluster do banco de dados é igual ou superior aos valores totais combinados de `max_logical_replication_workersautovacuum_max_workers` e `max_parallel_workers`. Um alto número de processos de trabalho em segundo plano pode impactar as workloads das aplicações em instâncias pequenas. Portanto, monitore o desempenho do banco de dados se você definir um valor de `max_worker_processes` mais alto que o padrão.
5. Ao usar o Aurora PostgreSQL como fonte com o CDC, defina como. `synchronous_commit` `0N`

Migrar um banco de dados Amazon RDS para PostgreSQL sem utilizar a conta de usuário mestre

Em alguns casos, é possível não utilizar a conta de usuário mestre para a instância de banco de dados Amazon RDS PostgreSQL que você está utilizando como origem. Nesses casos, crie vários objetos para capturar os eventos da linguagem de definição de dados (DDL). Crie esses objetos utilizando uma conta diferente da conta mestrr e, depois, crie um acionador na conta de usuário mestre.

> **Note**
>
> Se você definir a configuração do endpoint `captureDDLs` como `false` no endpoint de origem, não precisará criar a tabela e o acionador a seguir no banco de dados de origem.

Utilize o procedimento a seguir para criar esses objetos.

Como criar objetos

1. Escolha um esquema onde os objetos serão criados. O esquema padrão é `public`. Verifique se o esquema existe e pode ser acessado pela conta *`OtherThanMaster`*.
2. Faça login na instância de banco de dados PostgreSQL utilizando uma conta de usuário diferente da conta mestre, aqui a conta *`OtherThanMaster`*.
3. Crie a tabela `awsdms_ddl_audit` executando o comando a seguir, substituindo *`objects_schema`* no código seguinte pelo nome do esquema a ser utilizado.

```
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. Crie o perfil `awsdms_intercept_ddl` executando o comando a seguir, substituindo *`objects_schema`* no código seguinte pelo nome do esquema a ser utilizado.

```
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. Faça logout da conta *OtherThanMaster* e faça login com uma conta que tenha o perfil `rds_superuser` atribuído.
6. Crie o trigger de evento `awsdms_intercept_ddl` executando o comando a seguir.

```
CREATE EVENT TRIGGER awsdms_intercept_ddl ON ddl_command_end
EXECUTE PROCEDURE objects_schema.awsdms_intercept_ddl();
```

7. Verifique se todos os usuários e perfis que acessam esses eventos têm as permissões de DDL necessárias. Por exemplo: .

```
grant all on public.awsdms_ddl_audit to public;
grant all on public.awsdms_ddl_audit_c_key_seq to public;
```

Ao concluir o procedimento anterior, você poderá criar o endpoint de origem do AWS DMS utilizando a conta *OtherThanMaster*.

> **Note**
>
> Esses eventos são acionados pelas instruções `CREATE TABLE`, `ALTER TABLE` e `DROP TABLE`.

Ativar a captura de dados de alteração (CDC) utilizando replicação lógica

É possível utilizar o recurso de replicação lógica nativa do PostgreSQL para ativar a captura de dados de alteração (CDC) durante a migração do banco de dados de origens do PostgreSQL. É possível utilizar esse recurso com o PostgreSQL autogerenciado e também com uma instância do banco de dados Amazon RDS para PostgreSQL. Essa abordagem reduz o tempo de inatividade e ajuda a garantir que o banco de dados de destino esteja sincronizado com o banco de dados PostgreSQL de origem.

AWS DMS suporta CDC para tabelas PostgreSQL com chaves primárias. Se uma tabela não tiver uma chave primária, os logs de gravação antecipada (WAL) não incluirão uma imagem anterior da linha do banco de dados. Nesse caso, o DMS não pode atualizar a tabela. Aqui, é possível utilizar configurações adicionais e utilizar a identidade da réplica da tabela como uma solução alternativa. No entanto, essa abordagem pode gerar logs adicionais. É recomendável utilizar a identidade da réplica da tabela como solução alternativa somente após testes cuidadosos. Para ter mais informações, consulte Definições de configuração adicionais ao utilizar um banco de dados PostgreSQL como origem do DMS.

> **Note**
>
> A REPLICA IDENTITY FULL é compatível com um plug-in de decodificação lógica, mas não com um plug-in pglogical. Para obter mais informações, consulte a Documentação do pglogical.

Para tarefas de carga total e somente CDC e CDC, AWS DMS usa slots de replicação lógica para reter os registros do WAL para replicação até que os registros sejam decodificados. Ao reiniciar (não ao retomar) para uma carga máxima e uma tarefa de CDC ou uma tarefa de somente de CDC, o slot de replicação é recriado.

> **Note**
>
> Para decodificação lógica, o DMS utiliza o plug-in test_decoding ou pglogical. Se o plug-in pglogical estiver disponível em um banco de dados PostgreSQL de origem, o DMS criará um slot de replicação utilizando pglogical, caso contrário, um plug-in test_decoding será utilizado. Para obter mais informações sobre o plug-in test_decoding, consulte a Documentação do PostgreSQL.
>
> Se o parâmetro `max_slot_wal_keep_size` do banco de dados for definido como um valor não padrão e o `restart_lsn` de um slot de replicação ficar atrás do LSN atual em mais do que esse tamanho, a tarefa do DMS falhará devido à remoção dos arquivos WAL necessários.

Configurar o plug-in pglogical

Implementado como uma extensão do PostgreSQL, o plug-in pglogical é um sistema de replicação lógica e um modelo para replicação seletiva de dados. A tabela a seguir identifica as versões de origem do banco de dados PostgreSQL que são compatíveis com o plug-in pglogical.

Origem PostgreSQL	Compatível com o pglogical
PostgreSQL 9.4 autogerenciado ou superior	Sim
Amazon RDS PostgreSQL 9.5 ou inferior	Não
Amazon RDS PostgreSQL 9.6 ou superior	Sim
Aurora PostgreSQL 1.x até 2.5.x	Não
Aurora PostgreSQL 2.6.x ou superior	Sim
Aurora PostgreSQL 3.3.x ou superior	Sim

Antes de configurar o pglogical para uso com AWS DMS, primeiro habilite a replicação lógica para captura de dados de alteração (CDC) em seu banco de dados de origem do PostgreSQL.

- Para obter informações sobre como ativar a replicação lógica para CDC em bancos de dados de origem PostgreSQL autogerenciados, consulte Habilitando o CDC usando um banco de dados PostgreSQL autogerenciado como fonte AWS DMS
- Para obter informações sobre como ativar a replicação lógica para CDC em bancos de dados de origem PostgreSQL gerenciado pela AWS, consulte Habilitando o CDC com uma instância de banco AWS de dados PostgreSQL gerenciada com AWS DMS.

Depois que a replicação lógica estiver ativada no banco de dados de origem PostgreSQL, utilize as etapas a seguir para configurar o pglogical para utilização com o DMS.

Para usar o plug-in pglogical para replicação lógica em um banco de dados de origem PostgreSQL com AWS DMS

1. Crie uma extensão de pglogical no banco de dados de origem PostgreSQL:

   a. Defina o parâmetro correto:

      - Para bancos de dados PostgreSQL autogerenciados, defina o parâmetro `shared_preload_libraries= 'pglogical'` do banco de dados.
      - Para o PostgreSQL nos bancos de dados Amazon RDS e Amazon Aurora edição compatível com o PostgreSQL, defina o parâmetro `shared_preload_libraries` como `pglogical` no mesmo grupo de parâmetros do RDS.

   b. Reinicie o banco de dados de origem PostgreSQL.

   c. No banco de dados PostgreSQL, execute o comando `create extension pglogical;`.

2. Execute o comando a seguir para verificar se a instalação do pglogical foi bem-sucedida:

```
select * FROM pg_catalog.pg_extension
```

Agora você pode criar uma AWS DMS tarefa que realiza a captura de dados de alteração para o endpoint do banco de dados de origem do PostgreSQL.

> **Note**
>
> Se você não ativar o pglogical no banco de dados de origem PostgreSQL, o AWS DMS utilizará o plug-in `test_decoding` por padrão. Quando pglogical está habilitado para decodificação lógica, AWS DMS usa pglogical por padrão. Mas é possível definir o atributo

de conexão adicional, para que o `PluginName` utilize o plug-in `test_decoding` em vez disso.

Utilizar pontos de início nativos da CDC para configurar uma carga de CDC de uma origem PostgreSQL

Para ativar os pontos de início nativos da CDC com o PostgreSQL como origem, defina o atributo de conexão adicional `slotName` como o nome de um slot de replicação lógica existente ao criar o endpoint. Esse slot de replicação lógica mantém as alterações contínuas desde a hora da criação do endpoint, portanto, ele é compatível com a replicação de um ponto no tempo.

O PostgreSQL grava as alterações do banco de dados em arquivos WAL que são descartados somente depois que o AWS DMS faz a leitura das alterações do slot de replicação lógica com êxito. O uso de slots de replicação lógica pode impedir que as alterações registradas sejam excluídas antes de serem consumidas pelo mecanismo de replicação.

No entanto, dependendo da taxa de alteração e consumo, as alterações mantidas em um slot de replicação lógica podem causar a utilização elevado do disco. É recomendável definir alarmes de utilização de espaço na instância de origem PostgreSQL ao utilizar slots de replicação lógica. Para obter mais informações sobre como definir o atributo de conexão adicional `slotName`, consulte Configurações de endpoint e atributos extras de conexão (ECAs) ao usar o PostgreSQL como fonte de DMS.

O procedimento a seguir demonstra essa abordagem com mais detalhes.

Como utilizar um ponto inide início nativo da CDC para configurar uma carga de CDC de um endpoint de origem PostgreSQL

1. Identifique o slot de replicação lógica utilizado por uma tarefa de replicação anterior (uma tarefa pai) que você queira utilizar como ponto de partida. Consulte a visualização `pg_replication_slots` no banco de dados de origem para se certificar de que esse slot não tenha conexões ativas. Se isso acontecer, resolva-os e feche-os antes de continuar.

   Para as etapas a seguir, vamos supor que o slot de replicação lógica seja `abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef`.
2. Crie um novo endpoint de origem que inclua a configuração de atributo de conexão adicional a seguir.

```
slotName=abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef;
```

3. Crie uma nova tarefa somente para CDC usando o console AWS CLI ou AWS DMS a API. Por exemplo, utilizando a CLI, é possível executar o seguinte comando `create-replication-task`.

```
aws dms create-replication-task --replication-task-identifier postgresql-slot-name-
test
--source-endpoint-arn arn:aws:dms:us-
west-2:012345678901:endpoint:ABCD1EFGHIJK2LMNOPQRST3UV4
--target-endpoint-arn arn:aws:dms:us-
west-2:012345678901:endpoint:ZYX9WVUTSRQONM8LKJIHGF7ED6
--replication-instance-arn arn:aws:dms:us-
west-2:012345678901:rep:AAAAAAAAAAA5BB4CCC3DDDD2EE
--migration-type cdc --table-mappings "file://mappings.json" --cdc-start-position
 "4AF/B00000D0"
--replication-task-settings "file://task-pg.json"
```

No comando anterior, as seguintes opções são definidas:

- A opção `source-endpoint-arn` é definida como o valor criado na etapa 2.
- A opção `replication-instance-arn` é definida como o mesmo valor da tarefa pai da etapa 1.
- As opções `table-mappings` e `replication-task-settings` são definidas como os mesmos valores da tarefa pai na etapa 1.
- A opção `cdc-start-position` é definida como um valor de posição de início. Para localizar essa posição de início, consulte a visualização `pg_replication_slots` no banco de dados de origem ou visualize os detalhes do console da tarefa pai na etapa 1. Para ter mais informações, consulte Determinar um ponto de início nativo de CDC.

Para ativar o modo de início personalizado do CDC ao criar uma nova tarefa somente para CDC usando o AWS DMS console, faça o seguinte:

- Na seção Configurações da tarefa, em Modo de início da CDC para transações de origem, escolha Ativar o modo de início da CDC personalizado.
- Em Ponto de início da CDC personalizado para transações de origem, escolha Especificar um número de sequência de log. Especifique o número de alteração do sistema ou escolha

Especificar um ponto de verificação de recuperação e forneça um ponto de verificação de recuperação.

Quando essa tarefa do CDC é AWS DMS executada, gera um erro se o slot de replicação lógica especificado não existir. Ele também gerará um erro se a tarefa não for criada com uma configuração válida para `cdc-start-position`.

Ao utilizar pontos de início nativos da CDC com o plug-in pglogical e quiser utilizar um novo slot de replicação, conclua as etapas de configuração a seguir antes de criar uma tarefa de CDC.

Como utilizar um novo slot de replicação não criado anteriormente como parte de outra tarefa do DMS

1. Crie um slot de replicação, conforme mostrado a seguir:

```
SELECT * FROM pg_create_logical_replication_slot('replication_slot_name',
 'pglogical');
```

2. Depois que o banco de dados criar o slot de replicação, obtenha e anote os valores de restart_lsn e de confirmed_flush_lsn do slot:

```
select * from pg_replication_slots where slot_name like 'replication_slot_name';
```

   Observe que a posição de início da CDC nativo para uma tarefa de CDC criada após o slot de replicação não pode ser mais antiga do que o valor de confirmed_flush_lsn.

   Para obter informações sobre os valores de restart_lsn e de confirmed_flush_lsn, consulte pg_replication_slots

3. Crie um nó pglogical.

```
SELECT pglogical.create_node(node_name := 'node_name', dsn := 'your_dsn_name');
```

4. Crie dois conjuntos de replicação utilizando o perfil `pglogical.create_replication_set`. O primeiro conjunto de replicação rastreia as atualizações e as exclusões das tabelas que têm chaves primárias. O segundo conjunto de replicação rastreia somente inserções e tem o mesmo nome do primeiro conjunto de replicação, com o prefixo 'i' adicionado.

```
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. Adicione uma tabela ao conjunto de réplicas.

```
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. Defina o atributo de conexão adicional (ECA) a seguir ao criar o endpoint de origem.

```
PluginName=PGLOGICAL;slotName=slot_name;
```

Agora é possível criar uma tarefa somente de CDC com um ponto de início nativo do PostgreSQL utilizando o novo slot de replicação. Para obter mais informações sobre o plug-in pglogical, consulte a Documentação do pglogical 3.7

Migrando do PostgreSQL para o PostgreSQL usando AWS DMS

Quando você migra de um mecanismo de banco de dados diferente do PostgreSQL para um banco de dados PostgreSQL, é quase sempre a melhor ferramenta AWS DMS de migração a ser usada. No entanto, ao migrar de um banco de dados PostgreSQL para um banco de dados PostgreSQL, as ferramentas do PostgreSQL podem ser mais eficazes.

Utilize ferramentas nativas do PostgreSQL para migrar dados

É recomendável utilizar ferramentas nativas de migração do banco de dados PostgreSQL, como pg_dump, nas seguintes condições:

- Quando há uma migração homogênea, na qual você migra de um banco de dados PostgreSQL de origem para um banco de dados PostgreSQL de destino.
- Quando for migrar um banco de dados inteiro.
- As ferramentas nativas permitem que você migre os dados com um tempo mínimo de inatividade.

O utilitário pg_dump utiliza o comando COPY para criar um esquema e um despejo de dados de um banco de dados PostgreSQL. O script de despejo gerado pelo pg_dump carrega os dados em um banco de dados com o mesmo nome e recria as tabelas, os índices e as chaves estrangeiras. Para restaurar os dados para um banco de dados com outro nome, utilize o comando `pg_restore` e o parâmetro `-d`.

Se estiver migrando dados de um banco de dados de origem PostgreSQL sendo executado no EC2 para um destino Amazon RDS para PostgreSQL, você poderá utilizar o plug-in pglogical.

Para obter mais informações sobre como importar de um banco de dados PostgreSQL para o Amazon RDS para PostgreSQL ou para a edição do Amazon Aurora compatível com PostgreSQL, consulte https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/PostgreSQL.Procedural.Importing.html.

Utilizar o DMS para migrar dados do PostgreSQL para o PostgreSQL

AWS DMS pode migrar dados, por exemplo, de um banco de dados PostgreSQL de origem que está no local para uma instância de destino do Amazon RDS for PostgreSQL ou do Aurora PostgreSQL. Os tipos de dados do PostgreSQL fundamentais ou básicos geralmente migram com êxito.

> **Note**
>
> Ao replicar tabelas particionadas de uma origem PostgreSQL para o destino PostgreSQL, você não precisa mencionar a tabela pai como parte dos critérios de seleção na tarefa do DMS. Mencionar a tabela principal faz com que os dados sejam duplicados nas tabelas filho no destino, possivelmente causando uma violação de PK. Ao selecionar apenas as tabelas filho nos critérios de seleção do mapeamento de tabelas, a tabela pai é preenchida automaticamente.

Os tipos de dados compatíveis com o banco de dados de origem, mas que não são compatíveis com o destino, podem não ser migrados com êxito. AWS DMS transmite alguns tipos de dados como cadeias de caracteres se o tipo de dados for desconhecido. Alguns tipos de dados, como XML e JSON, podem ser migrados com êxito como arquivos pequenos, mas podem falhar se forem documentos grandes.

Ao executar a migração do tipo de dados, lembre-se de que:

- Em alguns casos, o tipo de dados PostgreSQL NUMERIC(p,s) não especifica nenhuma precisão e escala. Nas versões 3.4.2 e anteriores do DMS, o DMS utiliza uma precisão de 28 e uma

escala de 6 por padrão, NUMERIC(28,6). Por exemplo, o valor 0,611111104488373 da origem é convertido em 0,611111 no destino do PostgreSQL.

- Uma tabela com o tipo de dados ARRAY deve ter uma chave primária. Uma tabela com um tipo de dados ARRAY sem uma chave primária é suspensa durante a carga máxima.

A tabela a seguir mostra os tipos de dados do PostgreSQL de origem e se podem ser migrados com êxito.

Tipo de dados	Será migrado com êxito	Migra parcialmente	Não migra	Comentários
INTEGER	X			
SMALLINT	X			
BIGINT	X			
NUMERIC/DECIMAL(p,s)		X		Em que 0<p<39 e 0<s
NUMERIC/DECIMAL		X		Em que p>38 ou p=s=0
REAL	X			
DOUBLE	X			
SMALLSERIAL	X			
SERIAL	X			
BIGSERIAL	X			
MONEY	X			
CHAR		X		Sem precisão especificada
CHAR(n)	X			

Tipo de dados	Será migrado com êxito	Migra parcialmente	Não migra	Comentários
VARCHAR		X		Sem precisão especificada
VARCHAR(n)	X			
TEXT	X			
BYTEA	X			
TIMESTAMP	X			Os valores infinitos positivos e negativos são truncados para '9999-12-31 23:59:59' e '4713-01-01 00:00:00 BC', respectivamente.
TIMESTAMP WITH TIME ZONE		X		
DATA	X			
TIME	X			
TIME WITH TIME ZONE		X		
INTERVAL		X		
BOOLEAN	X			
ENUM			X	

Tipo de dados	Será migrado com êxito	Migra parcialmente	Não migra	Comentários
CIDR	X			
INET			X	
MACADDR			X	
TSVECTOR			X	
TSQUERY			X	
XML		X		
POINT	X			Tipo de dados espaciais do PostGIS
LINE			X	
LSEG			X	
BOX			X	
PATH			X	
POLYGON	X			Tipo de dados espaciais do PostGIS
CIRCLE			X	
JSON		X		
ARRAY	X			Requer chave primária

Tipo de dados	Será migrado com êxito	Migra parcialmente	Não migra	Comentários
COMPOSITE			X	
RANGE			X	
LINESTRING	X			Tipo de dados espaciais do PostGIS
MULTIPOINT	X			Tipo de dados espaciais do PostGIS
MULTILINESTRING	X			Tipo de dados espaciais do PostGIS
MULTIPOLYGON	X			Tipo de dados espaciais do PostGIS
GEOMETRYCOLLECTION	X			Tipo de dados espaciais do PostGIS

Migrar tipos de dados espaciais do PostGIS

Dados espaciais identificam a informação de geometria de um objeto ou local no espaço. Os bancos de dados relacionais de objetos PostgreSQL são compatíveis com os tipos de dados espaciais do PostGIS.

Antes de migrar objetos de dados espaciais do PostgreSQL, verifique se o plug-in PostGIS está ativado no nível global. Isso garante a AWS DMS criação exata das colunas de dados espaciais de origem para a instância de banco de dados de destino do PostgreSQL.

Para AWS DMS migrações homogêneas de PostgreSQL para PostgreSQL, suporta a migração de tipos e subtipos de objetos de dados geométricos e geográficos (coordenadas geodésicas) PostGIS, como os seguintes:

- POINT
- LINESTRING
- POLYGON
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION

Migração do Babelfish para o Amazon Aurora PostgreSQL usando AWS DMS

Você pode migrar as tabelas de origem do PostgreSQL do Babelfish for Aurora para qualquer endpoint de destino compatível usando o. AWS DMS

Ao criar seu endpoint de AWS DMS origem usando o console do DMS, a API ou os comandos da CLI, você define a origem como Amazon Aurora PostgreSQL e o nome do banco de dados como. **`babelfish_db`** Na seção Configurações do Endpoint, verifique se o DatabaseModeestá definido como Babelfish e BabelfishDatabaseNameestá definido como o nome do banco de dados Babelfish T-SQL de origem. Em vez de usar a porta TCP do Babelfish**`1433`**, use a porta TCP do Aurora PostgreSQL. **`5432`**

Você deve criar suas tabelas antes de migrar dados para garantir que o DMS use os tipos de dados e metadados de tabela corretos. Se você não criar suas tabelas no destino antes de executar a migração, o DMS poderá criar as tabelas com tipos de dados e permissões incorretos.

Adicionar regras de transformação à tarefa de migração

Ao criar uma tarefa de migração para uma fonte do Babelfish, você precisa incluir regras de transformação que garantam que o DMS use as tabelas de destino pré-criadas.

Se você definiu o modo de migração de vários bancos de dados ao definir seu cluster Babelfish para PostgreSQL, adicione uma regra de transformação que renomeia o nome do esquema para o esquema T-SQL. Por exemplo, se o nome do esquema T-SQL for e o nome do esquema do Babelfish para PostgreSQL fordbo, renomeie o esquema para usar uma regra de `mydb_dbo` transformação. dbo Para encontrar o nome do esquema PostgreSQL, consulte a arquitetura Babelfish no Guia do usuário do Amazon Aurora.

Se você usa o modo de banco de dados único, não precisa usar uma regra de transformação para renomear esquemas de banco de dados. Os nomes dos esquemas do PostgreSQL têm one-to-one um mapeamento para os nomes dos esquemas no banco de dados T-SQL.

O exemplo de regra de transformação a seguir mostra como renomear o nome do esquema de `mydb_dbo` back para: 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": []
        }
    ]
}
```

Limitações para usar um endpoint de origem PostgreSQL com tabelas Babelfish

As seguintes limitações se aplicam ao usar um endpoint de origem do PostgreSQL com tabelas do Babelfish:

- O DMS suporta apenas a migração do Babelfish versão 16.2/15.6 e posterior e do DMS versão 3.5.3 e posterior.
- O DMS não replica as alterações na definição da tabela do Babelfish para o endpoint de destino. Uma solução alternativa para essa limitação é primeiro aplicar as alterações na definição da tabela no destino e, em seguida, alterar a definição da tabela na fonte do Babelfish.
- Quando você cria tabelas do Babelfish com o tipo de dados BYTEA, o DMS as converte no tipo de `varbinary(max)` dados ao migrar para o SQL Server como destino.
- O DMS não oferece suporte ao modo LOB completo para tipos de dados binários. Em vez disso, use o modo LOB limitado para tipos de dados binários.
- O DMS não suporta a validação de dados para o Babelfish como fonte.
- Para a configuração da tarefa do modo de preparação da tabela de destino, use somente os modos Não fazer nada ou Truncar. Não utilize o modo Abandonar tabelas no destino. Ao usar Drop tables on target, o DMS pode criar as tabelas com tipos de dados incorretos.
- Ao usar a replicação contínua (CDC ou carga total e CDC), defina o atributo de conexão `PluginName` extra (ECA) como. `TEST_DECODING`

Removendo AWS DMS artefatos de um banco de dados de origem do PostgreSQL

Para capturar eventos DDL, AWS DMS cria vários artefatos no banco de dados PostgreSQL quando uma tarefa de migração é iniciada. Quando a tarefa é concluída, é recomendável remover esses artefatos.

Para remover os artefatos, execute os comandos a seguir (na ordem em que são exibidos), em que `{AmazonRDSMigration}` é o esquema no qual os artefatos foram criados: O descarte de um esquema deve ser feito com extremo cuidado. Nunca descarte um esquema operacional, especialmente um esquema que seja público.

```
drop event trigger awsdms_intercept_ddl;
```

O trigger do evento não pertence a um esquema específico.

```
drop function {AmazonRDSMigration}.awsdms_intercept_ddl()
```

```
drop table {AmazonRDSMigration}.awsdms_ddl_audit
drop schema {AmazonRDSMigration}
```

Definições de configuração adicionais ao utilizar um banco de dados PostgreSQL como origem do DMS

É possível adicionar definições de configuração ao migrar dados de um banco de dados PostgreSQL de duas maneiras:

- É possível adicionar valores ao atributo de conexão adicional para capturar eventos de DDL e especificar o esquema no qual os artefatos de banco de dados DDL operacionais são criados. Para ter mais informações, consulte Configurações de endpoint e atributos extras de conexão (ECAs) ao usar o PostgreSQL como fonte de DMS.
- É possível substituir os parâmetros da string de conexão. Escolha esta opção para realizar uma das seguintes ações:
  - Especifique AWS DMS os parâmetros internos. Esses parâmetros são raramente necessários e, portanto, não são expostos na interface do usuário.
  - Especifique valores de passagem (passthru) para o cliente de banco de dados específico. AWS DMS inclui parâmetros de passagem na cadeia de conexão passada para o cliente do banco de dados.
- Ao utilizar o parâmetro `REPLICA IDENTITY` em nível de tabela nas versões 9.4 e superiores do PostgreSQL, é possível controlar as informações gravadas em logs de gravação antecipada (WALs). Em particular, ele faz isso para WALs que identificam linhas que são atualizadas ou excluídas. O `REPLICA IDENTITY FULL` registra os valores antigos de todas as colunas na linha. Utilize `REPLICA IDENTITY FULL` com cuidado para cada tabela, já que o `FULL` gera uma quantidade adicional de WALs que podem não ser necessários. Para obter mais informações, consulte ALTER TABLE-REPLICA IDENTITY

Usando a configuração de MapBooleanAsBoolean endpoint do PostgreSQL

É possível utilizar as configurações de endpoint do PostgreSQL para mapear um booleano como um booleano da origem PostgreSQL para um destino Amazon Redshift. Por padrão, um tipo BOOLEAN é migrado como varchar(5). É possível especificar `MapBooleanAsBoolean` para permitir que o PostgreSQL migre o tipo booleano como booleano, conforme mostrado no exemplo a seguir.

```
--postgre-sql-settings '{"MapBooleanAsBoolean": true}'
```

Observe que você deve definir essa configuração nos endpoints de origem e de destino para que ela tenha efeito.

Como o MySQL não tem um tipo BOOLEAN, utilize uma regra de transformação em vez dessa configuração ao migrar dados BOOLEAN para o MySQL.

Configurações de endpoint e atributos extras de conexão (ECAs) ao usar o PostgreSQL como fonte de DMS

Você pode usar configurações de endpoint e atributos extras de conexão (ECAs) para configurar seu banco de dados de origem do PostgreSQL. Você especifica as configurações do endpoint ao criar o endpoint de origem usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint e ECAs que você pode usar com o PostgreSQL como fonte.

Nome do atributo	Descrição
`CaptureDDLs`	Para capturar eventos DDL, AWS DMS cria vários artefatos no banco de dados PostgreSQL quando a tarefa é iniciada. É possível remover esses artefatos posterior mente, conforme descrito em Removendo AWS DMS artefatos de um banco de dados de origem do PostgreSQ L. Se o valor estiver definido como falso, você não precisará criar tabelas ou acionadores no banco de dados de origem. Os eventos de DDL transmitidos são capturados. Valor padrão: `true` Valores válidos: verdadeiro/falso Exemplo: `--postgre-sql-settings '{"Captur eDDLs": true}'`

Nome do atributo	Descrição
`ConsumeMonotonicEvents`	Utilizado para controlar como as transações monolíticas com números de sequência de log (LSNs) duplicados são replicadas. Quando esse parâmetro é `false`, os eventos com LSNs duplicados são consumidos e replicados no destino. Quando esse parâmetro é `true`, somente o primeiro evento é replicado, enquanto eventos com LSNs duplicados não são consumidos nem replicados no destino. Valor padrão: `false` Valores válidos: falso/verdadeiro Exemplo: `--postgre-sql-settings '{"ConsumeMonotonicEvents": true}'`
`DdlArtifactsSchema`	Define o esquema no qual os artefatos do banco de dados da DDL operacional são criados. Valor padrão: público Valores válidos: string Exemplo: `--postgre-sql-settings '{"DdlArtifactsSchema": "  xyzddlschema "}'`
`ExecuteTimeout`	Define o tempo limite da instrução do cliente para a instância do PostgreSQL, em segundos. O valor padrão é de 60 segundos. Exemplo: `--postgre-sql-settings '{"ExecuteTimeout": 100}'`

Nome do atributo	Descrição
`FailTasksOnLobTruncation`	Quando definido como `true`, esse valor causará uma falha na tarefa, se o tamanho real de uma coluna LOB for maior que o `LobMaxSize` especificado. Se a tarefa for definida como modo LOB limitado e essa opção estiver definida como true, a tarefa falhará em vez de truncar os dados de LOB. Valor padrão: falso Valores válidos: booleano Exemplo: `--postgre-sql-settings '{"FailTasksOnLobTruncation": true}'`
`fetchCacheSize`	Esse atributo de conexão adicional (ECA) define o número de linhas que o cursor buscará durante a operação de carga máxima. Dependendo dos recursos disponíveis na instância de replicação, é possível ajustar o valor para cima ou para baixo. Valor padrão: `10000` Valores válidos: número Exemplo de ECA: `fetchCacheSize=10000;`
`HeartbeatFrequency`	Define a frequência de pulsação WAL (em minutos). Valor padrão: `5` Valores válidos: número Exemplo: `--postgre-sql-settings '{"HeartbeatFrequency": 1}'`

Nome do atributo	Descrição
`HeartbeatSchema`	Define o esquema no qual os artefatos de pulsação são criados. Valor padrão: `public` Valores válidos: string Exemplo: `--postgre-sql-settings '{"HeartbeatSchema": "xyzheartbeatschema"}'`
`MapJsonbAsClob`	Por padrão, AWS DMS mapeia JSONB para NCLOB. É possível especificar `MapJsonbAsClob` para permitir que o PostgreSQL migre o tipo JSONB como CLOB. Exemplo: `--postgre-sql-settings='{"MapJsonbAsClob": "true"}'`
`MapLongVarcharAs`	Por padrão, AWS DMS mapeia VARCHAR para WSTRING. É possível especificar `MapLongVarcharAs` para permitir que o PostgreSQL migre o tipo VARCHAR(N) (em que N é maior que 16387) para os seguintes tipos: • WSTRING • CLOB • NCLOB Exemplo: `--postgre-sql-settings='{"MapLongVarcharAs": "CLOB"}'`

Nome do atributo	Descrição
`MapUnboundedNumericAsString`	Esse parâmetro trata colunas com tipos de dados NUMERIC ilimitados como STRING para migrar com sucesso sem perder a precisão do valor numérico. Utilize esse parâmetro somente para replicação da origem do PostgreSQL para o destino do PostgreSQL ou bancos de dados compatíveis com o PostgreSQL. Valor padrão: `false` Valores válidos: `false`/`true` Exemplo: `--postgre-sql-settings '{"MapUnboundedNumericAsString": true}'` A utilização desse parâmetro pode resultar em alguma degradação do desempenho da replicação devido à transformação de numérico para string e de volta para numérico. Esse parâmetro é compatível para utilização pelo DMS versão 3.4.4 e superior **Note** Utilize `MapUnboundedNumericAsString` somente em endpoints de origem e de destino do PostgreSQL juntos. A utilização de `MapUnboundedNumericAsString` em endpoints PostgreSQL de origem restringe a precisão a 28 durante a CDC. A utilização de `MapUnboundedNumericAsString` em endpoints de destino migra os dados com Precisão 28 Escala 6. Não utilize `MapUnboundedNumericAsString` com destinos que não sejam do PostgreSQL.

Nome do atributo	Descrição
`PluginName`	Especifica o plug-in a ser utilizado para criar um slot de replicação. Valores válidos: `pglogical` , `test_decoding` Exemplo: `--postgre-sql-settings '{"Plugin Name": "test_decoding"}'`

Nome do atributo	Descrição
SlotName	Define o nome de um slot de replicação lógica criado anteriormente para uma carga CDC da instância do PostgreSQL de origem. Quando usado com o parâmetro de CdcStartPosition solicitação de AWS DMS API, esse atributo também permite o uso de pontos iniciais nativos do CDC. O DMS verifica se o slot de replicação lógica especificado existe antes de iniciar a tarefa de carregamento da CDC. Ele também verifica se a tarefa foi criada com uma configuração válida de CdcStartPosition . Se o slot especificado não existir ou a tarefa não tiver uma configuração válida CdcStartPosition , o DMS gerará um erro. Para obter mais informações sobre como configurar o parâmetro de solicitação CdcStartPosition , consulte Determinar um ponto de início nativo de CDC. Para obter mais informações sobre como utilizar CdcStartPosition , consulte a documentação das operações da API CreateReplicationTask , StartReplicationTask e ModifyReplicationTask na Referência da API do AWS Database Migration Service. Valores válidos: string Exemplo: --postgre-sql-settings '{"SlotName": "abc1d2efghijk_34567890_z0yx98w7_6v54_32ut_1srq_1a2b34c5d67ef"}'
unboundedVarcharMaxSize	Esse Atributo de Conexão Extra (ECA) define o tamanho máximo de uma coluna de dados definida como tipo VarChar sem um especificador de comprimento máximo. O padrão é 8000 bytes. O valor máximo é 10485760 bytes.

Limitações ao utilizar um banco de dados PostgreSQL como origem do DMS

As limitações a seguir se aplicam à utilização do PostgreSQL como uma origem do AWS DMS:

- AWS DMS não funciona com o Amazon RDS for PostgreSQL 10.4 ou o Amazon Aurora PostgreSQL 10.4 como origem ou destino.
- Uma tabela capturada deve ter uma chave primária. Se uma tabela não tiver uma chave primária, AWS DMS ignora as operações de registro DELETE e UPDATE dessa tabela. Como solução alternativa, consulte Ativar a captura de dados de alteração (CDC) utilizando a replicação lógica.

  Observação: não é recomendável migrar sem uma chave primária/índice exclusivo, caso contrário, limitações adicionais se aplicarão, como a capacidade de aplicação em “NÃO” lote, a capacidade de LOB completo, a validação de dados e a incapacidade de replicar para o destino do Redshift de forma eficiente.
- AWS DMS ignora uma tentativa de atualizar um segmento de chave primária. Nesses casos, o destino identifica a atualização como uma que não atualizou nenhuma linha. No entanto, como os resultados da atualização de uma chave primária no PostgreSQL são imprevisíveis, nenhum registro é gravado na tabela de exceções.
- AWS DMS não suporta a opção Iniciar alterações do processo a partir da execução do carimbo de data/hora.
- AWS DMS não replica as alterações resultantes de operações de partição ou subpartição (`ADD`,`DROP`, ou`TRUNCATE`).
- A replicação de várias tabelas com o mesmo nome, onde cada nome tem maiúsculas e minúsculas diferentes (por exemplo, table1, TABLE1 e Table1), pode causar um comportamento imprevisível. Devido a esse problema, AWS DMS não oferece suporte a esse tipo de replicação.
- Na maioria dos casos, AWS DMS oferece suporte ao processamento de alterações das instruções DDL CREATE, ALTER e DROP para tabelas. AWS DMS não suporta esse processamento de alterações se as tabelas forem mantidas em uma função interna ou bloco de corpo de procedimento ou em outras construções aninhadas.

  Por exemplo, a seguinte alteração não é capturada.

```
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;
$$;
```

- Atualmente, os tipos de dados `boolean` em uma origem do PostgreSQL são migrados para um destino do SQL Server como o tipo de dados `bit` com valores inconsistentes. Como solução alternativa, pré-crie a tabela com um tipo de `VARCHAR(1)` dados para a coluna (ou faça com que o AWS DMS crie a tabela). Depois, deixe o processamento downstream tratar um "F" como Falso e um "T" como Verdadeiro.
- AWS DMS não oferece suporte ao processamento de alterações das operações TRUNCATE.
- O tipo de dados OID LOB não é migrado para o destino.
- AWS DMS suporta o tipo de dados PostGIS somente para migrações homogêneas.
- Se a origem for um banco de dados PostgreSQL on-premises ou em uma instância do Amazon EC2, verifique se o plug-in de saída test_decoding está instalado no endpoint de origem. É possível encontrar esse plug-in no pacote contrib do PostgreSQL. Para obter mais informações sobre o plug-in de teste de decodificação, consulte a documentação do PostgreSQL.
- AWS DMS não oferece suporte ao processamento de alterações para definir e cancelar a definição dos valores padrão da coluna (usando a cláusula ALTER COLUMN SET DEFAULT nas instruções ALTER TABLE).
- AWS DMS não oferece suporte ao processamento de alterações para definir a nulidade da coluna (usando a cláusula ALTER COLUMN [SET|DROP] NOT NULL nas instruções ALTER TABLE).
- Quando a replicação lógica está ativada, o número máximo de alterações mantidas na memória por transação é de 4 MB. Depois disso, as alterações são transferidas para o disco. Como resultado, `ReplicationSlotDiskUsage` aumenta e `restart_lsn` não avança até que a transação seja concluída ou interrompida e a reversão seja concluída. Como é uma transação longa, ela pode demorar muito tempo para reverter. Portanto, evite transações de longa duração ou muitas subtransações quando a replicação lógica estiver habilitada. Em vez disso, divida a transação em várias transações menores.

  Nas versões 13 e posteriores do Aurora PostgreSQL, você pode ajustar o `logical_decoding_work_mem` parâmetro para controlar quando o DMS derrama dados alterados para o disco. Para ter mais informações, consulte Derrame arquivos no Aurora PostgreSQL.
- Uma tabela com o tipo de dados ARRAY deve ter uma chave primária. Uma tabela com um tipo de dados ARRAY sem uma chave primária é suspensa durante a carga máxima.
- AWS DMS não oferece suporte à replicação de tabelas particionadas. Quando uma tabela particionada é detectada, ocorre o seguinte:

  - O endpoint relata uma lista de tabelas pai e filho.
  - AWS DMS cria a tabela no destino como uma tabela normal com as mesmas propriedades das tabelas selecionadas.
  - Se a tabela pai do banco de dados de origem tiver o mesmo valor de chave primária que suas tabelas filhas, um erro de "chave duplicada" será gerado.
- Para replicar tabelas particionadas de uma origem PostgreSQL para um destino PostgreSQL, primeiro crie manualmente as tabelas pai e filho no destino. Defina uma tarefa separada a fim de replicar nessas tabelas. Nesse caso, defina a configuração da tarefa como Truncar antes de carregar.
- O tipo de dados `NUMERIC` do PostgreSQL não tem um tamanho fixo. Na transferência de dados do tipo `NUMERIC` sem precisão e escala, o DMS utiliza `NUMERIC(28,6)` (uma precisão de 28 e escala 6) por padrão. Por exemplo, o valor 0,611111104488373 da origem é convertido em 0,611111 no destino do PostgreSQL.
- AWS DMS oferece suporte ao Aurora PostgreSQL Serverless V1 como fonte somente para tarefas de carga total. AWS DMS oferece suporte ao Aurora PostgreSQL Serverless V2 como fonte para carga total, carga total e tarefas CDC e somente CDC.
- AWS DMS não suporta a replicação de uma tabela com um índice exclusivo criado com uma função de coalescência.
- Ao utilizar o modo LOB, tanto a tabela de origem quanto a tabela de destino correspondente devem ter uma chave primária idêntica. Se uma das tabelas não tiver uma chave primária, o resultado das operações de registro DELETE e UPDATE será imprevisível.
- Ao utilizar o recurso Carga paralela, a segmentação de tabelas de acordo com partições ou subpartições não é compatível. Para obter mais informações sobre Carga paralela, consulte Utilizar carga paralela para tabelas, visualizações e coleções selecionadas.
- AWS DMS não oferece suporte a restrições diferidas.
- AWS DMS a versão 3.4.7 suporta o PostgreSQL 14.x como fonte com estas limitações:
  - AWS DMS não suporta o processamento de alterações de confirmações em duas fases.
  - AWS DMS não oferece suporte à replicação lógica para transmitir transações longas em andamento.
- AWS DMS não oferece suporte ao CDC para Amazon RDS Proxy for PostgreSQL como fonte.
- Ao utilizar filtros de origem que não contêm uma coluna de chave primária, as operações `DELETE` não serão capturadas.

- Se o banco de dados de origem também for um destino para outro sistema de replicação de terceiros, as alterações de DDL podem não ser migradas durante a CDC. Porque essa situação pode impedir que o acionador de eventos `awsdms_intercept_ddl` seja acionado. Para contornar a situação, modifique esse acionador no banco de dados de origem da seguinte maneira:

```
alter event trigger awsdms_intercept_ddl enable always;
```

- AWS DMS não oferece suporte ao CDC para o cluster de banco de dados Multi-AZ do Amazon RDS para PostgreSQL como fonte, pois os clusters de banco de dados Multi-AZ do RDS for PostgreSQL não oferecem suporte à replicação lógica.

Tipos de dados de origem para o PostgreSQL

A tabela a seguir mostra os tipos de dados de origem do PostgreSQL que são compatíveis com o AWS DMS uso e o mapeamento AWS DMS padrão para os tipos de dados.

Para obter informações sobre como visualizar o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que você está usando.

Para obter informações adicionais sobre AWS DMS os tipos de dados, consulteTipos de dados do AWS Database Migration Service.

Tipos de dados do PostgreSQL	Tipos de dados do DMS
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
NUMERIC (p,s)	Se a precisão estiver entre 0 e 38, use NUMERIC. Se a precisão for maior ou igual a 39, use STRING.
DECIMAL(P,S)	Se a precisão estiver entre 0 e 38, use NUMERIC.

Tipos de dados do PostgreSQL	Tipos de dados do DMS
	Se a precisão for maior ou igual a 39, use STRING.
REAL	REAL4
DOUBLE	REAL8
SMALLSERIAL	INT2
SERIAL	INT4
BIGSERIAL	INT8
MONEY	NUMERIC(38,4) O tipo de dados MONEY é mapeado para FLOAT no SQL Server.
CHAR	WSTRING (1)
CHAR(N)	WSTRING (n)
VARCHAR(N)	WSTRING (n)
TEXT	NCLOB
CITEXTO	NCLOB
BYTEA	BLOB
TIMESTAMP	DATETIME
TIMESTAMP WITH TIME ZONE	DATETIME
DATA	DATA
TIME	TIME
TIME WITH TIME ZONE	TIME

Tipos de dados do PostgreSQL	Tipos de dados do DMS
INTERVAL	STRING (128)—1 YEAR, 2 MONTHS, 3 DAYS, 4 HOURS, 5 MINUTES, 6 SECONDS
BOOLEAN	CHAR (5) falso ou verdadeiro
ENUM	STRING (64)
CIDR	STRING (50)
INET	STRING (50)
MACADDR	STRING (18)
BIT (n)	STRING (n)
BIT VARYING (n)	STRING (n)
UUID	STRING
TSVECTOR	CLOB
TSQUERY	CLOB
XML	CLOB
POINT	STRING (255) "(x,y)"
LINE	STRING (255) "(x,y,z)"
LSEG	STRING (255) "((x1,y1),(x2,y2))"
BOX	STRING (255) "((x1,y1),(x2,y2))"
PATH	CLOB "((x1,y1),(xn,yn))"
POLYGON	CLOB "((x1,y1),(xn,yn))"
CIRCLE	STRING (255) "(x,y),r"
JSON	NCLOB

Tipos de dados do PostgreSQL	Tipos de dados do DMS
JSONB	NCLOB
ARRAY	NCLOB
COMPOSITE	NCLOB
HSTORE	NCLOB
INT4RANGE	STRING (255)
INT8RANGE	STRING (255)
NUMRANGE	STRING (255)
STRRANGE	STRING (255)

Como trabalhar com tipos de dados de origem LOB para PostgreSQL

Os tamanhos de colunas do PostgreSQL afetam a conversão de tipos de dados LOB do PostgreSQL em tipos de dados do AWS DMS . Para trabalhar com isso, siga estas etapas para os seguintes tipos de dados do AWS DMS :

- BLOB: defina Limitar tamanho de LOB em o valor de Tamanho máximo de LOB (KB) na criação da tarefa.
- CLOB: a replicação trata cada caractere como um caractere UTF8. Portanto, localize o tamanho do texto de caracteres mais longo na coluna, mostrado aqui como `max_num_chars_text`. Utilize esse tamanho para especificar o valor de Limitar o tamanho do LOB em. Se os dados incluírem caracteres de 4 bytes, multiplique por 2 para especificar o valor de Limitar o valor de LOB em, que é em bytes. Nesse caso, Limitar o tamanho de LOB para será igual a `max_num_chars_text` multiplicado por 2.
- NCLOB: a replicação trata cada caractere como um caractere de byte duplo. Portanto, localize o tamanho do texto de caracteres mais longo na coluna (`max_num_chars_text`) e multiplique por 2. Faça isso para especificar o valor de Limitar o tamanho do LOB em. Nesse caso, Limitar o tamanho de LOB para será igual a `max_num_chars_text` multiplicado por 2. Se os dados incluírem caracteres de 4 bytes, multiplique por 2 novamente. Nesse caso, Limitar o tamanho de LOB para será igual a `max_num_chars_text` multiplicado por 4.

Utilizar um banco de dados compatível com MySQL como origem do AWS DMS

Você pode migrar dados de qualquer banco de dados compatível com MySQL (MySQL, MariaDB ou Amazon Aurora MySQL) usando o Database Migration Service. AWS

Para obter informações sobre as versões do MySQL compatíveis com o AWS DMS como origem, consulte Fontes para AWS DMS.

Você pode usar o SSL para criptografar conexões entre o endpoint compatível com MySQL e a instância de replicação. Para obter mais informações sobre o uso do SSL com um endpoint compatível com MySQL, consulte Usando SSL com AWS Database Migration Service.

Nas seções a seguir, o termo "autogerenciado" se aplica a qualquer banco de dados instalado on-premises ou no Amazon EC2. O termo "gerenciado pela AWS" se aplica a qualquer banco de dados no Amazon RDS, no Amazon Aurora, ou no Amazon S3.

Para obter detalhes adicionais sobre como trabalhar com bancos de dados compatíveis com MySQL AWS DMS, consulte as seções a seguir.

Tópicos

Migração do MySQL para o MySQL utilizando o AWS DMS

Para uma migração heterogênea, em que você está migrando de um mecanismo de banco de dados diferente do MySQL para um banco de dados MySQL AWS DMS , é quase sempre a melhor ferramenta de migração a ser usada. Mas para uma migração homogênea, na qual você está

migrando de um banco de dados MySQL para um banco de dados MySQL, é recomendável utilizar um projeto de migração de dados homogênea. As migrações de dados homogêneas utilizam ferramentas de banco de dados nativas para fornecer um desempenho e precisão aprimorados da migração de dados em comparação com o AWS DMS.

Usando qualquer banco de dados compatível com MySQL como fonte para AWS DMS

Antes de começar a trabalhar com um banco de dados MySQL como fonte para AWS DMS, verifique se você tem os seguintes pré-requisitos. Esses pré-requisitos se aplicam a fontes autogerenciadas ou gerenciadas. AWS

Você deve ter uma conta para AWS DMS que tenha a função de administrador de replicação. O perfil precisa dos seguintes privilégios:

- REPLICATION CLIENT: este privilégio é necessário apenas para tarefas de CDC. Em outras palavras, full-load-only as tarefas não exigem esse privilégio.
- REPLICATION SLAVE: este privilégio é necessário apenas para tarefas de CDC. Em outras palavras, full-load-only as tarefas não exigem esse privilégio.
- SUPER: este privilégio é necessário somente nas versões do MySQL anteriores à versão 5.6.6.

O AWS DMS usuário também deve ter privilégios SELECT para as tabelas de origem designadas para replicação.

Usando um banco de dados autogerenciado compatível com MySQL como fonte para AWS DMS

É possível utilizar os bancos de dados autogerenciados a seguir, compatíveis com MySQL como origens para o 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

Para utilizar a CDC, ative o registro em log binário. Para habilitar o registro binário, os parâmetros a seguir devem ser configurados nos arquivos `my.ini` (Windows) ou `my.cnf` (UNIX) do MySQL.

Parâmetro	Valor
`server_id`	Defina este parâmetro com um valor maior ou igual a 1.
`log-bin`	Defina a rota para o arquivo de log binário, por exemplo `log-bin=E:\MySql_Logs\BinLog` . Não inclua a extensão do arquivo.
`binlog_format`	Defina este parâmetro como `ROW`. Essa configuração é recomendável durante a replicação porque, em certos casos, quando `binlog_format` está definido como `STATEMENT` , ele pode causar inconsistência ao replicar dados para o destino. O mecanismo de banco de dados também grava dados inconsistentes semelhantes no destino quando `binlog_format` está definido como `MIXED`, porque o mecanismo de banco de dados muda automaticamente para o registro em log baseado em `STATEMENT` , o que pode resultar na gravação de dados inconsistentes no banco de dados de destino.
`expire_logs_days`	Defina este parâmetro com um valor maior ou igual a 1. Para evitar o uso excessivo de espaço em disco, recomendamos que você não utilize o valor padrão de 0.
`binlog_checksum`	Defina esse parâmetro `NONE` para a versão 3.4.7 ou anterior do DMS.
`binlog_row_image`	Defina este parâmetro como `FULL`.
`log_slave_updates`	Defina este parâmetro como `TRUE` se você estiver utilizando uma réplica de leitura do MySQL ou MariaDB como origem.

Se sua origem NDB (cluster) utiliza o mecanismo de banco de dados, os parâmetros a seguir precisarão ser configurados para habilitar CDC em tabelas que utilizam esse mecanismo de armazenamento. Adicione essas alterações no arquivo `my.ini` do MySQL (Windows) ou `my.cnf` (UNIX).

Parâmetro	Valor
`ndb_log_bin`	Defina este parâmetro como `ON`. Este valor garante que as alterações em tabelas clusterizadas são registradas no log binário.
`ndb_log_u pdate_as_write`	Defina este parâmetro como `OFF`. Este valor impede o registro de instruções UPDATE como instruções INSERT no log binário.
`ndb_log_u pdated_only`	Defina este parâmetro como `OFF`. Este valor garante que o log binário contém a linha completa e não apenas as colunas alteradas.

Usando um banco AWS de dados compatível com MySQL gerenciado como fonte para AWS DMS

Você pode usar os seguintes bancos de dados compatíveis AWS com MySQL gerenciados como fontes para: AWS DMS

- MySQL Community Edition
- MariaDB Community Edition
- Amazon Aurora Edição Compatível com MySQL

Ao usar um banco AWS de dados compatível com MySQL gerenciado como fonte para AWS DMS, verifique se você tem os seguintes pré-requisitos para o CDC:

- Para ativar os logs binários para o RDS para MySQL e para o RDS para MariaDB, ative backups automáticos no nível da instância. Para ativar logs binários para um cluster do Aurora MySQL, altere a variável `binlog_format` no grupo de parâmetros.

  Para obter mais informações sobre a configuração de backups automáticos, consulte Trabalhar com backups automáticos no Guia do usuário do Amazon RDS.

  Para obter mais informações sobre como configurar o registro em log binário para um banco de dados Amazon RDS para MySQL, consulte Configuração do formato de registro em log binário no Guia do usuário do Amazon RDS.

Para obter mais informações sobre como configurar o registro em log binário para um cluster do Aurora MySQL, consulte Como faço para ativar o registro em log binário para meu cluster do Amazon Aurora MySQL?.

- Se você planejar utilizar a CDC, ative o registro em log binário. Para obter mais informações sobre como configurar o registro em log binário para um banco de dados Amazon RDS para MySQL, consulte Configuração do formato de registro em log binário no Guia do usuário do Amazon RDS.
- Certifique-se de que os registros binários estejam disponíveis para AWS DMS o. Como os bancos AWS de dados compatíveis com MySQL gerenciados eliminam os registros binários o mais rápido possível, você deve aumentar o tempo em que os registros permanecem disponíveis. Por exemplo, para aumentar a retenção de log para 24 horas, execute o comando a seguir.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

- Defina o parâmetro `binlog_format` como `"ROW"`.

> **Note**
>
> No MySQL ou no MariaDB, `binlog_format` é um parâmetro dinâmico, portanto, você não precisa reinicializar para que o novo valor entre em vigor. No entanto, o novo valor só se aplicará às novas sessões. Se você alternar `ROW` como `binlog_format` para fins de replicação, o banco de dados ainda poderá criar registros em logs binários subsequentes utilizando o formato `MIXED`, se essas sessões tiverem iniciado antes da alteração do valor. Isso pode AWS DMS impedir a captura adequada de todas as alterações no banco de dados de origem. Ao alterar a configuração `binlog_format` em um banco de dados MariaDB ou MySQL, reinicie o banco de dados para fechar todas as sessões existentes ou reinicie qualquer aplicação executando operações DML (linguagem de manipulação de dados). Forçar seu banco de dados a reiniciar todas as sessões após alterar o `binlog_format` parâmetro para `ROW` garantirá que seu banco de dados grave todas as alterações subsequentes no banco de dados de origem usando o formato correto, para que AWS DMS possa capturar essas alterações adequadamente.

- Defina o parâmetro `binlog_row_image` como `"Full"`.

- Defina o `binlog_checksum` parâmetro `"NONE"` para a versão 3.4.7 ou anterior do DMS. Para obter mais informações sobre a configuração de parâmetros no Amazon RDS MySQL, consulte Trabalhar com backups automáticos no Guia do usuário do Amazon RDS.
- Se estiver utilizando uma réplica de leitura do Amazon RDS MySQL ou do Amazon RDS MariaDB como origem, ative os backups na réplica de leitura e garanta que o parâmetro `log_slave_updates` esteja definido como `TRUE`.

Limitações no uso de um banco de dados MySQL como fonte para AWS DMS

Ao utilizar um banco de dados MySQL como uma origem, considere o seguinte:

- A captura de dados de alteração (CDC) não é compatível com o Amazon RDS MySQL 5.5 ou inferior. Para o Amazon RDS MySQL, utilize a versão 5.6, 5.7 ou 8.0 para ativar a CDC. A CDC é compatível com origens do MySQL 5.5 autogerenciado.
- Para CDC, `CREATE TABLE`, `ADD COLUMN` e `DROP COLUMN` que alteram o tipo de dados da coluna e `renaming a column` são compatíveis. No entanto, `DROP TABLE`, `RENAME TABLE` e as atualizações feitas em outros atributos, como o valor padrão da coluna, a nulidade da coluna, o conjunto de caracteres e assim por diante, não são compatíveis.
- Para tabelas particionadas na origem, quando você define o modo de preparação da tabela Target como Drop tables on target, AWS DMS cria uma tabela simples sem partições no destino do MySQL. Para migrar tabelas particionadas para uma tabela particionada no destino, pré-crie as tabelas particionadas no banco de dados MySQL de destino.
- Não há suporte para utilizar uma instrução `ALTER TABLE` *`table_name`* `ADD COLUMN` *`column_name`* para adicionar colunas ao início (FIRST) ou no meio de uma tabela (AFTER). As colunas são sempre adicionadas ao final da tabela.
- A CDC não é compatível quando um nome de tabela contém caracteres maiúsculos e minúsculos, e o mecanismo de origem está hospedado em um sistema operacional com nomes de arquivos que não diferenciam maiúsculas de minúsculas. Um exemplo é o Microsoft Windows ou o OS X que utilizam HFS+.
- Você pode usar a Edição Serverless v1 compatível com o Aurora MySQL para carga total, mas não pode usá-la para CDC. Isso ocorre porque não é possível ativar os pré-requisitos para o MySQL. Para obter mais informações, consulte Grupos de parâmetros e o Aurora Sem Servidor v1.

  A edição Serverless v2 compatível com o Aurora MySQL é compatível com CDC.

- O atributo AUTO_INCREMENT em uma coluna não é migrado para uma coluna do banco de dados de destino.
- A captura de alterações quando os logs binários não estão armazenados em armazenamento de bloco padrão não é compatível. Por exemplo, a CDC não funciona quando os logs binários são armazenados no Amazon S3.
- AWS DMS cria tabelas de destino com o mecanismo de armazenamento InnoDB por padrão. Se precisar utilizar um mecanismo de armazenamento diferente do InnoDB, crie a tabela manualmente e migre-a utilizando o modo Não executar nenhuma ação.
- Você não pode usar réplicas do Aurora MySQL como fonte, a AWS DMS menos que seu modo de tarefa de migração do DMS seja Migrar dados existentes — somente com carga total.
- Se a origem compatível com MySQL for interrompida durante a carga máxima, a tarefa do AWS DMS não será interrompida com um erro. A tarefa terminará com êxito, mas o destino poderá estar fora de sincronia com a origem. Se isso acontecer, reinicie a tarefa ou recarregue as tabelas afetadas.
- Índices criados em uma parte do valor de uma coluna não são migrados. Por exemplo, o índice CREATE INDEX first_ten_chars ON customer (name(10)) não é criado no destino.
- Em alguns casos, a tarefa é configurada para não replicar LOBs (” SupportLobs "é falso nas configurações da tarefa ou Não incluir colunas LOB é escolhido no console de tarefas). Nesses casos, AWS DMS não migra nenhuma coluna MEDIUMBLOB, LONGBLOB, MEDIUMTEXT e LONGTEXT para o destino.

  As colunas BLOB, TINYBLOB, TEXT e TINYTEXT não são afetadas e são migradas para o destino.
- Tabelas de dados temporais ou tabelas com versão do sistema não são compatíveis com bancos de dados MariaDB de origem e de destino.
- Ao migrar entre dois clusters do Amazon RDS Aurora MySQL, o endpoint de origem do RDS Aurora MySQL deve ser uma instância de leitura/gravação, não uma instância de réplica.
- AWS DMS atualmente não oferece suporte à migração de visualizações para o MariaDB.
- AWS DMS não suporta alterações de DDL para tabelas particionadas para MySQL. Para ignorar a suspensão de tabela para alterações de DDL da partição durante a CDC, defina `skipTableSuspensionForPartitionDdl` como `true`.
- AWS DMS só suporta transações XA na versão 3.5.0 e superior. As versões anteriores não oferecem suporte a transações XA. AWS DMS não suporta transações XA na versão 10.6 do MariaDB. Para obter mais informações, consulte the section called “Compatibilidade com transações XA” a seguir.

- AWS DMS não usa GTIDs para replicação, mesmo que os dados de origem os contenham.
- AWS DMS não oferece suporte à compressão de transações de log binário.
- AWS DMS não propaga eventos ON DELETE CASCADE e ON UPDATE CASCADE para bancos de dados MySQL usando o mecanismo de armazenamento InnoDB. Para esses eventos, o MySQL não gera log binário de eventos para refletir as operações em cascata nas tabelas secundárias. Consequentemente, não é AWS DMS possível replicar as alterações correspondentes nas tabelas secundárias. Para ter mais informações, consulte Índices, chaves estrangeiras ou atualizações ou exclusões em cascata não migrados.
- AWS DMS não captura alterações nas colunas computadas (`VIRTUALeGENERATED ALWAYS`). Para trabalhar com essa limitação, faça o seguinte.
  - Pré-crie a tabela de destino no banco de dados de destino e crie a tarefa AWS DMS com a configuração de tarefa de carga máxima `DO_NOTHING` ou `TRUNCATE_BEFORE_LOAD`.
  - Adicione uma regra de transformação para remover a coluna computada do escopo da tarefa. Para obter mais informações sobre regras transformação, consulte Regras de transformação e ações.

Compatibilidade com transações XA

Uma transação de arquitetura estendida (XA) é uma transação que pode ser utilizada para agrupar uma série de operações de vários recursos transacionais em uma única transação global confiável. Uma transação XA utiliza um protocolo de confirmação de duas fases. Em geral, a captura de alterações enquanto há transações XA abertas pode resultar em perda de dados. Se o banco de dados não utilizar transações XA, é possível ignorar essa permissão e a configuração `IgnoreOpenXaTransactionsCheck` utilizando o valor padrão `TRUE`. Para começar a replicar a partir de uma origem que possui transações XA, faça o seguinte:

- Certifique-se de que o usuário do AWS DMS endpoint tenha a seguinte permissão:

```
grant XA_RECOVER_ADMIN on *.* to 'userName'@'%';
```

- Defina a configuração do endpoint `IgnoreOpenXaTransactionsCheck` como `false`.

> **Note**
>
> AWS DMS não suporta transações XA no MariaDB Source DB versão 10.6.

Configurações de endpoint ao usar o MySQL como fonte para AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados MySQL de origem de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de origem usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--my-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o MySQL como origem.

Nome	Descrição
`EventsPollInterval`	Especifica a frequência com que se deve verificar o log binário para ver se há alterações/eventos novos quando o banco de dados está inativo. Valor padrão: 5 Valores válidos: 1 a 60 Exemplo: `--my-sql-settings '{"EventsPollInterval": 5}'` No exemplo, AWS DMS verifica as alterações nos registros binários a cada cinco segundos.
`ExecuteTimeout`	Para AWS DMS as versões 3.4.7 e superiores, define o tempo limite da instrução do cliente para um endpoint de origem do MySQL, em segundos. Valor padrão: 60 Exemplo: `--my-sql-settings '{"ExecuteTimeout": 1500}'`
`ServerTimezone`	Especifica o fuso horário para o banco de dados MySQL de origem.

Nome	Descrição
	Exemplo: `--my-sql-settings '{"Server Timezone": "  US/Pacific "}'`
`AfterConnectScript`	Especifica um script a ser executado imediatamente após a AWS DMS conexão com o endpoint. A tarefa de migração continuará em execução, independentemente se a instrução SQL for bem-sucedida ou não. Valores válidos: uma ou mais instruções SQL válidas, separadas por ponto e vírgula. Exemplo: `--my-sql-settings '{"AfterConnectScript": "ALTER SESSION SET CURRENT_SCHEMA=system"}'`
`CleanSrcMetadataOnMismatch`	Limpa e recria as informações dos metadados da tabela na instância de replicação quando ocorre uma incompatibilidade. Por exemplo, em uma situação em que a execução de um DDL alternativo na tabela pode resultar em diferentes informações sobre a tabela armazenada em cache na instância de replicação. Booliano. Valor padrão: `false` Exemplo: `--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'`

Nome	Descrição
`skipTableSuspensionForPartitionDdl`	AWS DMS não suporta alterações de DDL para tabelas particionadas para MySQL. Para AWS DMS as versões 3.4.6 e superiores, definir isso para `true` ignorar a suspensão da tabela para alterações de DDL de partição durante o CDC. AWS DMS ignora o partitioned-table-related DDL e continua processando outras alterações no log binário. Valor padrão: `false` Exemplo: `--my-sql-settings '{"skipTableSuspensionForPartitionDdl": true}'`
`IgnoreOpenXaTransactionsCheck`	Para AWS DMS as versões 3.5.0 e posteriores, especifica se as tarefas devem ignorar as transações XA abertas ao iniciar. Defina isso como `false` se a origem tiver transações XA. Valor padrão: `true` Exemplo: `--my-sql-settings '{"IgnoreOpenXaTransactionsCheck": false}'`

Tipos de dados de origem do MySQL

A tabela a seguir mostra os tipos de dados de origem do banco de dados MySQL que são suportados durante o uso AWS DMS e o mapeamento padrão dos tipos de AWS DMS dados.

Para obter informações sobre como visualizar o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que você está usando.

Para obter informações adicionais sobre AWS DMS os tipos de dados, consulteTipos de dados do AWS Database Migration Service.

Tipos de dados do MySQL	AWS DMS tipos de dados
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(65.535)
LONGBLOB	BLOB
MEDIUMBLOB	BLOB
TINYBLOB	BYTES(255)
DATA	DATA
DATETIME	DATETIME

Tipos de dados do MySQL	AWS DMS tipos de dados
	DATETIME sem um valor entre parênteses é replicado sem milissegundos. DATETIME com um valor entre parênteses de 1 a 5 (como`DATETIME(5)`) é replicado com milissegundos. Ao replicar uma coluna DATETIME, a hora permanece a mesma no destino. Não é convertida em UTC.
TIME	STRING
TIMESTAMP	DATETIME Ao replicar uma coluna TIMESTAMP, a hora é convertida em UTC no destino.
YEAR	INT2
DOUBLE	REAL8
FLOAT	REAL(DOUBLE) Se os valores FLOAT não estiverem no intervalo a seguir, utilize uma transformação para mapear FLOAT para STRING. Para obter mais informações sobre transformações, consulte Regras de transformação e ações. Os intervalos compatíveis com FLOAT são -1.79E+308 a -2.23E-308, 0 e 2.23E-308 a 1.79E+308
VARCHAR (45)	WSTRING (45)
VARCHAR (2000)	WSTRING (2000)
VARCHAR (4000)	WSTRING (4000)

Tipos de dados do MySQL	AWS DMS tipos de dados
VARBINARY (4000)	BYTES (4000)
VARBINARY (2000)	BYTES (2000)
CHAR	WSTRING
TEXT	WSTRING
LONGTEXT	NCLOB
MEDIUMTEXT	NCLOB
TINYTEXT	WSTRING(255)
GEOMETRY	BLOB
POINT	BLOB
LINESTRING	BLOB
POLYGON	BLOB
MULTIPOINT	BLOB
MULTILINESTRING	BLOB
MULTIPOLYGON	BLOB
GEOMETRYCOLLECTION	BLOB
ENUM	WSTRING (*tamanho*) Aqui, *tamanho* é o tamanho do valor mais longo em ENUM.
SET	WSTRING (*tamanho*) Aqui, *tamanho* é o tamanho total de todos os valores no SET, incluindo vírgulas.

Tipos de dados do MySQL	AWS DMS tipos de dados
JSON	CLOB

> **Note**
>
> Em alguns casos, é possível especificar os tipos de dados DATETIME e TIMESTAMP com um valor “zero” (ou seja, 0000-00-00). Nesse caso, verifique se o banco de dados de destino na tarefa de replicação é compatível com valores “zero” para os tipos de dados DATETIME e TIMESTAMP. Caso contrário, esses valores serão registrados como nulos no destino.

Utilizar um banco de dados SAP ASE como origem do AWS DMS

É possível migrar dados de um banco de dados SAP Adaptive Server Enterprise (ASE), conhecido anteriormente como Sybase, utilizando o AWS DMS. Com o banco de dados SAP ASE como origem, é possível migrar dados para qualquer um dos outros bancos de dados de destino compatíveis com o AWS DMS.

Para obter informações sobre as versões do SAP ASE compatíveis com o AWS DMS como origem, consulte Fontes para AWS DMS.

Para obter mais detalhes sobre a utilização dos bancos de dados SAP ASE e AWS DMS, consulte as seguintes seções.

Tópicos

- Pré-requisitos para a utilização de um banco de dados SAP ASE como origem do AWS DMS
- Limitações de uso do SAP ASE como origem do AWS DMS
- Permissões necessárias para utilizar o SAP ASE como origem do AWS DMS
- Remover o ponto de truncamento
- Configurações de endpoint ao utilizar o SAP ASE como origem para o AWS DMS
- Tipos de dados de origem do SAP ASE

Pré-requisitos para a utilização de um banco de dados SAP ASE como origem do AWS DMS

Para que um banco de dados SAP ASE ser uma origem do AWS DMS, faça o seguinte:

- Ative a replicação do SAP ASE para tabelas utilizando o comando `sp_setreptable`. Para obter mais informações, consulte Sybase Infocenter Archive.
- Desabilite `RepAgent` no banco de dados SAP ASE. Para obter mais informações, consulte Interromper e desativar o thread RepAgent no banco de dados primário.
- Para replicar para o SAP ASE versão 15.7 em uma instância do Windows EC2 configurado para caracteres não latinos (por exemplo, chinês), instale o SAP ASE 15.7 SP121 no computador de destino.

> **Note**
>
> Para a replicação de captura de dados de alteração (CDC), o DMS executa `dbcc logtransfer` e `dbcc log` para ler os dados do log de transações.

Limitações de uso do SAP ASE como origem do AWS DMS

As seguintes limitações se aplicam quando um banco de dados SAP ASE é utilizado como origem do AWS DMS:

- É possível executar somente uma tarefa do AWS DMS com replicação contínua ou CDC para cada banco de dados SAP ASE. É possível executar várias tarefas somente de carga máxima em paralelo.
- Não é possível renomear uma tabela. Por exemplo, o comando a seguir falha.

```
sp_rename 'Sales.SalesRegion', 'SalesReg;
```

- Não é possível renomear uma coluna. Por exemplo, o comando a seguir falha.

```
sp_rename 'Sales.Sales.Region', 'RegID', 'COLUMN';
```

- Os valores zero presentes no final de strings de tipos de dados binários são truncados quando replicados para o banco de dados de destino. Por exemplo,

`0x00000000000000000000000001000001000000000` na tabela de origem torna-se `0x00000000000000000000000001000001` na tabela de destino.

- O AWS DMS cria a tabela de destino com colunas que não permitem valores NULL, caso o padrão no banco de dados esteja definido para não permitir valores NULL. Portanto, se uma tarefa de replicação de carga máxima ou CDC contiver valores vazios, o AWS DMS lançará um erro. É possível evitar esses erros permitindo valores NULL no banco de dados de origem com os seguintes comandos.

```
sp_dboption database_name, 'allow nulls by default', 'true'
go
use database_name
CHECKPOINT
go
```

- O comando de índice `reorg rebuild` não é compatível.
- O AWS DMS não é compatível com clusters nem utiliza MSA (Multi-Site Availability)/Warm Standby como origem.
- Quando a expressão do cabeçalho de transformação `AR_H_TIMESTAMP` é utilizada em regras de mapeamento, os milissegundos não serão capturados para uma coluna adicionada.
- A execução de operações de mesclagem durante a CDC resultará em um erro não recuperável. Para sincronizar o destino novamente, execute uma carga máxima.
- Os eventos de acionador de reversão não são compatíveis com tabelas que utilizam um esquema de bloqueio de linhas de dados.
- O AWS DMS não pode retomar uma tarefa de replicação depois de eliminar uma tabela dentro do escopo da tarefa de um banco de dados SAP de origem. Se a tarefa de replicação do DMS foi interrompida e executou qualquer operação DML (INSERT, UPDATE, DELETE) seguida pelo descarte da tabela, reinicie a tarefa de replicação.

Permissões necessárias para utilizar o SAP ASE como origem do AWS DMS

Para utilizar um banco de dados SAP ASE como origem em uma tarefa do AWS DMS, é necessário conceder permissões. Conceda à conta de usuário especificada nas definições do banco de dados do AWS DMS as seguintes permissões no banco de dados SAP ASE:

- sa_role
- replication_role

- sybase_ts_role
- Por padrão, quando uma permissão é necessária para executar o procedimento armazenado `sp_setreptable`, o AWS DMS ativa a opção de replicação do SAP ASE. Se você quiser executar `sp_setreptable` em uma tabela diretamente no endpoint do banco de dados e não por meio do próprio AWS DMS, é possível utilizar o atributo de conexão adicional do `enableReplication`. Para obter mais informações, consulte Configurações de endpoint ao utilizar o SAP ASE como origem para o AWS DMS.

Remover o ponto de truncamento

Ao iniciar uma tarefa, o AWS DMS cria uma entrada `$replication_truncation_point` na exibição do sistema `syslogshold`, indicando que o processo de replicação está em andamento. Enquanto o AWS DMS está trabalhando, ele avança o ponto de truncamento de replicação em intervalos regulares, de acordo com o volume dos dados que já foram copiados para o destino.

Assim que a entrada `$replication_truncation_point` for criada, mantenha a tarefa do AWS DMS em execução para evitar que o log do banco de dados cresça excessivamente. Se quiser parar permanentemente a tarefa do AWS DMS, remova o ponto de truncamento de replicação com o seguinte comando:

```
dbcc settrunc('ltm','ignore')
```

Depois que o ponto de truncamento for removido, não será possível retomar a tarefa do AWS DMS. O log continuará a ser automaticamente truncado nos pontos de verificação (se o truncamento automático for definido).

Configurações de endpoint ao utilizar o SAP ASE como origem para o AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados de origem do SAP ASE de maneira semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de origem utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--sybase-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o SAP ASE como origem.

Nome	Descrição
`Charset`	Defina esse atributo para o nome do SAP ASE que corresponde ao conjunto de caracteres internacional. Valor padrão: `iso_1` Exemplo: `--sybase-settings '{"Charset": "utf8"}'` Valores válidos: • `acsii_8` • `big5hk` • `cp437` • `cp850` • `cp852` • `cp852` • `cp855` • `cp857` • `cp858` • `cp860` • `cp864` • `cp866` • `cp869` • `cp874` • `cp932` • `cp936` • `cp950` • `cp1250` • `cp1251` • `cp1252` • `cp1253`

Nome	Descrição
	• cp1254 • cp1255 • cp1256 • cp1257 • cp1258 • deckanji • euccns • eucgb • eucjis • eucksc • gb18030 • greek8 • iso_1 • iso88592 • iso88595 • iso88596 • iso88597 • iso88598 • iso88599 • iso15 • kz1048 • koi8 • roman8 • iso88599 • sjis • tis620 • turkish8 • utf8

Nome	Descrição
	Em caso de outras dúvidas sobre os conjuntos de caracteres compatíveis em um banco de dados SAP ASE, consulte Adaptive Server Enterprise: conjuntos de caracteres compatíveis.
`EnableReplication`	Defina esse atributo se quiser ativar `sp_setreptable` em tabelas no final do banco de dados e não por meio do AWS DMS. Valor padrão: `true` Valores válidos: `true` ou `false` Exemplo: `--sybase-settings '{"EnableReplication": false}'`
`EncryptPassword`	Defina esse atributo se você tiver ativado `"net password encryption reqd"` no banco de dados de origem. Valor padrão: `0` Valores válidos: `0`, 1 ou 2 Exemplo: `--sybase-settings '{"EncryptPassword": 1}'` Para obter mais informações sobre esses valores de parâmetros, consulte Adaptive Server Enterprise: utilizando a propriedade da string de conexão EncryptPassword.

Nome	Descrição
`Provider`	Defina esse atributo se quiser utilizar o Transport Layer Security (TLS) 1.2 para versões do ASE 15.7 e superiores. Observe que a AWS requer o TLS versão 1.2 ou posterior e recomenda a versão 1.3. Valor padrão: `Adaptive Server Enterprise` Valores válidos: `Adaptive Server Enterprise 16.03.06` Exemplo: `--sybase-settings '{"Provider": "Adaptive Server Enterprise 16.03.06"}'`

Tipos de dados de origem do SAP ASE

Para obter uma lista dos tipos de dados de origem do SAP ASE compatíveis ao utilizar o AWS DMS e o mapeamento padrão dos tipos de dados do AWS DMS, consulte a tabela a seguir. O AWS DMS não é compatível com tabelas de origem do SAP ASE com colunas do tipo de dados tipos definidos pelo usuário (UDT). Colunas replicadas com esse tipo de dados são criadas como NULL.

Para obter informações sobre como exibir o tipo de dados mapeado no destino, consulte a seção Destinos para a migração de dados relativa ao seu endpoint de destino.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipos de dados do SAP ASE	Tipos de dados do AWS DMS
BIGINT	INT8
UNSIGNED BIGINT	UINT8
INT	INT4
UNSIGNED INT	UINT4
SMALLINT	INT2

Tipos de dados do SAP ASE	Tipos de dados do AWS DMS
UNSIGNED SMALLINT	UINT2
TINYINT	UINT1
DECIMAL	NUMERIC
NUMERIC	NUMERIC
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
MONEY	NUMERIC
SMALLMONEY	NUMERIC
DATETIME	DATETIME
BIGDATETIME	DATETIME(6)
SMALLDATETIME	DATETIME
DATE	DATE
TIME	TIME
BIGTIME	TIME
CHAR	STRING
UNICHAR	WSTRING
NCHAR	WSTRING
VARCHAR	STRING
UNIVARCHAR	WSTRING

Tipos de dados do SAP ASE	Tipos de dados do AWS DMS
NVARCHAR	WSTRING
BINARY	BYTES
VARBINARY	BYTES
BIT	BOOLEAN
TEXT	CLOB
UNITEXT	NCLOB
IMAGE	BLOB

Utilizar o MongoDB como origem do AWS DMS

Para obter informações sobre as versões do MongoDB compatíveis com o AWS DMS como origem, consulte Fontes para AWS DMS.

Observe o seguinte sobre a compatibilidade com a versão do MongoDB:

- As versões 3.4.5 e posteriores do AWS DMS são compatíveis com as versões 4.2 e 4.4 do MongoDB.
- As versões 3.4.5 e posteriores do AWS DMS e as versões 4.2 e posteriores do MongoDB são compatíveis com transações distribuídas. Para obter mais informações sobre as transações distribuídas do MongoDB, consulte Transações na Documentação do MongoDB.
- As versões 3.5.0 e posteriores do AWS DMS não são compatíveis com às versões anteriores à 3.6 do MongoDB.
- As versões 3.5.1 e posteriores do AWS DMS são compatíveis com a versão 5.0 do MongoDB.
- As versões 3.5.2 e posteriores do AWS DMS são compatíveis com a versão 6.0 do MongoDB.

Se você é novo no MongoDB, esteja ciente quanto aos seguintes conceitos importantes sobre o banco de dados MongoDB.

- Um registro no MongoDB é um documento, que é uma estrutura de dados composta por pares de campo e valor. O valor de um campo pode incluir outros documentos, matrizes e matrizes de documentos. Um documento é aproximadamente equivalente a uma linha em uma tabela de banco de dados relacional.
- Uma coleção no MongoDB é um grupo de documentos e é aproximadamente equivalente a uma tabela de banco de dados relacional.
- Um banco de dados no MongoDB é um conjunto de coleções e é aproximadamente equivalente a uma tabela de banco de dados relacional.
- Internamente, um documento do MongoDB é armazenado como um arquivo binário JSON (BSON) em um formato compactado que inclui um tipo para cada campo no documento. Cada documento tem um ID exclusivo.

O AWS DMS é compatível com dois modos de migração ao utilizar o MongoDB como origem, Modo documento ou Modo tabela. Você especifica o modo de migração a ser utilizado ao criar o endpoint do MongoDB ou ao configurar o parâmetro do Modo metadadosAWS DMS no console do . Opcionalmente, é possível criar uma segunda coluna chamada `_id` que atue como a chave primária selecionando o botão de marca de seleção para _id como uma coluna separada no painel de configuração do endpoint.

A escolha do modo de migração afeta o formato resultante dos dados de destino, conforme explicado a seguir.

Modo de documentos

No modo de documentos, o documento do MongoDB é migrado "no estado em que se encontra", ou seja, os dados do documento são consolidados em uma única coluna chamada `_doc` em uma tabela de destino. O modo de documentos é a configuração padrão ao utilizar o MongoDB como um endpoint de origem.

Por exemplo, considere os seguintes documentos em uma coleção do MongoDB chamada myCollection.

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Após a migração dos dados para uma tabela de banco de dados relacional usando o modo de documentos, os dados são estruturados como mostrado a seguir. Os campos de dados no documento do MongoDB são consolidados na coluna `_doc`.

oid_id	_doc
5a94815f40bd44d1b02bdfe0	`{ "a" : 1, "b" : 2, "c" : 3 }`
5a94815f40bd44d1b02bdfe1	`{ "a" : 4, "b" : 5, "c" : 6 }`

Opcionalmente, é possível definir o atributo de conexão adicional `extractDocID` como verdadeiro para criar uma segunda coluna chamada `"_id"`, que servirá como a chave primária. Se for utilizar a CDC, defina esse parâmetro como verdadeiro.

No modo de documentos, o AWS DMS gerencia a criação e a renomeação das coleções da seguinte forma:

- Se você adicionar uma nova coleção ao banco de dados de origem, o AWS DMS criará uma nova tabela de destino para a coleção e replicara todos os documentos.
- Se você renomear uma coleção existente no banco de dados de origem, o AWS DMS não renomeará a tabela de destino.

Se o endpoint de destino for o Amazon DocumentDB, execute a migração no Modo documento.

Modo de tabelas

No modo de tabela, o AWS DMS transforma cada campo de nível superior de um documento do MongoDB em uma coluna na tabela de destino. Se um campo estiver aninhado, o AWS DMS nivelará os valores aninhados em uma única coluna. O AWS DMS então adiciona um campo de chave e os tipos de dados ao conjunto de colunas da tabela de destino.

Para cada documento do MongoDB, o AWS DMS adiciona cada chave e tipo ao conjunto de colunas da tabela de destino. Por exemplo, usando o modo de tabelas, o AWS DMS migra o exemplo anterior para a tabela a seguir.

oid_id	a	b	c
5a94815f4 0bd44d1b02bdfe0	1	2	3

5a94815f4 0bd44d1b02bdfe1	4	5	6

Os valores aninhados são simplificados em uma coluna que contém os nomes das chaves separados por pontos. A coluna é nomeado concatenação dos nomes dos campos simplificados, separados por pontos. Por exemplo, o AWS DMS migra um documento JSON com um campo de valores aninhados como `{"a" : {"b" : {"c": 1}}}` para uma coluna chamada `a.b.c`.

Para criar as colunas de destino, o AWS DMS verifica um número específico de documentos do MongoDB e cria um conjunto de todos os campos e seus tipos. O AWS DMS utiliza esse conjunto para criar as colunas da tabela de destino. Se criar ou modificar o endpoint de origem do MongoDB utilizando o console, especifique o número de documentos para verificação. O valor padrão são 1.000 documentos. Ao utilizar a AWS CLI, o atributo de conexão adicional `docsToInvestigate` pode ser utilizado.

No modo de tabelas, o AWS DMS gerencia documentos e coleções da seguinte forma:

- Ao adicionar um documento a uma coleção existente, o documento é replicado. Se houver campos que não existem no destino, esses campos não serão replicados.
- Quando você atualiza um documento, o documento atualizado é replicado. Se houver campos que não existem no destino, esses campos não serão replicados.
- A exclusão de documentos é totalmente compatível.
- A adição de uma coleção nova não resultará na criação de uma nova tabela no destino quando feita durante uma tarefa de CDC.
- Na fase de captura de dados de alteração (CDC), o AWS DMS não é compatível com a renomeação de uma coleção.

Tópicos

- Tipos de dados de origem do MongoDB

Permissões necessárias ao utilizar o MongoDB como origem do AWS DMS

Para uma migração do AWS DMS com uma origem do MongoDB, é possível criar uma conta de usuário com privilégios de raiz ou um usuário com permissões para migração somente no banco de dados.

O código a seguir cria um usuário para ser a conta raiz.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  }
)
```

Para um origem do MongoDB 3.x, o código a seguir cria um usuário com privilégios mínimos no banco de dados a ser migrado.

```
use database_to_migrate
db.createUser(
{
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "local" }, "read"]
})
```

Para um MongoDB 4.x de origem, o código a seguir cria um usuário com privilégios mínimos.

```
{ resource: { db: "", collection: "" }, actions: [ "find", "changeStream" ] }
```

Por exemplo, crie o seguinte perfil no banco de dados “admin”.

```
use admin
db.createRole(
{
role: "changestreamrole",
privileges: [
```

```
{ resource: { db: "", collection: "" }, actions: [ "find","changeStream" ] }
],
roles: []
}
)
```

Quando o perfil estiver criado, crie um usuário no banco de dados a ser migrado.

```
> use test
> db.createUser(
{
user: "dms-user12345",
pwd: "password",
roles: [ { role: "changestreamrole", db: "admin" }, "read"]
})
```

Configurar um conjunto de réplicas do MongoDB para a CDC

Para utilizar a replicação contínua ou a CDC com o MongoDB, o AWS DMS requer acesso ao log de operações do MongoDB (oplog). Para criar o oplog, é necessário implantar um conjunto de réplicas, caso não exista. Para obter mais informações, consulte a documentação do MongoDB.

É possível utilizar a CDC com o nó primário ou secundário de um conjunto de réplicas do MongoDB como o endpoint de origem.

Para converter uma instância independente em um conjunto de réplicas

1. utilizando a linha de comando, conecte-se ao `mongo`.

   ```
   mongo localhost
   ```

2. Interrompa o serviço `mongod`.

   ```
   service mongod stop
   ```

3. Reinicie o `mongod` utilizando o comando a seguir:

   ```
   mongod --replSet "rs0" --auth -port port_number
   ```

4. Teste a conexão com o conjunto de réplicas utilizando os seguintes comandos:

   ```
   mongo -u root -p password --host rs0/localhost:port_number
   ```

```
  --authenticationDatabase "admin"
```

Se planeja executar uma migração no modo de documentos, selecione a opção `_id as a separate column` ao criar o endpoint do MongoDB. Selecionar essa opção cria uma segunda coluna chamada `_id` que atua como a chave primária. Essa segunda coluna é exigida pelo AWS DMS para oferecer suporte às operações de linguagem de manipulação de dados (DML).

> **Note**
>
> O AWS DMS utiliza o log de operações (oplog) para capturar as alterações durante a replicação contínua. Se o MongoDB eliminar os registros do oplog antes de o AWS DMS lê-los, haverá falha nas tarefas. É recomendável dimensionar o oplog para reter as alterações por pelo menos 24 horas.

Requisitos de segurança ao utilizar o MongoDB como origem do AWS DMS

O AWS é compatível com dois métodos de autenticação para o MongoDB. Os dois métodos de autenticação são utilizados para criptografar a senha, e portanto só são utilizados quando o parâmetro `authType` está definido como PASSWORD.

Os métodos de autenticação do MongoDB são os seguintes:

- MONGODB-CR: para compatibilidade com versões anteriores
- SCRAM-SHA-1: o padrão ao utilizar o MongoDB versão 3.x e 4.0.

Se um método de autenticação não for especificado, o AWS DMS utilizará o método padrão da versão do MongoDB de origem.

Segmentar as coleções do MongoDB e migrar em paralelo

Para melhorar o desempenho de uma tarefa de migração, os endpoints de origem do MongoDB são compatíveis com duas opções para carga máxima paralela no mapeamento de tabela.

Em outras palavras, é possível migrar uma coleção em paralelo utilizando segmentação automática ou segmentação por intervalo com mapeamento de tabela para uma carga máxima paralela nas configurações do JSON. Com a segmentação automática, é possível especificar os critérios do AWS DMS para segmentar automaticamente a origem para migração em cada thread.

Com a segmentação de intervalo, é possível informar ao AWS DMS o intervalo específico de cada segmento para o DMS migrar em cada thread. Para obter mais informações sobre essas configurações, consulte Regras e operações de configurações de tabelas e coleções.

Migrar um banco de dados MongoDB em paralelo utilizando intervalos de segmentação automática

É possível migrar os documentos em paralelo especificando os critérios do AWS DMS para particionar (segmentar) automaticamente os dados de cada thread. Especificamente, informe o número de documentos a serem migrados por thread. Ao utilizar essa abordagem, o AWS DMS tenta otimizar os limites do segmento para obter o máximo desempenho por thread.

É possível especificar os critérios de segmentação utilizando as opções de configurações de tabela a seguir no mapeamento de tabela.

Opção de configurações de tabela	Descrição
`"type"`	(Obrigatório) Defina como `"partitions-auto"` para MongoDB como origem.
`"number-of-partitions"`	(Opcional) Número total de partições (segmentos) utilizadas para a migração. O padrão é 16.
`"collection-count-from-metadata"`	(Opcional) Se essa opção estiver definida como `true`, o AWS DMS utilizará uma contagem estimada da coleção para determinar o número de partições. Se essa opção estiver definida como`false`, o AWS DMS utilizará a contagem real da coleção. O padrão é `true`.
`"max-records-skip-per-page"`	(Opcional) O número de registros a serem ignorados imediatamente ao determinar os limites de cada partição. O AWS DMS utiliza uma abordagem de ignorar páginas ao determinar o limite mínimo de uma partição. O padrão é 10.000. A definição de um valor relativamente grande pode resultar em tempos limite do cursor e falhas na tarefa. A definição de um valor

Opção de configurações de tabela	Descrição
	relativamente baixo resulta em mais operações por página e em uma carga máxima mais lenta.
`"batch-size"`	(Opcional) Limita o número de documentos retornados em um lote. Cada lote requer uma viagem de ida e volta ao servidor. Se o tamanho do lote for zero (0), o cursor utilizará o tamanho máximo do lote definido pelo servidor. O padrão é 0.

O exemplo a seguir mostra um mapeamento de tabela para segmentação automática.

```
{
    "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
```

```
            }
        }
    ]
}
```

A segmentação automática tem a seguinte limitação. A migração de cada segmento busca a contagem da coleção e o `_id` mínimo da coleção separadamente. Ela utiliza um salto paginado para calcular o limite mínimo desse segmento.

Portanto, verifique se o valor mínimo de `_id` de cada coleção permanece constante até que todos os limites do segmento na coleção tenham sido calculados. Se você alterar o valor mínimo de `_id` de uma coleção durante o cálculo do limite do segmento, isso poderá causar perda de dados ou erros de linha duplicada.

Migrar de um banco de dados MongoDB em paralelo utilizando segmentação de intervalo

É possível migrar os documentos em paralelo especificando os intervalos de cada segmento em um thread. Ao utilizar essa abordagem, você informa ao AWS DMS os documentos específicos a serem migrados em cada thread de acordo com a sua escolha de intervalos de documentos por thread.

O exemplo a seguir mostra uma coleção do MongoDB que tem sete itens e `_id` como chave primária.

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

Para dividir a coleção em três segmentos específicos para o AWS DMS migrar em paralelo, é possível adicionar regras de mapeamento de tabela à tarefa de migração. Essa abordagem é mostrada no exemplo de JSON a seguir.

```
{ // 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
```

Essa definição de mapeamento de tabela divide a coleção de origem em três segmentos e migra em paralelo. Veja a seguir os limites de segmentação.

```
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)
```

Depois que a tarefa de migração for concluída, é possível verificar os logs de tarefas para saber se as tabelas foram carregadas em paralelo, conforme mostrado no exemplo a seguir. Também é possível verificar a cláusula `find` do MongoDB utilizada para descarregar cada segmento da tabela de origem.

```
[TASK_MANAGER    ] I:  Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86  (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is: { "_id" :
 { "$lte" : { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" :
 { "$numberInt" : "2" } } }  (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.

[TASK_MANAGER    ] I: Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86 (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is: { "_id" : { "$lte" :
 { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" : { "$numberInt" :
 "2" } } } (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.
```

```
[TARGET_LOAD     ] I: Load finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 1 rows received. 0 rows skipped. Volume
 transfered 480.

[TASK_MANAGER    ] I: Load finished for segment #1 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. 2 records transferred.
```

Atualmente, o AWS DMS é compatível com os seguintes tipos de dados do MongoDB como uma coluna de chave de segmento:

- Double
- String
- ObjectId
- Inteiro de 32 bits
- Inteiro de 64 bits

Migrar vários bancos de dados ao utilizar o MongoDB como origem do AWS DMS

O AWS DMS versões 3.4.5 e superiores são compatíveis com a migração de vários bancos de dados em uma única tarefa para todas as versões do MongoDB compatíveis. Para migrar vários bancos de dados, utilize as seguintes etapas:

1. Ao criar o endpoint de origem do MongoDB, siga um destes procedimentos:
   - Na página Criar endpoint do console do DMS, verifique se o Nome do banco de dados está vazio em Configuração do endpoint.
   - Utilizando o comando `CreateEndpoint` da AWS CLI, atribua um valor de string vazia ao parâmetro `DatabaseName` em `MongoDBSettings`.
2. Para cada banco de dados a ser migrado de uma origem do MongoDB, especifique o nome do banco de dados como um nome de esquema no mapeamento da tabela da tarefa. É possível fazer isso utilizando a entrada guiada no console ou diretamente no JSON. Para obter mais informações sobre a entrada guiada, consulte Especificar a seleção de tabelas e as regras de transformação no console. Para obter mais informações sobre o JSON, consulte Regras de seleção e ações.

Por exemplo, é possível especificar o JSON a seguir para migrar três bancos de dados do MongoDB.

Example Migrar todas as tabelas em um esquema

O JSON a seguir migra todas as tabelas dos bancos de dados `Customers`, `Orders` e `Suppliers` no endpoint de origem para o endpoint de destino.

```
{
    "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": []
        }
    ]
}
```

Limitações de uso do MongoDB como origem do AWS DMS

Veja a seguir as limitações ao utilizar o MongoDB como origem do AWS DMS:

- No modo tabela, os documentos em uma coleção devem ser consistentes com o tipo de dados que utilizam para o valor no mesmo campo. Por exemplo, se um documento em uma coleção incluir `'{ a:{ b:value ... }'`, todos os documentos da coleção que fazem referência ao *`value`* do campo `a.b` devem utilizar o mesmo tipo de dados para *`value`*, onde quer que apareçam na coleção.
- Quando a opção `_id` está definida como uma coluna separada, a string de ID não pode exceder 200 caracteres.
- As chaves de ID de objetos e de tipos de array são convertidas em colunas com o prefixo `oid` e `array` no modo de tabela.

  Internamente, essas colunas são referenciadas com os nomes prefixados. Se utilizar regras de transformação no AWS DMS que fazem referência a essas colunas, especifique a coluna prefixada. Por exemplo, especifique `${oid__id}` e não `${_id}`, ou `${array__addresses}` e não `${_addresses}`.
- Os nomes de coleções e os nomes de chaves não podem conter o caractere de cifrão ($).
- O AWS DMS não é compatível com coleções que contêm o mesmo campo com maiúsculas e minúsculas diferentes no modo tabela com um destino RDBMS. Por exemplo, o AWS DMS não é compatível com duas coleções chamadas `Field1` `efield1`.
- Os modos tabelas e documento possuem as limitações descritas anteriormente.
- A migração em paralelo que utiliza a segmentação automática possui as limitações descritas anteriormente.
- Os filtros de origem não são compatíveis com o MongoDB.
- O AWS DMS não é compatível com documentos em que o nível de aninhamento é maior que 97.
- O AWS DMS não é compatível com os seguintes recursos do MongoDB versão 5.0:
  - Refragmentação em tempo real
  - Criptografia em nível de campo do lado do cliente (CSFLE)
  - Migração de coleção de séries temporais

> **Note**
>
> Uma coleção de séries temporais migrada na fase de carga máxima será convertida em uma coleção normal no Amazon DocumentDB, porque o DocumentDB não é compatível com coleções de séries temporais.

Definições de configuração do endpoint ao utilizar o MongoDB como origem do AWS DMS

Ao configurar o endpoint de origem do MongoDB, é possível especificar várias configurações de endpoint utilizando o console do AWS DMS.

A tabela a seguir descreve as propriedades de configuração disponíveis ao utilizar um banco de dados MongoDB como origem do AWS DMS.

Configuração (atributo)	Valores válidos	Valor padrão e descrição
Modo de autenticação	`"none"` `"password"`	O valor `"password"` solicita um nome de usuário e uma senha. Quando `"none"` for especificado, os parâmetros de nome de usuário e senha não serão utilizados.
Origem da autenticação	Um nome de banco de dados MongoDB válido.	O nome do banco de dados MongoDB que você deseja utilizar para validar as credenciais para autenticação. O valor padrão é `"admin"`.

Mecanismo de autenticação	`"default"`
	`"mongodb_cr"`
	`"scram_sha_1"`
		O mecanismo de autenticação. O valor `"default"` é `"scram_sha_1"` . Essa configuração não é utilizada quando `authType` está definido como `"no"`.

Modo metadados	Documento e tabela	Escolhe o modo documento ou o modo tabela.

Configuração (atributo)	Valores válidos	Valor padrão e descrição
Número de documentos a serem verificados (`docsToInvestigate`)	Um inteiro positivo maior do que `0`.	Utilize essa opção no modo tabela somente para definir a configuração da tabela de destino.
_id como uma coluna separada	Marca de seleção na caixa	Marca de seleção opcional que cria uma segunda coluna chamada `_id` que atua como a chave primária.
`socketTimeoutMS`	NUMBER Somente atributo de conexão adicional (ECA).	Essa configuração está em unidades de milissegundos e configura o tempo limite de conexão para clientes MongoDB. Se o valor for menor ou igual a zero, o padrão do cliente MongoDB será utilizado.

`UseUpdateLookUp`	boolean
	
	`true`
	
	`false`
		Quando verdadeiro, durante os eventos de atualização da CDC, o AWS DMS copia todo o documento atualizado no destino. Quando definido como falso, o AWS DMS utiliza o comando de atualização do MongoDB para atualizar somente os campos modificados no documento no destino.

`ReplicateShardCollections`	boolean
	
	`true`
	
	`false`
		Quando verdadeiro, o AWS DMS replica os dados em coleções de fragmentos. OAWS DMS só utiliza essa configuração se o endpoint de destino for um cluster elástico do DocumentDB.
		
		Quando essa configuração for verdadeira, observe o seguinte:
		
		• Defina `TargetTablePrepMode` como `nothing`.
		• O AWS DMS define automaticamente `useUpdateLookup` como `false`.

Se você escolher o Documento como Modo metadados, opções diferentes estarão disponíveis.

Se o endpoint de destino for DocumentDB, execute a migração no Modo documento. Além disso, modifique o endpoint de origem e selecione a opção _id como coluna separada. Esse será um pré-requisito obrigatório se a workload de origem do MongoDB envolver transações.

Tipos de dados de origem do MongoDB

A migração de dados que utiliza o MongoDB como origem do AWS DMS é compatível com a maioria dos tipos de dados do MongoDB. Na tabela a seguir, encontre os tipos de dados de origem do MongoDB compatíveis ao utilizar o AWS DMS e o mapeamento padrão de tipos de dados do AWS DMS. Para obter mais informações sobre os tipos de dados do MongoDB, consulte Tipos de BSON na documentação do MongoDB.

Para obter informações sobre como exibir o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que está usando.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipos de dados do MongoDB	Tipos de dados do AWS DMS
Booleano	Bool
Binário	BLOB
Data	Data
Marca de data e hora	Data
Int	INT4
Longo	INT8
Double	REAL8
String (UTF-8)	CLOB
Array	CLOB
OID	String

Tipos de dados do MongoDB	Tipos de dados do AWS DMS
REGEX	CLOB
Código	CLOB

Usando o Amazon DocumentDB (com compatibilidade com o MongoDB) como fonte para AWS DMS

Para obter informações sobre as versões do Amazon DocumentDB (compatível com MongoDB) que são compatíveis com o AWS DMS como origem, consulte Fontes para AWS DMS.

Utilizando o Amazon DocumentDB como origem, é possível migrar dados de um cluster do Amazon DocumentDB para outro cluster do Amazon DocumentDB. Você também pode migrar dados de um cluster do Amazon DocumentDB para um dos outros endpoints de destino suportados pelo. AWS DMS

Se não estiver familiarizado com o Amazon DocumentDB, fique ciente dos seguintes conceitos importantes de bancos de dados do Amazon DocumentDB:

- Um registro no Amazon DocumentDB é um documento, uma estrutura de dados composta por pares de campo e valor. O valor de um campo pode incluir outros documentos, matrizes e matrizes de documentos. Um documento é aproximadamente equivalente a uma linha em uma tabela de banco de dados relacional.
- Uma coleção no Amazon DocumentDB é um grupo de documentos, e é aproximadamente equivalente a uma tabela de banco de dados relacional.
- Um banco de dados no Amazon DocumentDB é um conjunto de coleções, e é aproximadamente equivalente a um esquema em um banco de dados relacional.

AWS DMS suporta dois modos de migração ao usar o Amazon DocumentDB como fonte, modo de documento e modo de tabela. Você especifica o modo de migração ao criar o endpoint de origem do Amazon DocumentDB no AWS DMS console, usando a opção do modo de metadados ou o atributo de conexão extra. `nestingLevel` Veja a seguir uma explicação de como a opção do modo de migração afeta o formato resultante dos dados de destino.

Modo de documentos

No Modo documento, o documento JSON é migrado como está. Isso significa que os dados do documento são consolidados em um de dois itens. Ao utilizar um banco de dados relacional como destino, os dados são uma única coluna nomeada `_doc` em uma tabela de destino. Ao utilizar um banco de dados não relacional como destino, os dados são um único documento JSON. O modo documento é o modo padrão, que é recomendável ao migrar para um destino do Amazon DocumentDB.

Por exemplo, considere os seguintes documentos em uma coleção do Amazon DocumentDB chamada `myCollection`.

```
> db.myCollection.find()
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe0"), "a" : 1, "b" : 2, "c" : 3 }
{ "_id" : ObjectId("5a94815f40bd44d1b02bdfe1"), "a" : 4, "b" : 5, "c" : 6 }
```

Após a migração dos dados para uma tabela de banco de dados relacional usando o modo de documentos, os dados são estruturados como mostrado a seguir. Os campos de dados no documento são consolidados na coluna `_doc`.

oid_id	_doc
5a94815f40bd44d1b02bdfe0	`{ "a" : 1, "b" : 2, "c" : 3 }`
5a94815f40bd44d1b02bdfe1	`{ "a" : 4, "b" : 5, "c" : 6 }`

Opcionalmente, é possível definir o atributo de conexão adicional `extractDocID` como `true` para criar uma segunda coluna chamada `"_id"`, que servirá como a chave primária. Se você for utilizar a captura de dados de alteração (CDC), defina esse parâmetro como `true`, exceto ao utilizar o Amazon DocumentDB como destino.

> **Note**
>
> Se você adicionar uma nova coleção ao banco de dados de origem, AWS DMS cria uma nova tabela de destino para a coleção e replica todos os documentos.

Modo de tabelas

No Modo tabela, o AWS DMS transforma cada campo de nível superior de um documento do Amazon DocumentDB em uma coluna na tabela de destino. Se um campo estiver aninhado, AWS DMS nivela os valores aninhados em uma única coluna. AWS DMS em seguida, adiciona um campo-chave e tipos de dados ao conjunto de colunas da tabela de destino.

Para cada documento do Amazon DocumentDB, AWS DMS adiciona cada chave e tipo ao conjunto de colunas da tabela de destino. Por exemplo, usando o modo tabela, AWS DMS migra o exemplo anterior para a tabela a seguir.

oid_id	a	b	c
5a94815f4 0bd44d1b02bdfe0	1	2	3
5a94815f4 0bd44d1b02bdfe1	4	5	6

Os valores aninhados são simplificados em uma coluna que contém os nomes das chaves separados por pontos. A coluna é nomeada utilizando concatenação dos nomes dos campos simplificados, separados por pontos. Por exemplo, AWS DMS migra um documento JSON com um campo de valores aninhados, como `{"a" : {"b" : {"c": 1}}}` em uma coluna chamada `a.b.c.`

Para criar as colunas de destino, AWS DMS digitaliza um número específico de documentos do Amazon DocumentDB e cria um conjunto de todos os campos e seus tipos. AWS DMS em seguida, usa esse conjunto para criar as colunas da tabela de destino. Ao criar ou modificar o endpoint de origem do Amazon DocumentDB utilizando o console, especifique o número de documentos para verificação. O valor padrão são 1.000 documentos. Se você usar o AWS CLI, poderá usar o atributo de conexão extra`docsToInvestigate`.

No modo tabela, AWS DMS gerencia documentos e coleções da seguinte forma:

- Ao adicionar um documento a uma coleção existente, o documento é replicado. Se houver campos que não existem no destino, esses campos não serão replicados.
- Quando você atualiza um documento, o documento atualizado é replicado. Se houver campos que não existem no destino, esses campos não serão replicados.

- A exclusão de documentos é totalmente compatível.
- A adição de uma coleção nova não resultará na criação de uma nova tabela no destino quando feita durante uma tarefa de CDC.
- Na fase Change Data Capture (CDC), AWS DMS não oferece suporte à renomeação de uma coleção.

Tópicos

Definir permissões para utilizar o Amazon DocumentDB como origem

Ao usar a fonte do Amazon DocumentDB para uma AWS DMS migração, você pode criar uma conta de usuário com privilégios de root. Ou é possível criar um usuário com permissões somente para o banco de dados a ser migrado.

O código a seguir cria um usuário para ser a conta raiz.

```
use admin
db.createUser(
  {
    user: "root",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  })
```

Para o Amazon DocumentDB 3.6, o código a seguir cria um usuário com privilégios mínimos no banco de dados a ser migrado.

```
use database_to_migrate
db.createUser(
{
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "read", db: "db_name" }, "read"]
})
```

Para o Amazon DocumentDB 4.0 e superior, AWS DMS usa um fluxo de alterações em toda a implantação. Aqui, o código a seguir cria um usuário com privilégios mínimos.

```
db.createUser(
{
    user: "dms-user",
    pwd: "password",
    roles: [ { role: "readAnyDatabase", db: "admin" }]
})
```

Configurar a CDC para um cluster do Amazon DocumentDB

Para usar a replicação contínua ou o CDC com o Amazon DocumentDB AWS DMS , é necessário acesso aos fluxos de alterações do cluster Amazon DocumentDB. Para obter uma descrição da sequência ordenada por tempo dos eventos de atualização nas coleções e bancos de dados do cluster, consulte Como utilizar fluxos de alterações no Guia do desenvolvedor do Amazon DocumentDB.

Autentique-se no cluster do Amazon DocumentDB utilizando o shell do MongoDB. Execute o comando a seguir para ativar os fluxos de alterações.

```
db.adminCommand({modifyChangeStreams: 1,
    database: "DB_NAME",
    collection: "",
    enable: true});
```

Essa abordagem ativa o fluxo de alterações para todas as coleções no banco de dados. Depois que os fluxos de alterações forem habilitados, você poderá criar uma tarefa de migração que migre os dados existentes e, ao mesmo tempo, replique as alterações em andamento. AWS DMS continua

capturando e aplicando alterações mesmo após o carregamento dos dados em massa. Por fim, os bancos de dados de origem e de destino ficarão sincronizados, minimizando o tempo de inatividade de uma migração.

> **Note**
>
> AWS DMS usa o log de operações (oplog) para capturar as alterações durante a replicação contínua. Se o Amazon DocumentDB eliminar os registros do oplog antes de AWS DMS lê-los, suas tarefas falharão. É recomendável dimensionar o oplog para reter as alterações por pelo menos 24 horas.

Conectar-se ao Amazon DocumentDB utilizando TLS

Por padrão, um cluster recém-criado do Amazon DocumentDB aceita conexões seguras somente quando o Transport Layer Security (TLS) é utilizado. Quando o TLS está ativado, cada conexão ao Amazon DocumentDB requer uma chave pública.

Você pode recuperar a chave pública para o Amazon DocumentDB baixando o `rds-combined-ca-bundle.pem` arquivo de AWS um bucket do Amazon S3 hospedado. Para obter mais informações sobre como baixar esse arquivo, consulte Criptografar conexões utilizando TLS no Guia do desenvolvedor do Amazon DocumentDB.

Depois de baixar o `rds-combined-ca-bundle.pem` arquivo, você pode importar a chave pública que ele contém AWS DMS. As etapas a seguir descrevem como fazer isso.

Para importar sua chave pública usando o AWS DMS console

1. Faça login no AWS Management Console e escolha AWS DMS.
2. No painel de navegação, escolha Certificates.
3. Escolha Importar certificado. A página Importar novo certificado CA é exibida.
4. Na seção de Configuração de certificado, execute uma das seguintes ações:
   - Para Identificador do certificado, insira um nome exclusivo para o certificado, por exemplo `docdb-cert`.
   - Selecione Escolher arquivo, navegue até o local onde salvou o arquivo `rds-combined-ca-bundle.pem` e selecione-o.
5. Escolha Adicionar novo certificado CA.

O exemplo a AWS CLI seguir usa o AWS DMS `import-certificate` comando para importar o `rds-combined-ca-bundle.pem` arquivo de chave pública.

```
aws dms import-certificate \
    --certificate-identifier docdb-cert \
    --certificate-pem file://./rds-combined-ca-bundle.pem
```

Criar um endpoint de origem do Amazon DocumentDB

É possível criar um endpoint de origem do Amazon DocumentDB utilizando o console ou a AWS CLI. Utilize o procedimento a seguir com o console.

Para configurar um endpoint de origem do Amazon DocumentDB usando o console AWS DMS

1. Faça login no AWS Management Console e escolha AWS DMS.
2. No painel de navegação, escolha Endpoints e Criar endpoint.
3. Em Identificador do endpoint, forneça um nome que ajude a identificá-lo facilmente, como `docdb-source`.
4. Em Mecanismo de origem, escolha Amazon DocumentDB (compatível com MongoDB).
5. Em Nome do servidor, insira o nome do servidor em que o endpoint do banco de dados Amazon DocumentDB reside. Por exemplo, é possível inserir o nome DNS público da instância do Amazon EC2, como `democluster.cluster-cjf6q8nxfefi.us-east-2.docdb.amazonaws.com`.
6. Em Porta, insira 27017.
7. Para SSL mode (Modo SSL), escolha verificar-full. Se tiver desativado o SSL no cluster do Amazon DocumentDB, ignore essa etapa.
8. Em Certificado CA, escolha o certificado do Amazon DocumentDB, `rds-combined-ca-bundle.pem`. Para obter instruções sobre como adicionar esse certificado, consulte Conectar-se ao Amazon DocumentDB utilizando TLS.
9. Em Nome do banco de dados, insira o nome do banco de dados a ser migrado.

Utilize o procedimento a seguir com a CLI.

Para configurar um endpoint de origem do Amazon DocumentDB usando o AWS CLI

- Execute o AWS DMS `create-endpoint` comando a seguir para configurar um endpoint de origem do Amazon DocumentDB, substituindo os espaços reservados por seus próprios valores.

```
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
```

Segmentar as coleções do Amazon DocumentDB e migrar em paralelo

Para melhorar o desempenho de uma tarefa de migração, os endpoints de origem do Amazon DocumentDB são compatíveis com duas opções do recurso de carga máxima paralela no mapeamento de tabela. Em outras palavras, é possível migrar uma coleção em paralelo utilizando as opções de segmentação automática ou de segmentação de intervalo do mapeamento de tabela para uma carga máxima paralela nas configurações de JSON. As opções de segmentação automática permitem que você especifique os critérios AWS DMS para segmentar automaticamente sua fonte para migração em cada thread. As opções de segmentação de intervalo permitem que você informe AWS DMS o intervalo específico de cada segmento para o DMS migrar em cada thread. Para obter mais informações sobre essas configurações, consulte Regras e operações de configurações de tabelas e coleções.

Migrar um banco de dados Amazon DocumentDB em paralelo utilizando intervalos de segmentação automática

Você pode migrar seus documentos em paralelo especificando os critérios AWS DMS para particionar (segmentar) automaticamente seus dados para cada thread, especialmente o número de documentos a serem migrados por thread. Ao utilizar essa abordagem, o AWS DMS tenta otimizar os limites do segmento para obter o máximo desempenho por thread.

É possível especificar os critérios de segmentação utilizando as opções de configurações de tabela a seguir no mapeamento de tabela:

Opção de configurações de tabela	Descrição
`"type"`	(Obrigatório) Defina o Amazon DocumentDB. como a origem do `"partitions-auto"` .
`"number-of-partitions"`	(Opcional) Número total de partições (segmento s) utilizadas para a migração. O padrão é 16.
`"collection-count-from-metadata"`	(Opcional) Se definido como`true`, AWS DMS usa uma contagem estimada de coleta para determinar o número de partições. Se definido como`false`, AWS DMS usa a contagem real da coleta. O padrão é `true`.
`"max-records-skip-per-page"`	(Opcional) O número de registros a serem ignorados de uma vez ao determinar os limites de cada partição. AWS DMS usa uma abordagem de salto paginado para determina r o limite mínimo de uma partição. O padrão é 10000. A definição de um valor relativamente grande pode resultar em tempos limite do cursor e falhas na tarefa. A definição de um valor relativamente baixo resulta em mais operações por página e em uma carga máxima mais lenta.
`"batch-size"`	(Opcional) Limita o número de documentos retornados em um lote. Cada lote requer uma viagem de ida e volta ao servidor. Se o tamanho do lote for zero (0), o cursor utilizará o tamanho máximo do lote definido pelo servidor. O padrão é 0.

O exemplo a seguir mostra um mapeamento de tabela para segmentação automática.

```
{
    "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
            }
        }
    ]
}
```

A segmentação automática tem a seguinte limitação. A migração de cada segmento busca a contagem da coleção e o `_id` mínimo da coleção separadamente. Ela utiliza um salto paginado para calcular o limite mínimo desse segmento. Portanto, verifique se o valor mínimo de `_id` de cada coleção permanece constante até que todos os limites do segmento na coleção tenham sido calculados. Se você alterar o valor de `_id` mínimo de uma coleção durante o cálculo do limite do segmento, isso poderá causar perda de dados ou erros de linha duplicada.

Migrar um banco de dados Amazon DocumentDB em paralelo utilizando intervalos de segmentos específicos

O exemplo a seguir mostra uma coleção do Amazon DocumentDB que tem sete itens e `_id` como chave primária.

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

Para dividir a coleção em três segmentos e migrar paralelamente, é possível adicionar regras de mapeamento de tabela à tarefa de migração, conforme mostrado no exemplo de JSON a seguir.

```
{ // 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
```

Essa definição de mapeamento de tabela divide a coleção de origem em três segmentos e migra em paralelo. Veja a seguir os limites de segmentação.

```
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)
```

Depois que a tarefa de migração for concluída, é possível verificar os logs de tarefas para saber se as tabelas foram carregadas em paralelo, conforme mostrado no exemplo a seguir. Também é possível verificar a cláusula `find` do Amazon DocumentDB utilizada para descarregar cada segmento da tabela de origem.

```
[TASK_MANAGER    ] I:  Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86  (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I:  Range Segmentation filter for Segment #0 is: { "_id" :
 { "$lte" : { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" :
 { "$numberInt" : "2" } } }  (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.

[TASK_MANAGER    ] I: Start loading segment #1 of 3 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. Start load timestamp
 0005B191D638FE86 (replicationtask_util.c:752)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is initialized.
 (mongodb_unload.c:157)

[SOURCE_UNLOAD   ] I: Range Segmentation filter for Segment #0 is: { "_id" : { "$lte" :
 { "$oid" : "5f805c97873173399a278d79" } }, "num" : { "$lte" : { "$numberInt" :
 "2" } } } (mongodb_unload.c:328)

[SOURCE_UNLOAD   ] I: Unload finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 2 rows sent.
```

```
[TARGET_LOAD     ] I: Load finished for segment #1 of segmented table
 'testdatabase'.'testtable' (Id = 1). 1 rows received. 0 rows skipped. Volume
 transfered 480.

[TASK_MANAGER    ] I: Load finished for segment #1 of table
 'testdatabase'.'testtable' (Id = 1) by subtask 1. 2 records transferred.
```

Atualmente, AWS DMS oferece suporte aos seguintes tipos de dados do Amazon DocumentDB como uma coluna de chave de segmento:

- Double
- String
- ObjectId
- Inteiro de 32 bits
- Inteiro de 64 bits

Migração de vários bancos de dados ao usar o Amazon DocumentDB como fonte para AWS DMS

AWS DMS as versões 3.4.5 e superiores oferecem suporte à migração de vários bancos de dados em uma única tarefa somente para as versões 4.0 e superiores do Amazon DocumentDB. Para migrar vários bancos de dados, faça o seguinte:

1. Ao criar o endpoint de origem do Amazon DocumentDB:
   - No formulário AWS DMS, AWS Management Console deixe o nome do banco de dados vazio em Configuração do endpoint na página Criar endpoint.
   - No AWS Command Line Interface (AWS CLI), atribua um valor de string vazio ao DatabaseNameparâmetro em DocumentDBSettings que você especifica para a CreateEndpointação.
2. Para cada banco de dados a ser migrado que você quer migrar desse endpoint de origem do Amazon DocumentDB, especifique o nome de cada banco de dados como o nome de um esquema no mapeamento de tabela da tarefa utilizando a entrada guiada no console ou diretamente no JSON. Para obter mais informações sobre a entrada guiada, consulte a descrição de Especificar a seleção de tabelas e as regras de transformação no console. Para obter mais informações sobre o JSON, consulte Regras de seleção e ações.

Por exemplo, é possível especificar o JSON a seguir para migrar três bancos de dados do Amazon DocumentDB.

Example Migrar todas as tabelas em um esquema

O JSON a seguir migra todas as tabelas dos bancos de dados `Customers`, `Orders` e `Suppliers` no endpoint de origem para o endpoint de destino.

```
{
    "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"
        }
    ]
}
```

Limitações ao usar o Amazon DocumentDB como fonte para AWS DMS

A seguir estão as limitações ao usar o Amazon DocumentDB como fonte para: AWS DMS

- Quando a opção `_id` está definida como uma coluna separada, a string de ID não pode exceder 200 caracteres.
- As chaves de ID de objetos e de tipos de array são convertidas em colunas com o prefixo `oid` e `array` no modo de tabela.

  Internamente, essas colunas são referenciadas com os nomes prefixados. Se você usar regras de transformação para referenciar essas colunas, certifique-se de especificar a coluna prefixada.

AWS DMS Por exemplo, especifique `${oid__id}` e não `${_id}`, ou `${array__addresses}` e não `${_addresses}`.

- Os nomes de coleções e os nomes de chaves não podem conter o caractere de cifrão ($).
- Os modos de tabelas e documentos possuem as limitações discutidas anteriormente.
- A migração em paralelo que utiliza a segmentação automática possui as limitações descritas anteriormente.
- Uma origem do Amazon DocumentDB (compatível com MongoDB) não é compatível com a utilização de um timestamp específico como uma posição inicial para a captura de dados de alteração (CDC). Uma tarefa de replicação contínua começa a capturar as alterações, independentemente do timestamp.
- Ao utilizar o DocumentDB (compatível com MongoDB) como origem, o DMS pode tratar no máximo 250 registros por segundo.
- AWS DMS não suporta documentos em que o nível de aninhamento seja maior que 97.
- Filtros de origem não são compatíveis com o DocumentDB.
- AWS DMS não oferece suporte à replicação CDC (captura de dados de alteração) para DocumentDB como fonte no modo de cluster elástico.

Utilizar configurações de endpoint com o Amazon DocumentDB como origem

É possível utilizar as configurações de endpoint para configurar o banco de dados Amazon DocumentDB como destino de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de origem usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--doc-db-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint que podem ser utilizadas com o Amazon DocumentDB como origem.

Nome do atributo	Valores válidos	Valor padrão e descrição

`NestingLevel`	`"none"`
	
	`"one"`
		`"none"`: especifique `"none"` para utilizar o modo de documento. Especifique `"one"` para utilizar o modo de tabela.

Nome do atributo	Valores válidos	Valor padrão e descrição

`ExtractDocID`	boolean
	
	`true`
	
	`false`
		`false`: utilize este atributo quando `NestingLevel` estiver definido como `"none"`.
		
		Se o banco de dados de destino for o Amazon DocumentDB, defina `'{"ExtractDocID": true}'` .

`DocsToInvestigate`	Um inteiro positivo maior do que `0`.	`1000`: utilize este atributo quando `NestingLevel` estiver definido como `"one"`.

`ReplicateShardCollections`	boolean
	
	`true`
	
	`false`
		Quando verdadeiro, AWS DMS replica os dados em coleções de fragmentos. AWS DMS só usa essa configuração se o endpoint de destino for um cluster elástico DocumentDB.
		
		Quando essa configuração for verdadeira, observe o seguinte:
		
		• Defina `TargetTablePrepMode` como `nothing`.
		• AWS DMS define automaticamente `useUpdateLookup` como`false`.

Tipos de dados de origem do Amazon DocumentDB

Na tabela a seguir, é possível encontrar os tipos de dados de origem do Amazon DocumentDB que são compatíveis ao utilizar o AWS DMS. Você também pode encontrar o mapeamento padrão dos tipos de AWS DMS dados nesta tabela. Para obter mais informações sobre os tipos de dados, consulte Tipos BSON na documentação do MongoDB.

Para obter informações sobre como exibir o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que está usando.

Para obter informações adicionais sobre AWS DMS os tipos de dados, consulteTipos de dados do AWS Database Migration Service.

Tipos de dados do Amazon DocumentDB	AWS DMS tipos de dados
Booleano	Bool
Binário	BLOB
Data	Data
Marca de data e hora	Data
Int	INT4
Longo	INT8
Double	REAL8
String (UTF-8)	CLOB
Array	CLOB
OID	String

Usando o Amazon S3 como fonte para AWS DMS

Você pode migrar dados de um bucket do Amazon S3 usando. AWS DMS Para fazer isso, conceda acesso a um bucket do Amazon S3 que contenha um ou mais arquivos de dados. Neste bucket do S3, inclua um arquivo JSON que descreve o mapeamento entre os dados e as tabelas de banco de dados referentes aos dados nesses arquivos.

Os arquivos de dados de origem devem estar presentes no bucket do Amazon S3 para que a carga máxima seja iniciada. Especifique o nome do bucket utilizando o parâmetro `bucketName`.

Os arquivos de dados de origem podem estar nos seguintes formatos:

- Valor separado por vírgula (.csv)
- Parquet (DMS versão 3.5.3 e posterior). Para obter informações sobre como usar arquivos no formato Parquet, consulte. Usando arquivos no formato Parquet no Amazon S3 como fonte para AWS DMS

Para arquivos de dados de origem no formato de valores separados por vírgula (.csv), nomeie-os usando a seguinte convenção de nomenclatura. Nessa convenção, *schemaName* é o esquema de origem, e *tableName* é o nome de uma tabela dentro desse esquema.

```
/schemaName/tableName/LOAD001.csv
/schemaName/tableName/LOAD002.csv
/schemaName/tableName/LOAD003.csv
...
```

Por exemplo, suponha que os arquivos de dados estejam no `mybucket`, no seguinte caminho do Amazon S3.

```
s3://mybucket/hr/employee
```

No momento do carregamento, AWS DMS presume que o nome do esquema de origem é `hr` e que o nome da tabela de origem é. `employee`

Além disso `bucketName` (o que é obrigatório), você pode, opcionalmente, fornecer um `bucketFolder` parâmetro para especificar onde AWS DMS procurar arquivos de dados no bucket do Amazon S3. Continuando com o exemplo anterior, se você `bucketFolder` definir como`sourcedata`, AWS DMS lê os arquivos de dados no caminho a seguir.

```
s3://mybucket/sourcedata/hr/employee
```

É possível especificar o delimitador de coluna, o delimitador de linha, o indicador de valor nulo e outros parâmetros utilizando os atributos de conexão adicionais. Para ter mais informações, consulte Configurações de endpoint para o Amazon S3 como fonte para AWS DMS.

É possível especificar o proprietário do bucket e evitar o corte utilizando a configuração `ExpectedBucketOwner` do endpoint do Amazon S3, conforme mostrado a seguir. Ao fazer uma solicitação para testar uma conexão ou executar uma migração, o S3 verifica o ID da conta do proprietário do bucket em relação ao parâmetro especificado.

```
--s3-settings='{"ExpectedBucketOwner": "AWS_Account_ID"}'
```

Tópicos

- Definindo tabelas externas para o Amazon S3 como fonte para AWS DMS

- Utilizar a CDC com o Amazon S3 como origem do AWS DMS
- Pré-requisitos ao usar o Amazon S3 como fonte para AWS DMS
- Limitações ao usar o Amazon S3 como fonte para AWS DMS
- Configurações de endpoint para o Amazon S3 como fonte para AWS DMS
- Tipos de dados de origem do Amazon S3
- Usando arquivos no formato Parquet no Amazon S3 como fonte para AWS DMS

Definindo tabelas externas para o Amazon S3 como fonte para AWS DMS

Além dos arquivos de dados, você também deve fornecer uma definição de tabela externa. Uma definição de tabela externa é um documento JSON que descreve como AWS DMS interpretar os dados do Amazon S3. O tamanho máximo deste documento é 2 MB. Se você criar um endpoint de origem usando o AWS DMS Management Console, poderá inserir o JSON diretamente na caixa de mapeamento de tabelas. Se você usar o AWS Command Line Interface (AWS CLI) ou a AWS DMS API para realizar migrações, poderá criar um arquivo JSON para especificar a definição da tabela externa.

Suponha que você tem um arquivo de dados que inclui o seguinte.

```
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
```

Veja a seguir um exemplo de definição de tabela externa para esses dados.

```
{
    "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"
        }
    ]
}
```

Os elementos neste documento JSON são os seguintes:

`TableCount`: o número de tabelas de origem. Neste exemplo, há somente uma tabela.

`Tables`: uma matriz que consiste em um mapa JSON por tabela de origem. Neste exemplo, há somente um mapa. Cada mapa consiste nos seguintes elementos:

- `TableName`: o nome da tabela de origem.
- `TablePath`: o caminho no bucket do Amazon S3 em que o AWS DMS pode encontrar o arquivo de carga máxima de dados. Se um valor `bucketFolder` for especificado, esse valor será pré-associado ao caminho.
- `TableOwner`: o nome do esquema desta tabela.
- `TableColumns`: uma matriz de um ou mais mapas, cada um dos quais descrevendo uma coluna na tabela de origem:
  - `ColumnName`: o nome de uma coluna na tabela de origem.

- `ColumnType`: o tipo de dados da coluna. Para tipos de dados válidos, consulte Tipos de dados de origem do Amazon S3.
- `ColumnLength`: o número de bytes nesta coluna. O comprimento máximo da coluna é limitado a 2147483647 bytes (2.047 MegaBytes), pois uma fonte S3 não oferece suporte ao modo FULL LOB. `ColumnLengthé` válido para os seguintes tipos de dados:
  - BYTE
  - STRING
- `ColumnNullable`: um valor booleano que será `true`, se esta coluna puder conter valores NULL (padrão=`false`).
- `ColumnIsPk`: um valor booleano que será `true`, se esta coluna fizer parte da chave primária (padrão=`false`).
- `ColumnDateFormat`: o formato de data de entrada para uma coluna com os tipos DATE, TIME e DATETIME e utilizado para analisar uma string de dados em um objeto de data. Os possíveis valores incluem:

```
- 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`: o número total de colunas. Esse número deve corresponder ao número de elementos no array `TableColumns`.

Se você não especificar o contrário, AWS DMS presume que `ColumnLength` seja zero.

> **Note**
>
> Nas versões compatíveis do AWS DMS, os dados de origem do S3 também podem conter uma coluna de operação opcional como a primeira coluna antes do valor da `TableName` coluna. Essa coluna de operação identifica a operação (`INSERT`) utilizada para migrar os dados para um endpoint de destino do S3 durante a carga máxima.

Se presente, o valor dessa coluna é o caractere inicial da palavra-chave de operação `INSERT` (`I`). Se especificado, essa coluna geralmente indica que a origem do S3 foi criada pelo DMS como um destino do S3 durante uma migração anterior.
Nas versões do DMS anteriores a 3.4.2, essa coluna não estava presente nos dados de origem do S3 criados em uma carga máxima do DMS anterior. Adicionar essa coluna aos dados de destino do S3 permite que o formato de todas as linhas gravadas no destino do S3 seja consistente se forem gravadas durante uma carga máxima ou durante uma carga de CDC. Para obter mais informações sobre as opções de formatação de dados de destino do S3, consulte Indicar operações de banco de dados de origem em dados migrados do S3.

Para obter uma coluna do tipo NUMERIC, especifique a precisão e a escala. Precision é o número total de dígitos em um número, e scale é o número de dígitos à direita do ponto decimal. Você utiliza os elementos `ColumnPrecision` e `ColumnScale` para isso, como mostrado a seguir.

```
...
    {
        "ColumnName": "HourlyRate",
        "ColumnType": "NUMERIC",
        "ColumnPrecision": "5"
        "ColumnScale": "2"
    }
...
```

Para uma coluna do tipo DATETIME com dados que contêm segundos fracionários, especifique a escala. A Escala é o número de dígitos dos segundos fracionários e pode variar de 0 a 9. Você utiliza o elemento `ColumnScale` para isso, conforme mostrado a seguir.

```
...
{
      "ColumnName": "HireDate",
      "ColumnType": "DATETIME",
      "ColumnScale": "3"
}
...
```

Se você não especificar o contrário, AWS DMS assume que `ColumnScale` é zero e trunca os segundos fracionários.

Utilizar a CDC com o Amazon S3 como origem do AWS DMS

Depois de AWS DMS realizar um carregamento completo de dados, ele pode, opcionalmente, replicar as alterações de dados no endpoint de destino. Para fazer isso, você carrega arquivos de captura de dados alterados (arquivos CDC) no seu bucket do Amazon S3. AWS DMS lê esses arquivos CDC quando você os carrega e, em seguida, aplica as alterações no endpoint de destino.

Os arquivos de CDC são nomeados como segue:

```
CDC00001.csv
CDC00002.csv
CDC00003.csv
...
```

> **Note**
>
> Para replicar com êxito os arquivos CDC na pasta de dados de alteração, faça upload em ordem léxica (sequencial). Por exemplo, faça upload do arquiva CDC00002.csv antes do arquiva CDC00003.csv. Caso contrário, CDC00002.csv será ignorado e não será replicado se carregado depois da CDC00003.csv. No entanto, o arquiva CDC00004.csv será replicado com êxito se carregado depois da CDC00003.csv.

Para indicar onde AWS DMS encontrar os arquivos, especifique o `cdcPath` parâmetro. Continuando o exemplo anterior, se você definir `cdcPath` como *`changedata`*, o AWS DMS lerá os arquivos de CDC no seguinte caminho.

```
s3://mybucket/changedata
```

Se você definir `cdcPath` como *`changedata`* e `bucketFolder` como *`myFolder`*, o AWS DMS lerá os arquivos de CDC no caminho a seguir.

```
s3://mybucket/myFolder/changedata
```

Os registros em um arquivo de CDC são formatados da seguinte forma:

- Operação: a operação de alteração a ser executada: `INSERT` ou `I`, `UPDATE` ou `U` ou `DELETE` ou `D`. Esses valores de caractere e palavras-chave não fazem distinção entre maiúsculas e minúsculas.

> **Note**
>
> Nas AWS DMS versões suportadas, AWS DMS pode identificar a operação a ser executada para cada registro de carga de duas maneiras. AWS DMS pode fazer isso a partir do valor da palavra-chave do registro (por exemplo,`INSERT`) ou do caractere inicial da palavra-chave (por exemplo,`I`). Nas versões anteriores, AWS DMS reconhecia a operação de carregamento somente a partir do valor completo da palavra-chave.
> Nas versões anteriores do AWS DMS, o valor completo da palavra-chave era gravado para registrar os dados do CDC. Além disso, as versões anteriores gravavam o valor da operação para qualquer destino do S3 utilizando apenas a inicial da palavra-chave.
> O reconhecimento dos dois formatos permite lidar com AWS DMS a operação, independentemente de como a coluna de operação é gravada para criar os dados de origem do S3. Essa abordagem é compatível com a utilização de dados de destino do S3 como origem para uma migração posterior. Com essa abordagem, você não precisa alterar o formato de nenhum valor inicial de palavra-chave que aparece na coluna da operação de origem do S3 posterior.

- Nome da tabela: o nome da tabela de origem.
- Nome do esquema: o nome do esquema de origem.
- Dados: uma ou mais colunas que representam os dados a serem alterados.

Veja a seguir um exemplo de arquivo de CDC para uma tabela chamada `employee`.

```
INSERT,employee,hr,101,Smith,Bob,2014-06-04,New York
UPDATE,employee,hr,101,Smith,Bob,2015-10-08,Los Angeles
UPDATE,employee,hr,101,Smith,Bob,2017-03-13,Dallas
DELETE,employee,hr,101,Smith,Bob,2017-03-13,Dallas
```

Pré-requisitos ao usar o Amazon S3 como fonte para AWS DMS

Para usar o Amazon S3 como fonte AWS DMS, seu bucket S3 de origem deve estar na mesma AWS região da instância de replicação do DMS que migra seus dados. Além disso, a conta da AWS que você utiliza para a migração deve ter acesso de leitura ao bucket de origem.

A função AWS Identity and Access Management (IAM) atribuída à conta de usuário usada para criar a tarefa de migração deve ter o seguinte conjunto de permissões.

```
{
    "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*"
            ]
        }
    ]
}
```

A função AWS Identity and Access Management (IAM) atribuída à conta de usuário usada para criar a tarefa de migração deve ter o seguinte conjunto de permissões se o controle de versão estiver habilitado no bucket do Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "S3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::mybucket*/*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::mybucket*"
            ]
        }
    ]
}
```

Limitações ao usar o Amazon S3 como fonte para AWS DMS

As limitações a seguir se aplicam ao utilizar o Amazon S3 como origem:

- Não ative o versionamento para o S3. Se o versionamento do S3 for necessário, utilize políticas de ciclo de vida para excluir ativamente as versões antigas. Caso contrário, é possível encontrar falhas na conexão de teste de endpoint devido ao tempo limite de uma chamada `list-object` do S3. Para criar uma política de ciclo de vida para um bucket do S3, consulte Gerenciar o ciclo de vida do armazenamento. Para excluir a versão de um objeto do S3, consulte Excluir versões de objetos de um bucket com versionamento ativado.
- Um bucket do S3 ativado para VPC (VPC do gateway) é compatível com as versões 3.4.7 e superiores.
- O MySQL converte o tipo de dados em. `time` `string` Para ver os valores do tipo de **`time`** dados no MySQL, defina a coluna na tabela de destino como **`string`** e defina a configuração do modo de preparação da tabela de destino da tarefa como Truncar.
- AWS DMS usa o tipo de `BYTE` dados internamente para dados em ambos os tipos `BYTE` de `BYTES` dados.
- Os endpoints de origem do S3 não oferecem suporte ao recurso de recarga de tabela do DMS.
- AWS DMS não suporta o modo LOB completo com o Amazon S3 como fonte.

As seguintes limitações se aplicam ao usar arquivos no formato Parquet no Amazon S3 como fonte:

- As datas entram `MMYYYYDD` ou não `DDMMYYYY` são suportadas pelo recurso de particionamento de data do S3 Parquet Source.

Configurações de endpoint para o Amazon S3 como fonte para AWS DMS

É possível utilizar as configurações do endpoint para configurar o banco de dados de origem do Amazon S3 de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de origem usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--s3-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o Amazon S3 como origem.

Opção	Descrição
`BucketFolder`	(Opcional) Um nome da pasta no bucket do S3. Se o atributo de origem for fornecido, os arquivos de dados de origem e os arquivos da CDC serão lidos no caminho `s3://myBucket/bucketFolder/schemaName/tableName/` e `s3://myBucket/bucketFolder/` respectivamente. Se esse atributo não for especificado, o caminho utilizado será `schemaName/tableName/`. `'{"BucketFolder": " sourceData "}'`
`BucketName`	O nome do bucket do S3. `'{"BucketName": " myBucket"}'`
`CdcPath`	A localização dos arquivos da CDC. Esse atributo é necessário quando uma tarefa captura dados de alterações; caso contrário, ele é opcional. Se `CdcPath` estiver presente, AWS DMS lê os arquivos CDC desse caminho e replica as alterações de dados no endpoint de destino. Para ter mais informações, consulte Utilizar a CDC com o Amazon S3 como origem do AWS DMS. `'{"CdcPath": " changeData "}'`
`CsvDelimiter`	O delimitador utilizado para separar colunas nos arquivos de origem. O padrão é uma vírgula. Veja a seguir um exemplo. `'{"CsvDelimiter": ","}'`

Opção	Descrição
`CsvNullValue`	Uma string definida pelo usuário que é AWS DMS tratada como nula ao ler a partir da fonte. O padrão é uma string vazia. Se você não definir esse parâmetro, AWS DMS tratará uma string vazia como um valor nulo. Se você definir esse parâmetro como uma string como “\ N”, AWS DMS tratará essa string como o valor nulo e tratará as strings vazias como um valor de string vazio.
`CsvRowDelimiter`	O delimitador utilizado para separar linhas nos arquivos de origem. O padrão é uma nova linha (\n). `'{"CsvRowDelimiter": "\n"}'`
`DataFormat`	Defina esse valor `Parquet` para ler dados no formato Parquet. `'{"DataFormat": "Parquet"}'`
`IgnoreHeaderRows`	Quando esse valor é definido como 1, AWS DMS ignora o cabeçalho da primeira linha em um arquivo.csv. Um valor de 1 habilita o recurso, um valor de 0 desabilita o recurso. O padrão é 0. `'{"IgnoreHeaderRows": 1}'`
`Rfc4180`	Quando esse valor é definido como `true` ou `y`, as aspas duplas de abertura devem ser seguidas por aspas duplas de fechamento. Essa formatação está em conformidade com RFC 4180. Quando esse valor for definido para `false` ou `n`, os literais das strings serão copiados no destino como estão. Nesse caso, um delimitador (linha ou coluna) sinaliza o final do campo. Assim, você não poderá utilizar um delimitador como parte da string, pois ele sinalizará o final do valor. O padrão é `true`. Valores válidos: `true`, `false`, `y`, `n` `'{"Rfc4180": false}'`

Tipos de dados de origem do Amazon S3

Migração de dados que usa o Amazon S3 como fonte para AWS DMS necessidades de mapear dados do Amazon S3 AWS DMS para tipos de dados. Para ter mais informações, consulte Definindo tabelas externas para o Amazon S3 como fonte para AWS DMS.

Para obter informações sobre como visualizar o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que você está usando.

Para obter informações adicionais sobre tipos de AWS DMS dados, consulteTipos de dados do AWS Database Migration Service.

Os seguintes tipos de AWS DMS dados são usados com o Amazon S3 como fonte:

- BYTE: requer `ColumnLength`. Para ter mais informações, consulte Definindo tabelas externas para o Amazon S3 como fonte para AWS DMS.
- DATA
- TIME
- DATETIME: para obter mais informações e um exemplo, consulte o exemplo do tipo DATETIME em Definindo tabelas externas para o Amazon S3 como fonte para AWS DMS.
- INT1
- INT2
- INT4
- INT8
- NUMÉRICO — Requer `ColumnPrecision` e. `ColumnScale` AWS DMS suporta os seguintes valores máximos:
  - ColumnPrecision: 38
  - ColumnScale: 31

  Para obter mais informações e um exemplo, consulte o exemplo do tipo NUMERIC em Definindo tabelas externas para o Amazon S3 como fonte para AWS DMS.
- REAL4
- REAL8
- STRING: requer `ColumnLength`. Para ter mais informações, consulte Definindo tabelas externas para o Amazon S3 como fonte para AWS DMS.

- UINT1
- UINT2
- UINT4
- UINT8
- BLOB
- CLOB
- BOOLEAN

Usando arquivos no formato Parquet no Amazon S3 como fonte para AWS DMS

Na AWS DMS versão 3.5.3 e posterior, você pode usar arquivos no formato Parquet em um bucket do S3 como fonte para replicação de carga completa ou CDC.

O DMS só oferece suporte a arquivos no formato Parquet como fonte que o DMS gera ao migrar dados para um endpoint de destino do S3. Os nomes dos arquivos devem estar no formato compatível, ou o DMS não os incluirá na migração.

Para arquivos de dados de origem no formato Parquet, eles devem estar na seguinte pasta e convenção de nomenclatura.

```
schema/table1/LOAD00001.parquet
schema/table2/LOAD00002.parquet
schema/table2/LOAD00003.parquet
```

Para arquivos de dados de origem para dados CDC no formato Parquet, nomeie-os e armazene-os usando a seguinte pasta e convenção de nomenclatura.

```
schema/table/20230405-094615814.parquet
schema/table/20230405-094615853.parquet
schema/table/20230405-094615922.parquet
```

Para acessar arquivos no formato Parquet, defina as seguintes configurações de endpoint:

- Defina `DataFormat` como `Parquet`.
- Não defina a `cdcPath` configuração. Certifique-se de criar seus arquivos no formato Parquet nas pastas de esquema/tabela especificadas.

Para obter mais informações sobre as configurações dos endpoints do S3, consulte S3Settings na Referência da API.AWS Database Migration Service

Tipos de dados compatíveis com arquivos no formato Parquet

AWS DMS suporta os seguintes tipos de dados de origem e destino ao migrar dados de arquivos no formato Parquet. Certifique-se de que sua tabela de destino tenha colunas dos tipos de dados corretos antes de migrar.

Tipo de dados de origem	Tipo de dados de destino
BYTE	BINARY
DATE	DATE32
TIME	TIME32
DATETIME	TIMESTAMP
INT1	INT8
INT2	INT16
INT4	INT32
INT8	INT64
NUMERIC	DECIMAL
REAL4	FLOAT
REAL8	DOUBLE
STRING	STRING
UINT1	UINT8
UINT2	UINT16
UINT4	UINT32
UINT8	UINT

Tipo de dados de origem	Tipo de dados de destino
WSTRING	STRING
BLOB	BINARY
NCLOB	STRING
CLOB	STRING
BOOLEAN	BOOL

Usando o banco de dados IBM Db2 para Linux, Unix, Windows e Amazon RDS (Db2 LUW) como fonte para AWS DMS

Você pode migrar dados de um banco de dados IBM Db2 para Linux, Unix, Windows e Amazon RDS (Db2 LUW) para qualquer banco de dados de destino compatível usando (). AWS Database Migration Service AWS DMS

Para obter informações sobre as versões do Db2 no Linux, Unix, Windows e RDS que oferecem AWS DMS suporte como fonte, consulte. Fontes para AWS DMS

É possível utilizar SSL para criptografar conexões entre o endpoint do Db2 LUW e a instância de replicação. Para obter mais informações sobre a utilização de SSL com um endpoint do Db2 LUW, consulte Usando SSL com AWS Database Migration Service.

Pré-requisitos ao usar o Db2 LUW como fonte para AWS DMS

Os pré-requisitos a seguir são necessários antes de utilizar um banco de dados Db2 LUW como origem.

Para habilitar a replicação contínua, também chamada de captura de dados de alteração (CDC), faça o seguinte:

- Defina o banco de dados para ser recuperável, o que AWS DMS requer a captura de alterações. Um banco de dados será recuperável se um ou os dois parâmetros de configuração de banco de dados LOGARCHMETH1 e LOGARCHMETH2 estiverem definidos como ON.

  Se seu banco de dados for recuperável, AWS DMS poderá acessar o Db2, ARCHIVE LOG se necessário.

- Certifique-se de que os registros de transações do DB2 estejam disponíveis, com um período de retenção suficiente para serem processados. AWS DMS
- O DB2 requer autorização de `SYSADM` ou de `DBADM` para extrair registros de log de transações. Conceda à conta do usuário as seguintes permissões:
  - `SYSADM` ou `DBADM`
  - `DATAACCESS`

  > **Note**
  >
  > Para tarefas somente de carga máxima, a conta de usuário do DMS precisa da permissão DATAACCESS.

- Ao utilizar o IBM DB2 for LUW versão 9.7 como origem, defina o atributo de conexão adicional (ECA), `CurrentLSN`, da seguinte forma:

  `CurrentLSN=`*`LSN`* em que *`LSN`* especifica um número de sequência de log (LSN) em que você deseja que a replicação seja iniciada. Ou, `CurrentLSN=`*`scan`*.

Limitações ao usar o Db2 LUW como fonte para AWS DMS

AWS DMS não oferece suporte a bancos de dados em cluster. No entanto, é possível definir um Db2 LUW separado para cada um dos endpoints de um cluster. Por exemplo, é possível criar uma tarefa de migração de carga máxima com qualquer um dos nós no cluster e criar tarefas separadas em cada nó.

AWS DMS não suporta o tipo de `BOOLEAN` dados em seu banco de dados Db2 LUW de origem.

Ao utilizar a replicação contínua (CDC), aplicam-se as seguintes limitações:

- Quando uma tabela com várias partições é truncada, o número de eventos DDL mostrados no AWS DMS console é igual ao número de partições. Isso ocorre porque o Db2 LUW registra um DDL separado para cada partição.
- As ações de DDL a seguir não são compatíveis em tabelas particionadas:
  - ALTER TABLE ADD PARTITION
  - ALTER TABLE DETACH PARTITION
  - ALTER TABLE ATTACH PARTITION

- AWS DMS não suporta uma migração de replicação contínua de uma instância em espera de recuperação de desastres de alta disponibilidade (HADR) do DB2. O modo de espera é inacessível.
- O tipo de dados DECFLOAT não é compatível. Consequentemente, alterações nas colunas DECFLOAT são ignoradas durante a replicação contínua.
- A instrução RENAME COLUMN não é compatível.
- Ao realizar atualizações nas tabelas de agrupamento multidimensional (MDC), cada atualização é mostrada no AWS DMS console como INSERT + DELETE.
- Quando a configuração da tarefa Incluir colunas LOB na replicação não está ativada, qualquer tabela que tenha com colunas LOB é suspensa durante a replicação contínua.
- Para as versões 10.5 e superiores do Db2 LUW, as colunas de string de comprimento variável com dados armazenados são ignoradas. out-of-row Essa limitação se aplica somente a tabelas criadas com tamanho de linha estendido para colunas com tipos de dados, como VARCHAR e VARGRAPHIC. Para contornar essa limitação, mova a tabela para um espaço de tabela com um tamanho de página maior. Para obter mais informações, consulte O que posso fazer se eu quiser alterar o tamanho da página dos espaços de tabela do DB2.
- Para a replicação contínua, o DMS não é compatível com a migração de dados carregados em nível da página pelo utilitário DB2 LOAD. Em vez disso, utilize o utilitário IMPORT que utiliza inserções SQL. Para obter mais informações, consulte diferenças entre os utilitários de importação e de carregamento.
- Enquanto uma tarefa de replicação está em execução, o DMS captura DDLs CREATE TABLE somente se as tabelas tiverem sido criadas com o atributo DATA CAPTURE CHANGE.
- O DMS tem as seguintes limitações ao usar o Db2 Database Partition Feature (DPF):
  - O DMS não pode coordenar transações entre nós do Db2 em um ambiente DPF. Isso se deve às restrições na interface da API IBM DB2READLOG. No DPF, as transações podem abranger vários nós do Db2, dependendo de como o DB2 particiona os dados. Como resultado, sua solução DMS deve capturar transações de cada nó do Db2 de forma independente.
  - O DMS pode capturar transações locais de cada nó do Db2 no cluster do DPF configurando 1 em vários endpoints de `connectNode` origem do DMS. Essa configuração corresponde aos números de nós lógicos definidos no arquivo `db2nodes.cfg` de configuração do servidor DB2.
  - As transações locais em nós individuais do Db2 podem ser partes de uma transação global maior. O DMS aplica cada transação local de forma independente no destino, sem coordenação com transações em outros nós do Db2. Esse processamento independente pode levar a complicações, especialmente quando as linhas são movidas entre as partições.

- Quando o DMS se replica de vários nós do Db2, não há garantia da ordem correta das operações no destino, porque o DMS aplica as operações de forma independente para cada nó do Db2. Você deve garantir que a captura de transações locais independentemente de cada nó do Db2 funcione para seu caso de uso específico.
- Ao migrar de um ambiente DPF, recomendamos primeiro executar uma tarefa de carregamento completo sem eventos em cache e, em seguida, executar tarefas somente CDC. Recomendamos executar uma tarefa por nó do Db2, começando pelo timestamp de início do Full Load ou pelo LRI (identificador de registro de log) definido usando a configuração do endpoint. `StartFromContext` Para obter informações sobre como determinar seu ponto inicial de replicação, consulte Encontrando o valor LSN ou LRI para o início da replicação na documentação do IBM Support.

- Para replicação contínua (CDC), se você planejar iniciar a replicação a partir de um timestamp específico, defina o atributo de conexão `StartFromContext` com o timestamp requerido.
- Atualmente, o DMS não é compatível com o Db2 pureScale Feature, uma extensão do DB2 LUW que é possível utilizar para escalar a solução de banco de dados.
- AWS DMS não oferece suporte ao CDC ao usar o Db2 for Amazon RDS como fonte.

Configurações de endpoint ao usar o Db2 LUW como fonte para AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados de destino do Db2 LUW de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de origem usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o Db2 LUW como origem.

Nome	Descrição
`CurrentLSN`	Para replicação contínua (CDC), utilize `CurrentLSN` para especificar um número de sequência de log (LSN) em que você deseja que a replicação seja iniciada.
`MaxKBytesPerRead`	Número máximo de bytes por leitura, como um valor NUMBER. O padrão é 64 KB.

Nome	Descrição
`SetDataCaptureChanges`	Habilita a replicação contínua (CDC) como um valor BOOLEAN. O padrão é true.

Nome	Descrição

`StartFromContext`	Para replicação contínua (CDC), utilize `StartFromContext` para especificar o limite inferior de um log em que iniciar a replicação. `StartFromContext` aceita diferentes formas de valores. Os valores válidos são:
	• `timestamp` (UTC). Por exemplo: .
	`'{"StartFromContext": "timestamp:2021-09-21T13:00:00"}'`
	• `NOW`
	Para o IBM DB2 LUW versão 10.5 e superior, o NOW combinado com a verificação CurrentLSN: inicia a tarefa a partir do LSO mais recente. Por exemplo: .
	`'{"CurrentLSN": "scan", "StartFromContext": "NOW"}'`
	• Um LRI específico. Por exemplo: .
	`'{"StartFromContext": "0100000000000022C000000000004FB13"}'`
	Para determinar o intervalo de LRI/LSN de um arquivo de log, execute o comando `db2flsn` conforme mostrado no exemplo a seguir.
	`db2flsn -db SAMPLE -lrirange 2`
	O resultado desse exemplo é semelhante ao exemplo a seguir.

Nome	Descrição

	`S0000002.LOG: has LRI range 00000000000000100 00000000002254000000000004F9A6 to 000000000000000100000000000022CC000000000004 FB13`
	
	Nessa saída, o arquivo de log é S0000002.LOG e o valor do StartFromContextLRI são os 34 bytes no final do intervalo.
	
	`0100000000000022CC000000000004FB13`

Tipos de dados de origem para IBM Db2 LUW

A migração de dados que usa o Db2 LUW como fonte para AWS DMS suportar a maioria dos tipos de dados do Db2 LUW. A tabela a seguir mostra os tipos de dados de origem do Db2 LUW que são suportados durante o uso AWS DMS e o mapeamento padrão dos tipos de AWS DMS dados. Para obter mais informações sobre os tipos de dados do Db2 LUW, consulte a documentação do Db2 LUW.

Para obter informações sobre como visualizar o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que você está usando.

Para obter informações adicionais sobre AWS DMS os tipos de dados, consulteTipos de dados do AWS Database Migration Service.

Tipos de dados do Db2 LUW	AWS DMS tipos de dados
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s)
FLOAT	REAL8

Tipos de dados do Db2 LUW	AWS DMS tipos de dados
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Se a precisão for 16, REAL8; se a precisão for 34, STRING
GRAPHIC (n)	WSTRING, para strings de gráficos de comprimento fixo de caracteres de byte duplo com um comprimento maior que 0 e menor ou igual a 127
VARGRAPHIC (n)	WSTRING, para strings de gráficos de comprimento variável com um comprimen to maior que 0 e menor ou igual a 16.352 caracteres de byte duplo.
LONG VARGRAPHIC (n)	CLOB, para strings de gráficos de comprimen to variável com um comprimento maior que 0 e menor ou igual a 16.352 caracteres de byte duplo.
CHARACTER (n)	STRING, para strings de comprimento fixo de caracteres de byte duplo com um comprimento maior que 0 e menor ou igual a 255
VARCHAR (n)	STRING, para strings de comprimento variável de caracteres de byte duplo com um comprimento maior que 0 e menor ou igual a 32.704
LONG VARCHAR (n)	CLOB, para strings de comprimento variável de caracteres de byte duplo com um comprimento maior que 0 e menor ou igual a 32.704
CHAR (n) FOR BIT DATA	BYTES

Tipos de dados do Db2 LUW	AWS DMS tipos de dados
VARCHAR (n) FOR BIT DATA	BYTES
LONG VARCHAR FOR BIT DATA	BYTES
DATA	DATA
TIME	TIME
TIMESTAMP	DATETIME
BLOB (n)	BLOB O comprimento máximo é 2.147.483.647 bytes
CLOB (n)	CLOB O comprimento máximo é 2.147.483.647 bytes
DBCLOB (n)	CLOB O comprimento máximo é 1.073.741.824 caracteres de byte duplo
XML	CLOB

Utilizar o bancos de dados IBM Db2 for z/OS como origem do AWS DMS

É possível migrar dados de um banco de dados IBM for z/OS para qualquer banco de dados de destino compatível utilizando o AWS Database Migration Service (AWS DMS).

Para obter informações sobre as versões do Db2 for z/OS compatíveis com o AWS DMS como origem, consulte Fontes para AWS DMS.

Pré-requisitos ao utilizar o Db2 for z/OS como origem do AWS DMS

Para utilizar um banco de dados IBM Db2 for z/OS como origem no AWS DMS, conceda os seguintes privilégios ao usuário do Db2 for z/OS especificado nas configurações de conexão do endpoint.

```
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
```

Também conceda SELECT ON em tabelas de origem *user defined*.

Um endpoint de origem do IBM Db2 for z/OS do AWS DMS depende do IBM Data Server Driver for ODBC para acessar os dados. O servidor de banco de dados deve ter uma licença válida do IBM ODBC Connect para que o DMS se conecte a esse endpoint.

Limitações ao utilizar o Db2 for z/OS como origem do AWS DMS

As seguintes limitações se aplicam ao utilizar um banco de dados IBM Db2 for z/OS como origem do AWS DMS:

- Somente tarefas de replicação de carga máxima são compatíveis. A captura de dados de alteração (CDC) não é compatível.
- A carga paralela não é compatível.
- A validação de dados das visualizações não é compatível.
- Os nomes de esquema, tabela e colunas devem ser especificados em letras maiúsculas nos mapeamentos de tabela para transformações em nível de coluna/tabela e de filtros de seleção em nível de linha.

Tipos de dados de origem do IBM Db2 for z/OS

As migrações de dados que utilizam o Db2 for z/OS como origem do AWS DMS são compatíveis com a maioria dos tipos de dados do Db2 for z/OS. A tabela a seguir mostra os tipos de dados de origem do Db2 do z/OS compatíveis ao utilizar o AWS DMS e o mapeamento padrão de tipos de dados do AWS DMS.

Para obter mais informações sobre os tipos de dados do Db2 for z/OS, consulte a Documentação do IBM Db2 for z/OS.

Para obter informações sobre como visualizar o tipo de dados mapeado no destino, consulte a seção relativa ao endpoint de destino que você está utilizando.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipos de dados do Db2 for z/OS	Tipos de dados do AWS DMS
INTEGER	INT4
SMALLINT	INT2
BIGINT	INT8
DECIMAL (p,s)	NUMERIC (p,s) Se um ponto decimal estiver definido como uma vírgula (,) na configuração do DB2, configure a replicação para que seja compatível com a configuração do DB2.
FLOAT	REAL8
DOUBLE	REAL8
REAL	REAL4
DECFLOAT (p)	Se a precisão for 16, REAL8; se a precisão for 34, STRING
GRAPHIC (n)	Se n>= 127 que WSTRING, para strings de gráficos de tamanho fixo de strings de caracteres de byte duplo com um tamanho maior que 0 e menor que ou igual a 127
VARGRAPHIC (n)	WSTRING, para strings de gráficos de comprimento variável com um comprimento maior que 0 e menor ou igual a 16.352 caracteres de byte duplo.
LONG VARGRAPHIC (n)	CLOB, para strings de gráficos de comprimento variável com um comprimento maior que 0

Tipos de dados do Db2 for z/OS	Tipos de dados do AWS DMS
	e menor ou igual a 16.352 caracteres de byte duplo.
CHARACTER (n)	STRING, para strings de comprimento fixo de caracteres de byte duplo com um comprimento maior que 0 e menor ou igual a 255
VARCHAR (n)	STRING, para strings de comprimento variável de caracteres de byte duplo com um comprimento maior que 0 e menor ou igual a 32.704
LONG VARCHAR (n)	CLOB, para strings de comprimento variável de caracteres de byte duplo com um comprimento maior que 0 e menor ou igual a 32.704
CHAR (n) FOR BIT DATA	BYTES
VARCHAR (n) FOR BIT DATA	BYTES
LONG VARCHAR FOR BIT DATA	BYTES
DATE	DATE
TIME	TIME
TIMESTAMP	DATETIME
BLOB (n)	BLOB O comprimento máximo é 2.147.483.647 bytes
CLOB (n)	CLOB O comprimento máximo é 2.147.483.647 bytes

Tipos de dados do Db2 for z/OS	Tipos de dados do AWS DMS
DBCLOB (n)	CLOB O comprimento máximo é 1.073.741.824 caracteres de byte duplo
XML	CLOB
BINARY	BYTES
VARBINARY	BYTES
ROWID	BYTES. Para obter mais informações sobre como trabalhar com ROWID, consulte o seguinte.
TIMESTAMP WITH TIME ZONE	Não compatível.

As colunas ROWID são migradas por padrão quando o modo de preparação da tabela de destino para a tarefa é definido como DROP_AND_CREATE (o padrão). A validação de dados ignora essas colunas porque as linhas não têm sentido fora do banco de dados e da tabela específicos. Para desativar a migração dessas colunas, é possível executar uma das seguintes etapas preparatórias:

- Pré-crie a tabela de destino sem essas colunas. Defina o modo de preparação da tabela de destino da tarefa como DO_NOTHING ou TRUNCATE_BEFORE_LOAD. É possível utilizar o AWS Schema Conversion Tool (AWS SCT) para pré-criar a tabela de destino sem as colunas.
- Adicione uma regra de mapeamento de tabela a uma tarefa que filtra essas colunas para que elas sejam ignoradas. Para obter mais informações, consulte Regras de transformação e ações.

Agrupamentos EBCDIC no PostgreSQL para o serviço de modernização de mainframe da AWS

O AWS programa AWS Mainframe Modernization ajuda a modernizar aplicações de mainframe para ambientes de runtime gerenciados da AWS. Ele fornece ferramentas e recursos para ajudar a planejar e implementar os projetos de migração e de modernização. Para obter mais informações sobre modernização e migração de mainframe, consulte Modernização do mainframe com a AWS.

Alguns conjuntos de dados IBM Db2 for z/OS são codificados no conjunto de caracteres Extended Binary Coded Decimal Interchange (EBCDIC). Esse é um conjunto de caracteres que foi desenvolvido antes do ASCII (American Standard Code for Information Interchange) se tornar comumente utilizado. Uma página de código mapeia cada caractere do texto para os caracteres em um conjunto de caracteres. Uma página de código tradicional contém as informações de mapeamento entre um ponto de código e um ID de caractere. Um ID de caractere é uma string de dados de caracteres de 8 bytes. Um ponto de código é um número binário de 8 bits que representa um caractere. Os pontos de código geralmente são mostrados como representações hexadecimais de seus valores binários.

Se você utilizar o componente Micro Focus ou BluAge do serviço Mainframe Modernization no momento, informe ao AWS DMS para mudar (traduzir) certos pontos de código. É possível utilizar as configurações da tarefa do AWS DMS para executar as mudanças. O exemplo a seguir mostra como utilizar a operação `CharacterSetSettings` do AWS DMS para mapear a mudanças em uma configuração de tarefa do DMS.

```
"CharacterSetSettings": {
        "CharacterSetSupport": null,
        "CharacterReplacements": [
{"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
            }
        ]
    }
```

Já existem alguns agrupamentos de EBCDIC para o PostgreSQL que compreendem a mudança necessária. Várias páginas de código diferentes são compatíveis. As seções a seguir fornecem exemplos de JSON do que você deve mudar para todas as páginas de código compatíveis. É possível simplesmente copiar e colar o JSON necessário na tarefa do DMS.

Agrupamentos de EBCDIC específicos da Micro Focus

Para a Micro Focus, mude um subconjunto de caracteres conforme necessário para os seguintes agrupamentos.

```
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 Mudanças de dados da Micro Focus por agrupamento:

en_us_cp1140m

Mudança de código:

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1141m

Mudança de código:

```
0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1142m

Mudança de código:

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1143m

Mudança de código:

```
0000    0180
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1144m

Mudança de código:

```
0000    0180
```

```
00B8    0160
00BC    0161
00BD    017D
00BE    017E
00A8    0152
00B4    0153
00A6    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1145m

Mudança de código:

```
0000    0180
00A6    0160
00B8    0161
00A8    017D
00BC    017E
00BD    0152
00BE    0153
00B4    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "017D"}
```

```
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1146m

Mudança de código:

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

en_us_cp1147m

Mudança de código:

```
0000    0180
00B8    0160
00A8    0161
00BC    017D
00BD    017E
```

```
00BE    0152
00B4    0153
00A6    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0178"}
```

en_us_cp1148m

Mudança de código:

```
0000    0180
00A6    0160
00B8    0161
00BC    017D
00BD    017E
00BE    0152
00A8    0153
00B4    0178
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0000","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "00A6","TargetCharacterCodePoint": "0160"}
,{"SourceCharacterCodePoint": "00B8","TargetCharacterCodePoint": "0161"}
,{"SourceCharacterCodePoint": "00BC","TargetCharacterCodePoint": "017D"}
,{"SourceCharacterCodePoint": "00BD","TargetCharacterCodePoint": "017E"}
,{"SourceCharacterCodePoint": "00BE","TargetCharacterCodePoint": "0152"}
,{"SourceCharacterCodePoint": "00A8","TargetCharacterCodePoint": "0153"}
,{"SourceCharacterCodePoint": "00B4","TargetCharacterCodePoint": "0178"}
```

Agrupamentos de EBCDIC específicos do BluAge

Para o BluAge, altere todos os seguintes valores baixos e valores altos conforme necessário. Esses agrupamentos só devem ser utilizados para compatibilidade com o serviço Mainframe Migration BluAge.

```
da-DK-cp1142b-x-icu
 da-DK-cp277b-x-icu
 de-DE-cp1141b-x-icu
 de-DE-cp273b-x-icu
 en-GB-cp1146b-x-icu
 en-GB-cp285b-x-icu
 en-US-cp037b-x-icu
 en-US-cp1140b-x-icu
 es-ES-cp1145b-x-icu
 es-ES-cp284b-x-icu
 fi-FI-cp1143b-x-icu
 fi-FI-cp278b-x-icu
 fr-FR-cp1147b-x-icu
 fr-FR-cp297b-x-icu
 it-IT-cp1144b-x-icu
 it-IT-cp280b-x-icu
 nl-BE-cp1148b-x-icu
 nl-BE-cp500b-x-icu
```

Example Mudanças de dados do BluAge:

da-DK-cp277b e da-DK-cp1142b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
```

```
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

de-DE-273b e de-DE-1141b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
```

```
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
```

```
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

en-GB-285b e en-GB-1146b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
{"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
```

```
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
```

```
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

en-us-037b e en-us-1140b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
{"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
```

```
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

es-ES-284b e es-ES-1145b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
```

```
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

fi_FI-278b e fi-FI-1143b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
```

```
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

fr-FR-297b efr-FR-1147b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
{"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
```

```
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
```

```
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

it-IT-280b e it-IT-1144b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
```

```
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
```

```
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

nl-BE-500b e nl-BE-1148b

Mudança de código:

```
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
```

Mapeamento de entrada correspondente para uma tarefa do AWS DMS:

```
 {"SourceCharacterCodePoint": "0180","TargetCharacterCodePoint": "0180"}
,{"SourceCharacterCodePoint": "0001","TargetCharacterCodePoint": "0181"}
,{"SourceCharacterCodePoint": "0002","TargetCharacterCodePoint": "0182"}
,{"SourceCharacterCodePoint": "0003","TargetCharacterCodePoint": "0183"}
,{"SourceCharacterCodePoint": "009C","TargetCharacterCodePoint": "0184"}
,{"SourceCharacterCodePoint": "0009","TargetCharacterCodePoint": "0185"}
,{"SourceCharacterCodePoint": "0086","TargetCharacterCodePoint": "0186"}
,{"SourceCharacterCodePoint": "007F","TargetCharacterCodePoint": "0187"}
,{"SourceCharacterCodePoint": "0097","TargetCharacterCodePoint": "0188"}
,{"SourceCharacterCodePoint": "008D","TargetCharacterCodePoint": "0189"}
,{"SourceCharacterCodePoint": "008E","TargetCharacterCodePoint": "018A"}
,{"SourceCharacterCodePoint": "000B","TargetCharacterCodePoint": "018B"}
,{"SourceCharacterCodePoint": "000C","TargetCharacterCodePoint": "018C"}
,{"SourceCharacterCodePoint": "000D","TargetCharacterCodePoint": "018D"}
,{"SourceCharacterCodePoint": "000E","TargetCharacterCodePoint": "018E"}
,{"SourceCharacterCodePoint": "000F","TargetCharacterCodePoint": "018F"}
,{"SourceCharacterCodePoint": "0010","TargetCharacterCodePoint": "0190"}
,{"SourceCharacterCodePoint": "0011","TargetCharacterCodePoint": "0191"}
,{"SourceCharacterCodePoint": "0012","TargetCharacterCodePoint": "0192"}
,{"SourceCharacterCodePoint": "0013","TargetCharacterCodePoint": "0193"}
,{"SourceCharacterCodePoint": "009D","TargetCharacterCodePoint": "0194"}
,{"SourceCharacterCodePoint": "0085","TargetCharacterCodePoint": "0195"}
,{"SourceCharacterCodePoint": "0008","TargetCharacterCodePoint": "0196"}
,{"SourceCharacterCodePoint": "0087","TargetCharacterCodePoint": "0197"}
,{"SourceCharacterCodePoint": "0018","TargetCharacterCodePoint": "0198"}
,{"SourceCharacterCodePoint": "0019","TargetCharacterCodePoint": "0199"}
,{"SourceCharacterCodePoint": "0092","TargetCharacterCodePoint": "019A"}
,{"SourceCharacterCodePoint": "008F","TargetCharacterCodePoint": "019B"}
,{"SourceCharacterCodePoint": "001C","TargetCharacterCodePoint": "019C"}
,{"SourceCharacterCodePoint": "001D","TargetCharacterCodePoint": "019D"}
,{"SourceCharacterCodePoint": "001E","TargetCharacterCodePoint": "019E"}
,{"SourceCharacterCodePoint": "001F","TargetCharacterCodePoint": "019F"}
,{"SourceCharacterCodePoint": "0080","TargetCharacterCodePoint": "01A0"}
,{"SourceCharacterCodePoint": "0081","TargetCharacterCodePoint": "01A1"}
,{"SourceCharacterCodePoint": "0082","TargetCharacterCodePoint": "01A2"}
,{"SourceCharacterCodePoint": "0083","TargetCharacterCodePoint": "01A3"}
,{"SourceCharacterCodePoint": "0084","TargetCharacterCodePoint": "01A4"}
,{"SourceCharacterCodePoint": "000A","TargetCharacterCodePoint": "01A5"}
```

```
,{"SourceCharacterCodePoint": "0017","TargetCharacterCodePoint": "01A6"}
,{"SourceCharacterCodePoint": "001B","TargetCharacterCodePoint": "01A7"}
,{"SourceCharacterCodePoint": "0088","TargetCharacterCodePoint": "01A8"}
,{"SourceCharacterCodePoint": "0089","TargetCharacterCodePoint": "01A9"}
,{"SourceCharacterCodePoint": "008A","TargetCharacterCodePoint": "01AA"}
,{"SourceCharacterCodePoint": "008B","TargetCharacterCodePoint": "01AB"}
,{"SourceCharacterCodePoint": "008C","TargetCharacterCodePoint": "01AC"}
,{"SourceCharacterCodePoint": "0005","TargetCharacterCodePoint": "01AD"}
,{"SourceCharacterCodePoint": "0006","TargetCharacterCodePoint": "01AE"}
,{"SourceCharacterCodePoint": "0007","TargetCharacterCodePoint": "01AF"}
,{"SourceCharacterCodePoint": "0090","TargetCharacterCodePoint": "01B0"}
,{"SourceCharacterCodePoint": "0091","TargetCharacterCodePoint": "01B1"}
,{"SourceCharacterCodePoint": "0016","TargetCharacterCodePoint": "01B2"}
,{"SourceCharacterCodePoint": "0093","TargetCharacterCodePoint": "01B3"}
,{"SourceCharacterCodePoint": "0094","TargetCharacterCodePoint": "01B4"}
,{"SourceCharacterCodePoint": "0095","TargetCharacterCodePoint": "01B5"}
,{"SourceCharacterCodePoint": "0096","TargetCharacterCodePoint": "01B6"}
,{"SourceCharacterCodePoint": "0004","TargetCharacterCodePoint": "01B7"}
,{"SourceCharacterCodePoint": "0098","TargetCharacterCodePoint": "01B8"}
,{"SourceCharacterCodePoint": "0099","TargetCharacterCodePoint": "01B9"}
,{"SourceCharacterCodePoint": "009A","TargetCharacterCodePoint": "01BA"}
,{"SourceCharacterCodePoint": "009B","TargetCharacterCodePoint": "01BB"}
,{"SourceCharacterCodePoint": "0014","TargetCharacterCodePoint": "01BC"}
,{"SourceCharacterCodePoint": "0015","TargetCharacterCodePoint": "01BD"}
,{"SourceCharacterCodePoint": "009E","TargetCharacterCodePoint": "01BE"}
,{"SourceCharacterCodePoint": "001A","TargetCharacterCodePoint": "01BF"}
,{"SourceCharacterCodePoint": "009F","TargetCharacterCodePoint": "027F"}
```

Destinos para a migração de dados

O AWS Database Migration Service (AWS DMS) pode utilizar muitos dos bancos de dados mais populares como destino da replicação de dados. O destino pode estar em uma instância do Amazon Elastic Compute Cloud (Amazon EC2), uma instância do Amazon Relational Database Service (Amazon RDS) ou em um banco de dados on-premises.

Para obter uma lista abrangente de destinos válidos, consulte Destinos do AWS DMS.

> **Note**
>
> O AWS DMS não é compatível com a migração entre regiões da AWS para os seguintes tipos de endpoint de destino:
>
> - Amazon DynamoDB
> - Amazon OpenSearch Service
> - Amazon Kinesis Data Streams

Tópicos

Utilizar um banco de dados Oracle como destino do AWS Database Migration Service

Você pode migrar dados para destinos de banco de dados Oracle usando AWS DMS, seja de outro banco de dados Oracle ou de um dos outros bancos de dados suportados. É possível utilizar Secure Sockets Layer (SSL) para criptografar as conexões entre o endpoint do Oracle e a instância de replicação. Para obter mais informações sobre o uso de SSL com um endpoint Oracle, consulte. Usando SSL com AWS Database Migration Service AWS DMS também suporta o uso da criptografia transparente de dados (TDE) da Oracle para criptografar dados em repouso no banco de dados de destino, pois o Oracle TDE não exige uma chave ou senha de criptografia para gravar no banco de dados.

Para obter informações sobre as versões do Oracle que oferecem AWS DMS suporte como destino, consulteMetas para AWS DMS.

Ao utilizar o Oracle como destino, pressupomos que os dados devam ser migrados para o esquema ou usuário utilizado para a conexão de destino. Se você quiser migrar dados para um esquema diferente, use uma transformação de esquema. Por exemplo, suponha que seu endpoint de destino se conecte ao usuário `RDSMASTER` e você queira migrar do usuário `PERFDATA1` para `PERFDATA2`. Nesse caso, crie uma transformação como a seguinte.

```
{
   "rule-type": "transformation",
   "rule-id": "2",
   "rule-name": "2",
   "rule-action": "rename",
   "rule-target": "schema",
   "object-locator": {
   "schema-name": "PERFDATA1"
},
"value": "PERFDATA2"
}
```

Ao usar o Oracle como destino, AWS DMS migra todas as tabelas e índices para os espaços de tabela padrão e de índice no destino. Para migrar tabelas e índices para diferentes espaços para tabela de índices e tabelas, utilize uma transformação de espaço para tabela para fazer isso. Por exemplo, suponha que você tenha um conjunto de tabelas no esquema `INVENTORY` atribuído

a alguns espaços de tabela na origem do Oracle. Para a migração, você deseja atribuir todas essas tabelas a um único espaço de tabela `INVENTORYSPACE` no destino. Nesse caso, crie uma transformação como a seguinte.

```
{
   "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"
}
```

Para obter mais informações sobre transformações, consulte Especificar a seleção de tabelas e as regras de transformação utilizando JSON.

Se o Oracle for origem e destino, será possível preservar as atribuições de tablespace de tabela ou índice existentes definindo o atributo de conexão extra de origem Oracle, `enableHomogenousTablespace=true`. Para mais informações, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.

Para obter detalhes adicionais sobre como trabalhar com bancos de dados Oracle como destino AWS DMS, consulte as seções a seguir:

Tópicos

- Limitações do Oracle como alvo para AWS Database Migration Service
- Privilégios da conta de usuário necessários para utilizar o Oracle como destino
- Configurando um banco de dados Oracle como destino para AWS Database Migration Service
- Configurações de endpoint ao usar o Oracle como destino para AWS DMS
- Tipos de dados de destino do Oracle

Limitações do Oracle como alvo para AWS Database Migration Service

Veja a seguir as limitações ao utilizar o Oracle como destino para a migração de dados:

- AWS DMS não cria esquema no banco de dados Oracle de destino. Você deve criar todos os esquemas que deseja no banco de dados de destino do Oracle. O nome do esquema já deve existir para o destino do Oracle. As tabelas do esquema de origem são importadas para o usuário ou esquema, que é AWS DMS usado para se conectar à instância de destino. Para migrar vários esquemas, crie várias tarefas de replicação. Também é possível migrar dados para diferentes esquemas em um destino. Para fazer isso, você precisa usar as regras de transformação do esquema nos mapeamentos da AWS DMS tabela.
- AWS DMS não suporta a `Use direct path full load` opção de tabelas com INDEXTYPE CONTEXT. Como solução, use o carregamento de matriz.
- Com a opção de aplicar o lote otimizado, o carregamento na tabela de alterações líquidas usa um caminho direto, que não é compatível com o tipo XML. Como solução, use o modo de aplicação transacional.
- As strings vazias migradas de bancos de dados de origem podem ser tratadas de forma diferente pelo destino do Oracle (convertido em strings de um espaço, por exemplo). Isso pode resultar na AWS DMS validação relatando uma incompatibilidade.
- É possível expressar o número total de colunas por tabela compatível com o modo de aplicação otimizado em lote, utilizando a seguinte fórmula:

  ```
  2 * columns_in_original_table + columns_in_primary_key <= 999
  ```

  Por exemplo, se a tabela original tiver 25 colunas e a sua chave primária consistir em 5 colunas, o número total de colunas será 55. Se uma tabela exceder o número suportado de colunas, todas as alterações serão aplicadas no one-by-one modo.
- AWS DMS não oferece suporte ao banco de dados autônomo no Oracle Cloud Infrastructure (OCI).

Privilégios da conta de usuário necessários para utilizar o Oracle como destino

Para usar um alvo Oracle em uma AWS Database Migration Service tarefa, conceda os seguintes privilégios no banco de dados Oracle. É possível conceder esses privilégios à conta do usuário especificada nas definições do banco de dados Oracle do AWS DMS.

- SELECT ANY TRANSACTION
- SELECT on V$NLS_PARAMETERS
- SELECT on V$TIMEZONE_NAMES

- SELECT on ALL_INDEXES
- SELECT on ALL_OBJECTS
- SELECT on DBA_OBJECTS
- SELECT on ALL_TABLES
- SELECT on ALL_USERS
- SELECT on ALL_CATALOG
- SELECT on ALL_CONSTRAINTS
- SELECT on ALL_CONS_COLUMNS
- SELECT on ALL_TAB_COLS
- SELECT on ALL_IND_COLUMNS
- DROP ANY TABLE
- SELECT ANY TABLE
- INSERT ANY TABLE
- UPDATE ANY TABLE
- CREATE ANY VIEW
- DROP ANY VIEW
- CREATE ANY PROCEDURE
- ALTER ANY PROCEDURE
- DROP ANY PROCEDURE
- CREATE ANY SEQUENCE
- ALTER ANY SEQUENCE
- DROP ANY SEQUENCE
- DELETE ANY TABLE

Para os requisitos a seguir, conceda estes privilégios adicionais:

- Para utilizar uma lista de tabela específica, conceda SELECT em qualquer tabela replicada e ALTER em qualquer tabela replicada.
- Para permitir que um usuário crie uma tabela no espaço de tabela padrão, conceda o privilégio GRANT UNLIMITED TABLESPACE.
- Para fazer logon, conceda o privilégio CREATE SESSION.

- Ao utilizar um caminho direto (que é o padrão para carga máxima), `GRANT LOCK ANY TABLE to dms_user;`.
- Se o esquema for diferente ao utilizar o modo de preparação de tabela “DROP e CREATE”, `GRANT CREATE ANY INDEX to dms_user;`.
- Para alguns cenários de carga máxima, é possível escolher a opção “DROP and CREATE table” ou “TRUNCATE before loading” em que um esquema de tabela de destino é diferente daquele do usuário do DMS. Nesse caso, conceda DROP ANY TABLE.
- Para armazenar alterações em tabelas de alterações ou em uma tabela de auditoria em que o esquema da tabela de destino é diferente daquele do usuário do DMS, conceda CREATE ANY TABLE e CREATE ANY INDEX.

Privilégios de leitura necessários para o AWS Database Migration Service banco de dados de destino

A conta AWS DMS do usuário deve receber permissões de leitura para as seguintes tabelas do DBA:

- SELECT on DBA_USERS
- SELECT on DBA_TAB_PRIVS
- SELECT on DBA_OBJECTS
- SELECT on DBA_SYNONYMS
- SELECT on DBA_SEQUENCES
- SELECT on DBA_TYPES
- SELECT on DBA_INDEXES
- SELECT on DBA_TABLES
- SELECT on DBA_TRIGGERS
- SELECT on SYS.DBA_REGISTRY

Se algum dos privilégios necessários não puder ser concedido a V$xxx, conceda-o a V_$xxx.

Avaliações de pré-migração

Para usar as avaliações de pré-migração listadas Avaliações da Oracle com o Oracle como destino, você deve adicionar as seguintes permissões ao usuário do banco de `dms_user` dados no banco de dados de destino:

```
GRANT SELECT ON V_$INSTANCE TO dms_user;
```

Configurando um banco de dados Oracle como destino para AWS Database Migration Service

Antes de usar um banco de dados Oracle como destino de migração de dados, você deve fornecer uma conta de usuário Oracle para AWS DMS o. A conta de usuário deve ter privilégios de leitura/gravação no banco de dados Oracle, como especificado na seção Privilégios da conta de usuário necessários para utilizar o Oracle como destino.

Configurações de endpoint ao usar o Oracle como destino para AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados de destino do Oracle de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--oracle-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o Oracle como destino.

Nome	Descrição
`EscapeCharacter`	Defina esse atributo como um caractere de escape. Esse caractere de escape permite que um único caractere curinga se comporte como um caractere normal em expressões de mapeamento de tabela. Para ter mais informações, consulte Curingas no mapeamento de tabela. Valor padrão: nulo Valores válidos: qualquer caractere que não seja um caractere curinga Exemplo: `--oracle-settings '{"EscapeCharacter": "#"}'`

Nome	Descrição
`UseDirectPathFullLoad`	Quando definido comoY, AWS DMS usa um caminho direto com carga total. Especifique esse valor para habilitar o protocolo de caminho direto na OCI (Oracle Call Interface). Esse protocolo OCI permite a carga em massa de tabelas de destino do Oracle durante um carga máxima. Valor padrão: `true` Valores válidos: `true/false` Exemplo: `--oracle-settings '{"UseDirectPathFullLoad": false}'`
`DirectPathParallelLoad`	Quando definido como `true`, este atributo especifica uma carga paralela quando `UseDirectPathFullLoad` é definido como `Y`. Esse atributo também se aplica somente quando você usa o recurso de carregamento AWS DMS paralelo. Para obter mais informações, consulte a descrição da operação `parallel-load` em Regras e operações de configurações de tabelas e coleções. Uma limitação na especificação dessa configuração de carga paralela é que a tabela de destino não pode ter restrições nem índices. Para obter mais informações sobre essa limitação, consulte Habilitar restrições após uma carga paralela de caminho direto. Se restrições ou índices estiverem habilitados, definir esse atributo como `true` não terá efeito. Valor padrão: `false` Valores válidos: `true/false` Exemplo: `--oracle-settings '{"DirectPathParallelLoad": true}'`

Nome	Descrição
`DirectPathNoLog`	Quando definido como `true`, esse atributo ajuda a aumentar a taxa de confirmação no banco de dados de destino Oracle gravando diretamente nas tabelas e não gravando uma trilha nos logs do banco de dados. Para obter mais informações, consulte Carga direta INSERT. Esse atributo também se aplica somente quando `UseDirectPathFullLoad` é definido como `Y`. Valor padrão: `false` Valores válidos: `true/false` Exemplo: `--oracle-settings '{"DirectPathNoLog": true}'`
`CharLengthSemantics`	Especifica se o comprimento de uma coluna de caracteres está em bytes ou em caracteres. Para indicar que o comprimento da coluna de caracteres está em caracteres, defina esse atributo como `CHAR`. Caso contrário, o comprimento da coluna de caracteres está em bytes. Valor padrão: não definido como `CHAR` Valores válidos: `CHAR` Exemplo: `--oracle-settings '{"CharLengthSemantics": "CHAR"}'`

Nome	Descrição
`AlwaysReplaceEmptyString`	AWS DMS adiciona um espaço extra para replicar uma string vazia ao migrar para um destino Oracle. Em geral, o Oracle não tem notação para uma string vazia. Ao inserir uma string vazia em varchar2, você carrega strings vazias como NULL. Para inserir os dados como NULL no Oracle, defina esse atributo como FALSE. Valor padrão: `true` Valores válidos: `true/false` Exemplo: `--oracle-settings '{"Always ReplaceEmptyString": false}'`

Tipos de dados de destino do Oracle

Um banco de dados Oracle de destino usado com AWS DMS suporta a maioria dos tipos de dados Oracle. A tabela a seguir mostra os tipos de dados de destino da Oracle que são suportados durante o uso AWS DMS e o mapeamento padrão dos tipos de AWS DMS dados. Para obter mais informações sobre como visualizar o tipo de dados mapeado da origem, consulte a seção relativa à origem que você está usando.

AWS DMS tipo de dados	Tipo de dados do Oracle
BOOLEAN	NUMBER (1)
BYTES	RAW (tamanho)
DATA	DATETIME
TIME	TIMESTAMP (0)
DATETIME	TIMESTAMP (escala)
INT1	NUMBER (3)
INT2	NUMBER (5)

AWS DMS tipo de dados	Tipo de dados do Oracle
INT4	NUMBER (10)
INT8	NUMBER (19)
NUMERIC	NUMBER (p,s)
REAL4	FLOAT
REAL8	FLOAT
STRING	Com indicação de data: DATE Com indicação de tempo: TIMESTAMP Com indicação de time stamp: TIMESTAMP Com indicação de time stamp com fuso horário: TIMESTAMP WITH TIMEZONE Com indicação de time stamp com fuso horário local: TIMESTAMP WITH LOCAL TIMEZONE Com indicação de intervalo de tempo em anos e meses: INTERVAL YEAR TO MONTH Com indicação de intervalo de tempo em dias e segundos: INTERVAL DAY TO SECOND Se comprimento > 4.000: CLOB Em todos os demais casos: VARCHAR2 (comprimento)
UINT1	NUMBER (3)
UINT2	NUMBER (5)
UINT4	NUMBER (10)
UINT8	NUMBER (19)

AWS DMS tipo de dados	Tipo de dados do Oracle
WSTRING	Se o comprimento > 2000: NCLOB Em todos os demais casos: NVARCHAR2 (comprimento)
BLOB	BLOB Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de BLOBs para uma tarefa específica. Os tipos de dados BLOB são compatíveis somente com tabelas que possuem uma chave primária
CLOB	CLOB Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de CLOBs para uma tarefa específica. Durante uma captura de dados de alteração (CDC), os tipos de dados CLOB são compatíve is somente em tabelas que incluem uma chave primária. STRING Um tipo de dados Oracle VARCHAR2 na origem com um tamanho declarado maior que 4000 bytes mapeia através do AWS DMS CLOB para uma STRING no destino Oracle.
NCLOB	NCLOB Para usar esse tipo de dados com AWS DMS, você deve habilitar o uso de NCLobs para uma tarefa específica. Durante uma captura de dados de alteração (CDC), os tipos de dados NCLOB são compatíveis somente com tabelas que possuem uma chave primária. WSTRING Um tipo de dados Oracle VARCHAR2 na origem com um tamanho declarado maior que 4000 bytes mapeia por meio do AWS DMS NCLOB para uma WSTRING no destino Oracle.

AWS DMS tipo de dados	Tipo de dados do Oracle
XMLTYPE	O tipo de dados de destino XMLTYPE só é utilizado em tarefas de replicação de Oracle para Oracle. Quando o banco de dados de origem é o Oracle, os tipos de dados de origem são replicados como estão para o destino do Oracle. Por exemplo, um tipo de dados XMLTYPE na origem é criado como um tipo de dados XMLTYPE no destino.

Utilizar um banco de dados Microsoft SQL Server como destino do AWS Database Migration Service

É possível migrar dados para bancos de dados Microsoft SQL Server utilizando o AWS DMS. Com um banco de dados SQL Server como destino, é possível migrar dados de outro banco de dados SQL Server ou de um dos outros bancos de dados compatíveis.

Para obter informações sobre as versões do SQL Server que são compatíveis com o AWS DMS como destino, consulte Metas para AWS DMS.

O AWS DMS é compatível com as edições on-premises Enterprise, Standard, Workgroup e Developer do Amazon RDS.

Para obter mais detalhes sobre o trabalho com o AWS DMS e bancos de dados de destino SQL Server, consulte:

Tópicos

Limitações ao utilizar o SQL Server como destino do AWS Database Migration Service

As seguintes limitações se aplicam à utilização de um banco de dados SQL Server como destino do AWS DMS:

- Ao criar manualmente uma tabela de destino do SQL Server com uma coluna calculada, a replicação de carga máxima não é compatível ao utilizar o utilitário de cópia em massa BCP. Para utilizar a replicação de carga máxima, desative o carregamento de BCP definindo o atributo de conexão adicional (ECA) `'useBCPFullLoad=false'` no endpoint. Para obter informações sobre como configurar ECAs em endpoints, consulte Criar endpoints de origem e de destino. Para obter mais informações sobre como trabalhar com o BCP, consulte a documentação do Microsoft SQL Server.
- Ao replicar tabelas com tipos de dados espaciais do SQL Server (GEOMETRY e GEOGRAPHY), o AWS DMS substitui qualquer identificador de referência espacial (SRID) que você tenha pelo SRID padrão. O SRID padrão é 0 para GEOMETRY e 4326 para GEOGRAPHY.
- Tabelas temporais não são compatíveis. A migração de tabelas temporais pode funcionar com uma tarefa somente replicação no modo de aplicação transacional se essas tabelas forem criadas manualmente no destino.
- Atualmente, os tipos de dados `boolean` em uma origem do PostgreSQL são migrados para um destino do SQL Server como o tipo de dados `bit` com valores inconsistentes.

  Como alternativa, faça o seguinte:

  - Crie previamente a tabela com um tipo de dados `VARCHAR(1)` para a coluna (ou deixe o AWS DMS criar a tabela). Depois, deixe o processamento downstream tratar um "F" como Falso e um "T" como Verdadeiro.
  - Para evitar a necessidade de alterar o processamento downstream, adicione uma regra de transformação à tarefa para alterar os valores “F” para “0" e os valores “T” para 1 e armazená-los como o tipo de dados de bits do servidor SQL.
- O AWS DMS não é compatível com o processamento de alterações para definir a nulidade da coluna (utilizando a cláusula `ALTER COLUMN [SET|DROP] NOT NULL` com instruções `ALTER TABLE`).
- A Autenticação do Windows não é compatível.

Requisitos de segurança ao utilizar o SQL Server como destino do AWS Database Migration Service

Veja a seguir a descrição dos requisitos de segurança para utilizar o AWS DMS com um destino do Microsoft SQL Server:

- A conta do usuário do AWS DMS deve ter pelo menos o perfil do usuário `db_owner` no banco de dados SQL Server ao qual você está se conectando.
- Um administrador de sistema do SQL Server deve fornecer essa permissão a todas as contas de usuário do AWS DMS.

Configurações de endpoint ao utilizar o SQL Server como destino do AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados de destino do SQL Server de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--microsoft-sql-server-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o SQL Server como destino.

Nome	Descrição
`ControlTablesFileGroup`	Especifique um grupo de arquivos para as tabelas internas do AWS DMS. Quando a tarefa de replicaçã o é iniciada, todas as tabelas internas de controle do AWS DMS (awsdms_ apply_exception, awsdms_apply, awsdms_changes) são criadas no grupo de arquivos especificado. Valor padrão: n/d Valores válidos: string Exemplo: `--microsoft-sql-server-settings '{"ControlTablesFileGroup": "filegrou p1"}'` O exemplo a seguir apresenta instruções para criar um grupo de arquivos. `ALTER DATABASE replicate ADD FILEGROUP Test1FG1; GO ALTER DATABASE replicate`

Nome	Descrição
	`ADD FILE ( NAME = test1dat5, FILENAME = 'C:\temp\DATA\t1dat5.ndf', SIZE = 5MB, MAXSIZE = 100MB, FILEGROWTH = 5MB ) TO FILEGROUP Test1FG1; GO`
`ExecuteTimeout`	Utilize esse atributo de conexão adicional (ECA) para definir o tempo limite da instrução do cliente para a instância do SQL Server, em segundos. O valor padrão é de 60 segundos. Exemplo: `'{"ExecuteTimeout": 100}'`
`UseBCPFullLoad`	Utilize esse atributo para transferir dados para operações de carga máxima utilizando BCP. Quando a tabela de destino contém uma coluna de identidade que não existe na tabela de origem, desative a opção Utilizar BCP ao carregar tabelas. Valor padrão: verdadeiro Valores válidos: verdadeiro/falso Exemplo: `--microsoft-sql-server-settings '{"UseBCPFullLoad": false}'`

Tipos de dados de destino do Microsoft SQL Server

A tabela a seguir mostra os tipos de dados de destino do Microsoft SQL Server compatíveis com o AWS DMS e o mapeamento padrão relativo aos tipos de dados do AWS DMS. Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipo de dados do AWS DMS	Tipo de dados do SQL Server
BOOLEAN	TINYINT
BYTES	VARBINARY(tamanho)
DATA	No SQL Server 2008 e superior, utilize DATE. Para versões anteriores, se a escala for menor ou igual a 3, use DATETIME. Em todos os demais casos, use VARCHAR (37).
TIME	No SQL Server 2008 e superior, utilize DATETIME2 (%d). Para versões anteriores, se a escala for menor ou igual a 3, use DATETIME. Em todos os demais casos, use VARCHAR (37).
DATETIME	No SQL Server 2008 e superior, utilize DATETIME2 (escala). Para versões anteriores, se a escala for menor ou igual a 3, use DATETIME. Em todos os demais casos, use VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	FLOAT
STRING	Se a coluna for de data ou hora, faça o seguinte: • No SQL Server 2008 e superior, utilize DATETIME2. • Para versões anteriores, se a escala for menor ou igual a 3, use DATETIME. Em todos os demais casos, use VARCHAR (37).

Tipo de dados do AWS DMS	Tipo de dados do SQL Server
	Se a coluna não é uma data ou hora, use VARCHAR (tamanho).
UINT1	TINYINT
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR (tamanho)
BLOB	VARBINARY(máximo) IMAGE Para utilizar esse tipo de dados com o AWS DMS, ative a utilizaçã o de BLOBs em uma tarefa específica. O AWS DMS é compatíve l com os tipos de dados BLOB somente em tabelas que possuem uma chave primária.
CLOB	VARCHAR(máximo) Para usar esse tipo de dados no AWS DMS, é necessário habilitar o uso de CLOBs em uma tarefa específica. Durante uma captura de dados de alteração (CDC), o AWS DMS é compatível com os tipos de dados CLOB somente em tabelas que incluem uma chave primária.
NCLOB	NVARCHAR(máximo) Para usar esse tipo de dados no AWS DMS, é necessário habilitar o uso de NCLOBs em uma tarefa específica. Durante uma captura de dados de alteração (CDC), o AWS DMS oferece suporte aos tipos de dados NCLOB somente em tabelas que incluem uma chave primária.

Utilizar um banco de dados PostgreSQL como destino do AWS Database Migration Service

Você pode migrar dados para bancos de dados PostgreSQL AWS DMS usando, seja de outro banco de dados PostgreSQL ou de um dos outros bancos de dados compatíveis.

Para obter informações sobre as versões do PostgreSQL AWS DMS que oferecem suporte como destino, consulte. Metas para AWS DMS

> **Note**
>
> - O Amazon Aurora Serverless está disponível como destino para o Amazon Aurora com compatibilidade com o PostgreSQL. Para obter mais informações sobre o Amazon Aurora Serverless, consulte Usando o Amazon Aurora Serverless v2 no Guia do usuário do Amazon Aurora.
> - Os clusters de banco de dados do Aurora Sem Servidor são acessíveis apenas de uma Amazon VPC e não podem utilizar um Endereço IP público. Portanto, se você tiver uma instância de replicação em uma região diferente da do Aurora PostgreSQL Sem Servidor, configure o Emparelhamento da VPC. Caso contrário, verifique a disponibilidade das regiões do Aurora PostgreSQL Sem Servidor e decida utilizar uma dessas regiões para o Aurora PostgreSQL Sem Servidor e a instância de replicação.
> - O recurso Babelfish está incorporado ao Amazon Aurora sem custo adicional. Para obter mais informações, consulte Utilizar o Babelfish para Aurora PostgreSQL como destino do AWS Database Migration Service.

AWS DMS adota uma table-by-table abordagem ao migrar dados da origem para o destino na fase de carga total. Não é possível garantir a ordem da tabela durante a fase de Carregamento total. As tabelas ficam dessincronizadas durante a fase de Carregamento total e enquanto transações em cache de tabelas individuais estão sendo aplicadas. Como resultado, restrições de integridade referencial ativa podem gerar falhas de tarefas durante a fase de Carregamento total.

No PostgreSQL, chaves externas (restrições de integridade referencial) são implementadas usando triggers. Durante a fase de carga total, AWS DMS carrega cada tabela uma de cada vez. Recomendamos que você desative as restrições de chave externa durante um carregamento total, usando um dos seguintes métodos:

- Desative temporariamente todos os triggers da instância e conclua o carregamento total.
- Use o parâmetro `session_replication_role` no PostgreSQL.

Em determinado momento, um trigger pode estar em um dos seguintes estados: `origin`, `replica`, `always`, ou `disabled`. Quando o parâmetro `session_replication_role` é definido como `replica`, apenas triggers no estado `replica` ficam ativos, e eles são disparados quando chamados. Caso contrário, os triggers permanecem inativos.

PostgreSQL tem um mecanismo à prova de falhas para impedir que uma tabela seja truncada, mesmo quando `session_replication_role` é definido. É possível utilizar isso como alternativa para desativar os acionadores, para ajudar a executar a carga máxima até a conclusão. Para isso, defina o modo de preparação da tabela de destino `DO_NOTHING`. Caso contrário, as operações DROP e TRUNCATE falharão quando houver restrições de chave externa.

No Amazon RDS, é possível controlar esse parâmetro utilizando um grupo de parâmetros. Para uma instância do PostgreSQL em execução no Amazon EC2, é possível definir o parâmetro diretamente.

Para obter detalhes adicionais sobre como trabalhar com um banco de dados PostgreSQL como destino, consulte as AWS DMS seções a seguir:

Tópicos

Limitações no uso do PostgreSQL como alvo para AWS Database Migration Service

As seguintes limitações aplicam-se à utilização de um banco de dados PostgreSQL como destino para o AWS DMS:

- Para migrações heterogêneas, o tipo de dados JSON é convertido internamente no tipo de dados CLOB nativo.

- Em uma migração do Oracle para o PostgreSQL, se uma coluna no Oracle contiver um caractere NULL (valor hexadecimal U+0000), converterá o caractere NULL em um espaço (valor hexadecimal U+0020) AWS DMS . Isso deve-se a uma limitação do PostgreSQL.
- AWS DMS não oferece suporte à replicação para uma tabela com um índice exclusivo criado com a função coalesce.
- Se suas tabelas usarem sequências, atualize o valor de NEXTVAL para cada sequência no banco de dados de destino depois de interromper a replicação do banco de dados de origem. AWS DMS copia dados do seu banco de dados de origem, mas não migra sequências para o destino durante a replicação contínua.

Requisitos de segurança ao usar um banco de dados PostgreSQL como alvo para AWS Database Migration Service

Para fins de segurança, a conta de usuário usada para a migração de dados deve ser um usuário registrado em qualquer banco de dados PostgreSQL que você use como destino.

Seu endpoint de destino do PostgreSQL exige permissões mínimas de usuário para executar AWS DMS uma migração. Veja os exemplos a seguir.

```
    CREATE USER newuser WITH PASSWORD 'your-password';
    ALTER SCHEMA schema_name OWNER TO newuser;
```

Ou

```
    GRANT USAGE ON SCHEMA schema_name TO myuser;
    GRANT CONNECT ON DATABASE postgres to myuser;
    GRANT CREATE ON DATABASE postgres TO myuser;
    GRANT CREATE ON SCHEMA schema_name TO myuser;
    GRANT UPDATE, INSERT, SELECT, DELETE, TRUNCATE ON ALL TABLES IN SCHEMA schema_name
TO myuser;
    GRANT TRUNCATE ON schema_name."BasicFeed" TO myuser;
```

Configurações de endpoint e atributos extras de conexão (ECAs) ao usar o PostgreSQL como destino para AWS DMS

Você pode usar configurações de endpoint e Atributos de Conexão Extra (ECAs) para configurar seu banco de dados de destino do PostgreSQL.

Você especifica as configurações ao criar o endpoint de destino usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--postgre-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

Você especifica ECAs usando o `ExtraConnectionAttributes` parâmetro para seu endpoint.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o PostgreSQL como destino.

Nome	Descrição
`MaxFileSize`	Especifica o tamanho máximo (em KB) de um arquivo .csv usado para transferir dados para o PostgreSQL. Valor padrão: 32768 KB (32 MB) Valores válidos: 1 a 1.048.576 KB (até 1.1 GB) Exemplo: `--postgre-sql-settings '{"MaxFileSize": 512}'`
`ExecuteTimeout`	Define o tempo limite da instrução do cliente para a instância do PostgreSQL, em segundos. O valor padrão é de 60 segundos. Exemplo: `--postgre-sql-settings '{"ExecuteTimeout": 100}'`
`AfterConnectScript= SET session_replication_role = replica`	Esse atributo AWS DMS ignora chaves estrangeiras e acionadores de usuário para reduzir o tempo necessário para carregar dados em massa.

Nome	Descrição
`MapUnboundedNumericAsString`	Esse parâmetro trata colunas com tipos de dados NUMERIC ilimitados como STRING para migrar com sucesso sem perder a precisão do valor numérico. Utilize esse parâmetro somente para replicação da origem do PostgreSQL para o destino do PostgreSQL ou bancos de dados compatíveis com o PostgreSQL. Valor padrão: falso Valores válidos: falso/verdadeiro Exemplo: `--postgre-sql-settings '{"MapUnboundedNumericAsString": "true"}` A utilização desse parâmetro pode resultar em alguma degradação do desempenho da replicação devido à transformação de numérico para string e de volta para numérico. Esse parâmetro é compatível para utilização pelo DMS versão 3.4.4 e superior **Note** Utilize `MapUnboundedNumericAsString` somente em endpoints de origem e de destino do PostgreSQL juntos. A utilização de `MapUnboundedNumericAsString` em endpoints PostgreSQL de origem restringe a precisão a 28 durante a CDC. A utilização de `MapUnboundedNumericAsString` em endpoints de destino migra os dados com Precisão 28 Escala 6. Não utilize `MapUnboundedNumericAsString` com destinos que não sejam do PostgreSQL.

Nome	Descrição
`loadUsingCSV`	Use esse Atributo de Conexão Extra (ECA) para transferi r dados para operações de carga total usando o comando \ COPY. Valor padrão: `true` Valores válidos: verdadeiro/falso Exemplo de ECA: `loadUsingCSV=true;` Observação: definir esse ECA como falso pode resultar em alguma degradação do desempenho da replicação devido aos INSERTs serem executados diretamente.
`DatabaseMode`	Utilize esse atributo para alterar o comportamento padrão do tratamento da replicação de endpoints compatíve is com PostgreSQL que exigem alguma configuração adicional, como os endpoints do Babelfish. Valor padrão: `DEFAULT` Valores válidos: `DEFAULT`, `BABELFISH` Exemplo: `DatabaseMode=default;`
`BabelfishDatabaseName`	Utilize esse atributo para especificar o nome do banco de dados Babelfish T-SQL de destino que está sendo migrado. Isso será obrigatório se `DatabaseMode` estiver definido como `Babelfish` . Esse não é o banco de dados `babelfish_db` reservado. Exemplo: `BabelfishDatabaseName=TargetDb;`

Tipos de dados de destino do PostgreSQL

O endpoint do banco de dados PostgreSQL é AWS DMS compatível com a maioria dos tipos de dados do banco de dados PostgreSQL. A tabela a seguir mostra os tipos de dados de destino do

banco de dados PostgreSQL que são compatíveis com o AWS DMS uso e o mapeamento AWS DMS padrão dos tipos de dados.

Para obter informações adicionais sobre tipos de AWS DMS dados, consulteTipos de dados do AWS Database Migration Service.

AWS DMS tipo de dados	Tipo de dados do PostgreSQL
BOOLEAN	BOOLEAN
BLOB	BYTEA
BYTES	BYTEA
DATA	DATA
TIME	TIME
DATETIME	Se a escala for de 0 a 6, use TIMESTAMP. Se a escala for de 7 a 9, use VARCHAR (37).
INT1	SMALLINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (P,S)
REAL4	FLOAT4
REAL8	FLOAT8
STRING	Se o tamanho for de 1 a 21.845, use VARCHAR (tamanho em bytes). Se o tamanho for de 21.846 a 2.147.483.647, use VARCHAR (65535).

AWS DMS tipo de dados	Tipo de dados do PostgreSQL
UINT1	SMALLINT
UINT2	INTEGER
UINT4	BIGINT
UINT8	BIGINT
WSTRING	Se o tamanho for de 1 a 21.845, use VARCHAR (tamanho em bytes). Se o tamanho for de 21.846 a 2.147.483.647, use VARCHAR (65535).
NCLOB	TEXT
CLOB	TEXT

> **Note**
>
> Ao replicar de uma fonte do PostgreSQL AWS DMS , cria a tabela de destino com os mesmos tipos de dados para todas as colunas, exceto as colunas com tipos de dados definidos pelo usuário. Nesses casos, o tipo de dados é criado como "variante de caractere" no destino.

Usando o Babelfish para Aurora PostgreSQL como alvo para AWS Database Migration Service

É possível migrar as tabelas de origem do SQL Server para um destino do Babelfish para Amazon Aurora PostgreSQL utilizando o AWS Database Migration Service. Com o Babelfish, o Aurora PostgreSQL compreende o dialeto do T-SQL, do SQL proprietário do Microsoft SQL Server, e é compatível com o mesmo protocolo de comunicação. Portanto, as aplicações escritas para o SQL Server agora podem funcionar com o Aurora com menos alterações no código. O recurso Babelfish está incorporado ao Amazon Aurora sem custo adicional. É possível ativar o Babelfish no cluster do Amazon Aurora no console do Amazon RDS.

Ao criar seu endpoint de AWS DMS destino usando os comandos do AWS DMS console, da API ou da CLI, especifique o mecanismo de destino como Amazon Aurora PostgreSQL e nomeie o banco de dados como babelfish_db. Na seção Configurações de endpoint, adicione configurações para definir `DatabaseMode` como `Babelfish` e `BabelfishDatabaseName` para o nome do banco de dados Babelfish T-SQL de destino.

Adicionar regras de transformação à tarefa de migração

Ao definir uma tarefa de migração para um destino do Babelfish, inclua regras de transformação que garantam que o DMS utilize as tabelas T-SQL do Babelfish pré-criadas no banco de dados de destino.

Primeiro, adicione uma regra de transformação à tarefa de migração que coloque todos os nomes das tabelas em letras minúsculas. O Babelfish armazena em letras minúsculas no `pg_class` catálogo do PostgreSQL os nomes das tabelas que você cria utilizando o T-SQL. No entanto, quando você tem tabelas do SQL Server com nomes em maiúsculas e minúsculas, o DMS cria as tabelas utilizando tipos de dados nativos do PostgreSQL em vez dos tipos de dados compatíveis com T-SQL. Por esse motivo, adicione uma regra de transformação que torne todos os nomes de tabelas em minúsculas. Observe que os nomes de coluna não devem ser mudados para minúsculas.

Se você utilizou o modo de migração de vários bancos de dados ao definir o cluster, adicione uma regra de transformação que renomeia o esquema original do SQL Server. Renomeie o esquema do SQL Server para incluir o nome do banco de dados T-SQL. Por exemplo, se o nome do esquema original do SQL Server for dbo e o nome do banco de dados T-SQL for `mydb`, renomeie o esquema para `mydb_dbo` utilizando uma regra de transformação.

Se você utilizar o modo de banco de dados único, não será necessária uma regra de transformação para renomear os esquemas. Os nomes dos esquemas têm um one-to-one mapeamento com o banco de dados T-SQL de destino no Babelfish.

O exemplo de regra de transformação a seguir torna todos os nomes de tabelas em minúsculas e renomeia o esquema original do SQL Server de dbo para `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": []
      }
]
}
```

Limitações ao utilizar um endpoint de destino do PostgreSQL com tabelas do Babelfish

As limitações a seguir se aplicam ao utilizar um endpoint de destino do PostgreSQL com tabelas do Babelfish:

- Para o modo Preparação da tabela de destino, use somente os modos Não fazer nada ou Truncar. Não utilize o modo Abandonar tabelas no destino. Nesse modo, o DMS cria as tabelas como tabelas do PostgreSQL que o T-SQL talvez não reconheça.
- AWS DMS não suporta o tipo de dados sql_variant.

- O Babelfish não é compatível com os tipos de dados `HEIRARCHYID`, `GEOMETRY` e `GEOGRAPHY`. Para migrar esses tipos de dados, é possível adicionar regras de transformação para converter o tipo de dados em `wstring(250)`.
- O Babelfish é compatível com a migração de tipos de dados `BINARY`, `VARBINARY` e `IMAGE` utilizando o tipo de dados `BYTEA`. Em versões anteriores do Aurora PostgreSQL, é possível utilizar o DMS para migrar essas tabelas para um Endpoint de destino do Babelfish. Não é necessário especificar um tamanho para o tipo de dados `BYTEA`, conforme mostrado no exemplo a seguir.

  ```
  [Picture] [VARBINARY](max) NULL
  ```

  Altere o tipo de dados T-SQL anterior para o tipo de dados T-SQL compatível `BYTEA`.

  ```
  [Picture] BYTEA NULL
  ```

- Para versões anteriores do Aurora PostgreSQL Babelfish, se você criar uma tarefa de migração para replicação contínua do SQL Server para o Babelfish utilizando endpoint de destino do PostgreSQL, precisará atribuir o tipo de dados `SERIAL` a qualquer tabela que utilize colunas `IDENTITY`. Desde o Aurora PostgreSQL (versão 15.3/14.8 e superior) e do Babelfish (versão 3.2.0 e superior), a coluna de identidade é compatível e não é mais necessário atribuir o tipo de dados SERIAL. Para obter mais informações, consulte Utilizar SERIAL na seção Sequências e identidade do Manual de migração do SQL Server para o Aurora PostgreSQL. Crie a tabela no Babelfish, altere a definição da coluna da seguinte forma.

  ```
      [IDCol] [INT] IDENTITY(1,1) NOT NULL PRIMARY KEY
  ```

  Altere a anterior para a seguinte.

  ```
      [IDCol] SERIAL PRIMARY KEY
  ```

  O Aurora PostgreSQL compatível com o Babelfish cria uma sequência utilizando a configuração padrão e adiciona uma restrição `NOT NULL` à coluna. A sequência recém-criada se comporta como uma sequência normal (incrementada em 1) e não tem a opção de `SERIAL` composta.

- Depois de migrar dados com tabelas que utilizam colunas `IDENTITY` ou o tipo de dados `SERIAL`, redefina o objeto de sequência baseado em PostgreSQL com base no valor máximo da coluna. Depois de executar uma carga máxima das tabelas, utilize a consulta T-SQL a seguir para gerar instruções para definir o seed do objeto de sequência associado.

```
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(%';
```

A consulta gera uma série de instruções SELECT que você executa para atualizar os valores máximos de IDENTITY e SERIAL.

- Em versões do Babelfish anteriores à 3.2, o Modo LOB completo pode resultar em um erro de tabela. Se isso acontecer, crie uma tarefa separada para as tabelas que falharam no carregamento. Utilize o Modo LOB limitado para especificar o valor apropriado para o Tamanho máximo de LOB (KB). Outra opção é definir a configuração do atributo de conexão `ForceFullLob=True` do endpoint do SQL Server.
- Para versões do Babelfish anteriores à 3.2, a execução da validação de dados com tabelas do Babelfish que não utilizam chaves primárias baseadas em números inteiros gera uma mensagem de que uma chave exclusiva adequada não pode ser encontrada. Desde o Aurora PostgreSQL (versão 15.3/14.8 e superior) e do Babelfish (versão 3.2.0 e superior), a validação de dados para chaves primárias de número não inteiro é compatível.
- Devido às diferenças de precisão no número de casas decimais de segundos, o DMS relata falhas na validação de dados para tabelas do Babelfish que utilizam tipos de dados `DATETIME`. Para suprimir essas falhas, é possível adicionar o seguinte tipo de regra de validação para os tipos de dados `DATETIME`.

```
{
         "rule-type": "validation",
         "rule-id": "3",
         "rule-name": "3",
         "rule-target": "column",
         "object-locator": {
             "schema-name": "dbo",
             "table-name": "%",
             "column-name": "%",
             "data-type": "datetime"
         },
         "rule-action": "override-validation-function",
         "source-function": "case when ${column-name} is NULL then NULL else 0 end",
         "target-function": "case when ${column-name} is NULL then NULL else 0 end"
     }
```

Utilizar um banco de dados compatível com MySQL como destino do AWS Database Migration Service

Você pode migrar dados para qualquer banco de dados compatível com MySQL usando AWS DMS, de qualquer um dos mecanismos de dados de origem compatíveis. AWS DMS Se você estiver migrando para um banco de dados local compatível com MySQL, é AWS DMS necessário que seu mecanismo de origem resida no ecossistema. AWS O mecanismo pode estar em um serviço AWS gerenciado, como Amazon RDS, Amazon Aurora ou Amazon S3. Ou o mecanismo pode estar em um banco de dados autogerenciado no Amazon EC2.

Você pode usar o SSL para criptografar conexões entre o endpoint compatível com MySQL e a instância de replicação. Para obter mais informações sobre o uso do SSL com um endpoint compatível com MySQL, consulte Usando SSL com AWS Database Migration Service.

Para obter informações sobre as versões do MySQL que oferecem AWS DMS suporte como destino, consulte. Metas para AWS DMS

Você pode usar os seguintes bancos de dados compatíveis com MySQL como destinos para: 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**
>
> Independentemente do mecanismo de armazenamento de origem (MyISAM, MEMÓRIA, etc.), o AWS DMS cria uma tabela de destino compatível com MySQL como InnoDB por padrão.
>
> Se precisar de uma tabela em um mecanismo de armazenamento diferente do InnoDB, será possível criar manualmente a tabela no destino compatível com MySQL e migrá-la utilizando a opção Não fazer nada. Para ter mais informações, consulte Configurações de tarefa de carregamento completo.

Para obter mais detalhes sobre como trabalhar com bancos de dados compatíveis com MySQL como destino para o AWS DMS, consulte as seguintes seções.

Tópicos

- Usando qualquer banco de dados compatível com MySQL como alvo para AWS Database Migration Service
- Limitações no uso de um banco de dados compatível com MySQL como alvo para AWS Database Migration Service
- Configurações de endpoint ao usar um banco de dados compatível com MySQL como destino para AWS DMS
- Tipos de dados de destino do MySQL

Usando qualquer banco de dados compatível com MySQL como alvo para AWS Database Migration Service

Antes de começar a trabalhar com um banco de dados compatível com MySQL como destino do AWS DMS, confirme se você concluiu os seguintes pré-requisitos:

- Forneça uma conta de usuário AWS DMS que tenha privilégios de leitura/gravação no banco de dados compatível com MySQL. Para criar os privilégios necessários, execute os seguintes comandos.

```
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>'@'%';
```

- Durante a fase de migração de carga máxima, você precisa desativar as chaves externas nas suas tabelas de destino. Para desativar as verificações de chave estrangeira em um banco de dados compatível com MySQL durante um carregamento completo, você pode adicionar o seguinte comando à seção Atributos de conexão extra do AWS DMS console do seu endpoint de destino.

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

- Defina o parâmetro `local_infile = 1` do banco de dados para permitir que o AWS DMS carregue dados no banco de dados de destino.

Limitações no uso de um banco de dados compatível com MySQL como alvo para AWS Database Migration Service

Ao usar um banco de dados MySQL como destino, AWS DMS não oferece suporte ao seguinte:

- As instruções da linguagem de definição de dados (DDL) TRUNCATE PARTITION, DROP TABLE e RENAME TABLE.
- Uso de uma declaração `ALTER TABLE` *`table_name`* `ADD COLUMN` *`column_name`* para adicionar colunas ao início ou meio de uma tabela.
- Ao carregar dados em um destino compatível com MySQL em uma tarefa de carregamento total, AWS DMS não relata erros causados por restrições nos registros de tarefas, o que pode causar erros de chave duplicados ou incompatibilidades com o número de registros. Isso é causado pela forma como o MySQL trata dados locais com o comando `LOAD DATA`. Faça o seguinte durante a fase de carga máxima:
  - Desativar restrições
  - Use a AWS DMS validação para garantir que os dados sejam consistentes.

- Quando você atualiza o valor de uma coluna para seu valor existente, os bancos de dados compatíveis com MySQL retornam um aviso `0 rows affected`. Embora esse comportamento não seja tecnicamente um erro, ele é diferente de como a situação é controlada por outros mecanismos de banco de dados. Por exemplo, o Oracle executa uma atualização de uma linha. Para bancos de dados compatíveis com MySQL, AWS DMS gera uma entrada na tabela de controle awsdms_apply_exceptions e registra o seguinte aviso.

```
Some changes from the source database had no impact when applied to
the target database. See awsdms_apply_exceptions table for details.
```

- O Aurora Sem Servidor está disponível como destino para o Amazon Aurora versão 2, compatível com o MySQL versão 5.7. (Selecione o Aurora Sem Servidor versão 2.07.1 para poder utilizar o Aurora Sem Servidor compatível com o MySQL 5.7.) Para obter mais informações sobre o Aurora Serverless, consulte Como usar o Aurora Serverless v2 no Guia do usuário do Amazon Aurora.
- AWS DMS não suporta o uso de um endpoint de leitura para o Aurora ou o Amazon RDS, a menos que as instâncias estejam no modo gravável, ou seja, `read_only` os parâmetros `innodb_read_only` e estejam definidos como ou. `0` `OFF` Para obter mais informações sobre como utilizar o Amazon RDS e o Aurora como destinos, consulte:
  - Como determinar a qual instância de banco de dados você está conectado
  - Atualizar réplicas de leitura com o MySQL

Configurações de endpoint ao usar um banco de dados compatível com MySQL como destino para AWS DMS

É possível utilizar as configurações do endpoint para configurar o destino compatível com o MySQL de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino usando o AWS DMS console ou usando o `create-endpoint` comando no AWS CLI, com a sintaxe `--my-sql-settings '{"EndpointSetting": "value", ...}'` JSON.

A tabela a seguir mostra as configurações de endpoints que é possível utilizar com o MySQL como destino.

Nome	Descrição
`TargetDbType`	Especifica o destino para onde devem migrar as tabelas de origem, seja para um único banco de dados ou vários. Se você especificar`SPECIFIC_DATABASE` , precisará especificar o nome do banco de dados, AWS CLI seja ao usar AWS Management Console o. Valor padrão: `MULTIPLE_DATABASES` Valores válidos: {`SPECIFIC_DATABASE` , `MULTIPLE_DATABASES` } Exemplo: `--my-sql-settings '{"TargetDbType": "MULTIPLE_DATABASES"}'`
`ParallelLoadThreads`	Melhora o desempenho do carregamento de dados no banco de dados de destino compatível com MySQL. Especifica quantos threads devem ser usados para carregar dados no banco de dados de destino compatível com MySQL. Configurar um grande número de threads pode ter um efeito adverso no desempenho do banco de dados, pois cada thread requer uma conexão separada. Valor padrão: 1 Valores válidos: 1 a 5 Exemplo: `--my-sql-settings '{"ParallelLoadThreads": 1}'`
`AfterConnectScript`	Especifica um script para ser executado imediatamente após a conexão do AWS DMS com o endpoint. Por exemplo, é possível especificar que o destino compatível com MySQL deve converter as instruções recebidas no conjunto de caracteres latin1, que é o padrão compilado no conjunto de caracteres do

Nome	Descrição
	banco de dados. Esse parâmetro geralmente melhora o desempenho ao converter de clientes UTF8. Exemplo: `--my-sql-settings '{"AfterConnectScript": "SET character_set_connection='latin1'"}'`
`MaxFileSize`	Especifica o tamanho máximo (em KB) de um arquivo .csv utilizado para transferir dados para um banco de dados compatível com o MySQL. Valor padrão: 32768 KB (32 MB) Valores válidos: 1 a 1.048.576 `--my-sql-settings '{"MaxFileSize": 512}'`
`CleanSrcMetadataOnMismatch`	Limpa e recria as informações dos metadados da tabela na instância de replicação quando ocorre uma incompatibilidade. Por exemplo, em uma situação em que a execução de uma instrução de DDL alternativo em uma tabela pode resultar em informações diferentes sobre a tabela armazenada em cache na instância de replicação. Booliano. Valor padrão: `false` Exemplo: `--my-sql-settings '{"CleanSrcMetadataOnMismatch": false}'`

Também é possível utilizar atributos de conexão adicionais para configurar o banco de dados de destino compatível com MySQL.

A tabela a seguir mostra os atributos de conexão adicionais que podem ser utilizados com o MySQL como origem.

Nome	Descrição
`Initstmt=SET FOREIGN_KEY_CHECKS=0;`	Desabilita as verificações de chaves estrangeiras. Exemplo: `--extra-connection-attributes "Initstmt=SET FOREIGN_KEY_CHECKS=0;"`
`Initstmt=SET time_zone`	Especifica o fuso horário para o banco de dados de destino compatível com MySQL. Valor padrão: UTC Valores válidos: os nomes dos fusos horários disponíveis no banco de dados MySQL de destino. Exemplo: `--extra-connection-attributes "Initstmt=SET time_zone= US/Pacific ;"`

Como alternativa, é possível utilizar o parâmetro `AfterConnectScript` do comando `--my-sql-settings` para desativar as verificações de chave estrangeira e especificar o fuso horário do banco de dados.

Tipos de dados de destino do MySQL

A tabela a seguir mostra os tipos de dados de destino do banco de dados MySQL que são suportados durante o uso AWS DMS e o mapeamento padrão dos tipos de AWS DMS dados.

Para obter informações adicionais sobre AWS DMS os tipos de dados, consulteTipos de dados do AWS Database Migration Service.

AWS DMS tipos de dados	Tipos de dados do MySQL
BOOLEAN	BOOLEAN
BYTES	Se o tamanho for de 1 a 65.535, utilize VARBINARY (tamanho). Se o tamanho for de 65.536 a 2.147.483.647, utilize LONGLOB.

AWS DMS tipos de dados	Tipos de dados do MySQL
DATA	DATA
TIME	TIME
TIMESTAMP	“Se a escala for => 0 e =< 6, use: DATETIME (escala) Se a escala for => 7 e =< 9, use: VARCHAR (37)”
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	DECIMAL (p,s)
REAL4	FLOAT
REAL8	DOUBLE PRECISION
STRING	Se o tamanho for de 1 a 21.845, utilize VARCHAR (tamanho). Se o tamanho for de 21.846 a 2.147.483.647, utilize LONGTEXT.
UINT1	UNSIGNED TINYINT
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT

AWS DMS tipos de dados	Tipos de dados do MySQL
WSTRING	Se o tamanho for de 1 a 32.767, utilize VARCHAR (tamanho). Se o tamanho for de 32.768 a 2.147.483.647, utilize LONGTEXT.
BLOB	Se o tamanho for de 1 a 65.535, utilize BLOB. Se o tamanho for de 65.536 a 2.147.483.647, utilize LONGBLOB. Se o tamanho for 0, utilize LONGBLOB (suporte pleno ao tipo LOB).
NCLOB	Se o tamanho for de 1 a 65.535, use TEXT. Se o tamanho for de 65.536 a 2.147.483 .647, utilize LONGTEXT com ucs2 para CHARACTER SET. Se o tamanho for 0, utilize LONGTEXT (suporte pleno ao tipo LOB) e ucs2 para CHARACTER SET.
CLOB	Se o tamanho for de 1 a 65.535, use TEXT. Se o tamanho for de 65.536 a 2147483647, utilize LONGTEXT. Se o tamanho for 0, utilize LONGTEXT (suporte pleno ao tipo LOB).

Utilizar um banco de dados Amazon Redshift como destino do AWS Database Migration Service

É possível migrar dados para bancos de dados Amazon Redshift utilizando o AWS Database Migration Service. O Amazon Redshift é um serviço de data warehouse totalmente gerenciado e

em escala de petabytes na nuvem do . Com um banco de dados Amazon Redshift como destino, é possível migrar dados de todos os outros bancos de dados de origem compatíveis.

É possível utilizar o Amazon Redshift sem servidor como destino do AWS DMS. Para obter mais informações, consulte Utilizar o AWS DMS com o Amazon Redshift sem servidor como destino a seguir.

O cluster do Amazon Redshift deve estar na mesma conta da AWS e região da AWS que a instância de replicação.

Durante a migração de banco de dados para o Amazon Redshift, o AWS DMS primeiro move os dados para um bucket do Amazon S3. Quando os arquivos residem em um bucket do Amazon S3, o AWS DMS os transfere para as tabelas apropriadas no data warehouse do Amazon Redshift. O AWS cria o bucket do S3 na mesma região da AWS DMS que o banco de dados Amazon Redshift. A instância de replicação do AWS DMS deve estar na mesma região da AWS.

Se você utilizar a AWS CLI ou a API do DMS para migrar dados para o Amazon Redshift, configure um perfil do AWS Identity and Access Management (IAM) para permitir acesso ao S3. Para obter mais informações sobre a criação do perfil do IAM, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

O endpoint do Amazon Redshift fornece automação completa para:

- Geração de schema e mapeamento de tipo de dados
- Carregamento completo de tabelas de banco de dados de origem
- Carregamento incremental de alterações feitas a tabelas de origem
- Aplicação de alterações de schema em Data Definition Language (DDL - Linguagem de definição de dados) feitas a tabelas de origem
- Sincronização entre processos de carregamento completo e de captura de dados de alteração (CDC).

O AWS Database Migration Service oferece suporte a operações de carregamento completo e de processamento de alterações. O AWS DMS lê os dados no banco de dados de origem e cria uma série de arquivos de valores separados por vírgula (.csv). Para operações de carga máxima, o AWS DMS cria arquivos para cada tabela. O AWS DMS copia os arquivos de cada tabela em uma pasta separada no Amazon S3. Quando os arquivos são carregados no Amazon S3, o AWS DMS envia um comando copy, e os dados nos arquivos são copiados no Amazon Redshift. Para operações de

processamento de alterações, o AWS DMS copia as alterações líquidas nos arquivos .csv. O AWS DMS faz upload dos arquivos de alterações líquidas no Amazon S3 e copia os dados no Amazon Redshift.

Para obter mais detalhes sobre como trabalhar com o Amazon Redshift como destino para o AWS DMS, consulte as seguintes seções:

Tópicos

Pré-requisitos para a utilização de um banco de dados Amazon Redshift como destino do AWS Database Migration Service.

A lista a seguir descreve os pré-requisitos necessários para trabalhar com o Amazon Redshift como destino para a migração de dados:

- Utilize o console de gerenciamento da AWS para iniciar um cluster do Amazon Redshift. Observe as informações básicas sobre a sua conta da AWS e o cluster do Amazon Redshift, como a senha, o nome do usuário e o nome do banco de dados. Esses valores são necessários para criar o endpoint de destino do Amazon Redshift.

- O cluster do Amazon Redshift deve estar na mesma conta da AWS e na mesma região da AWS que a instância de replicação.
- A instância de replicação do AWS DMS precisa de conectividade de rede com o endpoint do Amazon Redshift (nome do host e porta) que o cluster utiliza.
- O AWS DMS utiliza um bucket do Amazon S3 para transferir dados para o banco de dados do Amazon Redshift. Para o AWS DMS criar um bucket, o console usa um perfil do IAM, `dms-access-for-endpoint`. Se você utilizar a AWS CLI ou a API do DMS para criar dados com o Amazon Redshift como o banco de dados de destino, é necessário criar esse perfil do IAM. Para obter mais informações sobre a criação do perfil, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.
- O AWS DMS converte BLOBs, CLOBs e NCLOBs para VARCHAR na instância do Amazon Redshift de destino. O Amazon Redshift não oferece suporte a tipos de dados VARCHAR maiores que 64 KB. Portanto, você não pode armazenar LOBs tradicionais no Amazon Redshift.
- Defina a tarefa de metadados de destino configurando BatchApplyEnabled como `true` para que o AWS DMS manipule as alterações nas tabelas de destino do Amazon Redshift durante a CDC. É necessária uma chave primária na tabela de origem e de destino. Sem uma chave primária, as alterações são aplicadas instrução por instrução. E isso pode afetar negativamente o desempenho da tarefa durante a CDC, casando latência de destino e afetando a fila de confirmação do cluster.

Privilégios necessários para utilizar o Redshift como destino

Utilize o comando GRANT para definir os privilégios de acesso do usuário ou do grupo de usuários. Os privilégios incluem opções de acesso como leitura de dados em tabelas e exibições, gravação de dados e criação de tabelas. Para obter mais informações sobre como utilizar GRANT com o Amazon Redshift, consulte GRANT no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Veja a seguir a sintaxe para fornecer privilégios específicos para uma tabela, banco de dados, esquema, perfil, procedimento ou privilégios em nível de linguagem em tabelas e visualizações do Amazon Redshift.

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | REFERENCES } [,...] | ALL
 [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...] | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { { CREATE | TEMPORARY | TEMP } [,...] | ALL [ PRIVILEGES ] }
    ON DATABASE db_name [, ...]
```

```
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { { CREATE | USAGE } [,...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schema_name [, ...]
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON { FUNCTION function_name ( [ [ argname ] argtype [, ...] ] ) [, ...] | ALL
 FUNCTIONS IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON { PROCEDURE procedure_name ( [ [ argname ] argtype [, ...] ] ) [, ...] | ALL
 PROCEDURES IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]

GRANT USAGE
    ON LANGUAGE language_name [, ...]
    TO { username [ WITH GRANT OPTION ] | GROUP group_name | PUBLIC } [, ...]
```

A seguir está a sintaxe para privilégios de nível de coluna em tabelas e visualizações do Amazon Redshift.

```
GRANT { { SELECT | UPDATE } ( column_name [, ...] ) [, ...] | ALL [ PRIVILEGES ]
 ( column_name [,...] ) }
     ON { [ TABLE ] table_name [, ...] }
     TO { username | GROUP group_name | PUBLIC } [, ...]
```

A seguir está a sintaxe para o privilégio ASSUMEROLE concedido a usuários e grupos com um perfil especificado.

```
GRANT ASSUMEROLE
    ON { 'iam_role' [, ...] | ALL }
    TO { username | GROUP group_name | PUBLIC } [, ...]
    FOR { ALL | COPY | UNLOAD } [, ...]
```

Limitações ao utilizar o Amazon Redshift como destino do AWS Database Migration Service

As seguintes limitações se aplicam ao utilizar um banco de dados Amazon Redshift como destino:

- Não ative o versionamento para o bucket do S3 utilizado como armazenamento intermediário para o destino do Amazon Redshift. Se o versionamento do S3 for necessário, utilize políticas de ciclo de vida para excluir ativamente as versões antigas. Caso contrário, é possível encontrar falhas na conexão de teste de endpoint devido ao tempo limite de uma chamada `list-object` do S3. Para criar uma política de ciclo de vida para um bucket do S3, consulte Gerenciar o ciclo de vida do armazenamento. Para excluir a versão de um objeto do S3, consulte Excluir versões de objetos de um bucket com versionamento ativado.
- O DDL a seguir não tem suporte:

```
ALTER TABLE table name MODIFY COLUMN column name data type;
```

- O AWS DMS não pode migrar ou replicar alterações para um esquema com um nome iniciado com sublinhado (_). Se você tiver esquemas com um nome que começa com um sublinhado, utilize transformações de mapeamento para renomear o esquema no destino.
- O Amazon Redshift não oferece suporte a VARCHARs maiores que 64 KB. LOBs de bancos de dados tradicionais não podem ser armazenados no Amazon Redshift.
- A aplicação de uma instrução DELETE a uma tabela com uma chave primária de várias colunas não tem suporte quando qualquer um dos nomes de coluna da chave primária usa uma palavra reservada. Acesse aqui para ver uma lista de palavras reservadas do Amazon Redshift.
- Poderá haver problemas de desempenho se o sistema de origem executar operações UPDATE na chave primária de uma tabela de origem. Esses problemas de desempenho ocorrem ao aplicar alterações no destino. Isso ocorre porque as operações UPDATE (e DELETE) dependem do valor da chave primária para identificar a linha de destino. Se você atualizar a chave primária de uma tabela de origem, o log de tarefas conterá mensagens, como as seguintes:

```
Update on table 1 changes PK to a PK that was previously updated in the same bulk
 update.
```

- O DMS não é compatível com nomes DNS personalizados ao configurar um endpoint para um cluster do Redshift, e você precisa utilizar o nome DNS fornecido pela Amazon. Como o cluster do Amazon Redshift deve estar na mesma conta e região da AWS da instância de replicação, a validação falhará se você utilizar um endpoint de DNS personalizado.
- O Amazon Redshift tem um tempo limite padrão de sessão ociosa de 4 horas. Quando não há nenhuma atividade na tarefa de replicação do DMS, o Redshift desconecta a sessão após 4 horas. Os erros podem resultar da incapacidade do DMS de se conectar e da possível necessidade de

reinicialização. Como solução alternativa, defina um limite de SESSION TIMEOUT maior que 4 horas para o usuário de replicação do DMS. Ou consulte a descrição de ALTER USER no Guia do desenvolvedor do banco de dados Amazon Redshift.

- Quando o AWS DMS replica os dados da tabela de origem sem uma chave primária ou exclusiva, a latência da CDC pode ser alta, resultando em um nível de desempenho inaceitável.

Configurar um banco de dados Amazon Redshift como destino do AWS Database Migration Service

O AWS Database Migration Service deve estar configurado para trabalhar com a instância do Amazon Redshift. A tabela a seguir descreve as propriedades de configuração disponíveis para o endpoint do Amazon Redshift.

Propriedade	Descrição
servidor	O nome do cluster do Amazon Redshift que você está utilizando.
porta	O número da porta do Amazon Redshift. O valor padrão é 5439.
username	Um nome de usuário do Amazon Redshift de um usuário registrado.
password	A senha do usuário nomeado na propriedade username.
banco de dados	O nome do data warehouse (serviço) do Amazon Redshift com o qual você está trabalhando.

Se quiser adicionar atributos de string de conexão adicional ao endpoint do Amazon Redshift, especifique os atributos `maxFileSize` e `fileTransferUploadStreams`. Para obter mais informações sobre esses atributos, consulte Configurações de endpoint ao utilizar o Amazon Redshift como destino do AWS DMS.

Utilizar o roteamento aprimorado da VPC com o Amazon Redshift como destino do AWS Database Migration Service

Se você utilizar o roteamento aprimorado da VPC com o destino do Amazon Redshift, todo o tráfego de COPY entre o cluster do Amazon Redshift e os repositórios de dados trafegarão pela VPC. Como

o roteamento aprimorado da VPC afeta a maneira como o Amazon Redshift acessa outros recursos, os comandos COPY poderão falhar se você não configurar a VPC corretamente.

O AWS DMS pode ser afetado por esse comportamento, já que utiliza o comando COPY para mover dados no S3 para um cluster do Amazon Redshift.

Veja a seguir as etapas que o AWS DMS realiza para carregar dados em um destino do Amazon Redshift:

1. O AWS DMS copia dados da origem para arquivos .csv no servidor de replicação.
2. O AWS DMS utiliza o SDK da AWS para copiar arquivos .csv em um bucket do S3 na conta.
3. O AWS DMS utiliza o comando COPY no Amazon Redshift para copiar dados dos arquivos .csv no S3 em uma tabela apropriada no Amazon Redshift.

Se o roteamento aprimorado da VPC não estiver ativado, o Amazon Redshift roteará o tráfego pela internet, incluindo o tráfego para outros serviços na rede da AWS. Se o recurso não estiver habilitado, não será necessário configurar o caminho de rede. Se o recurso estiver habilitado, você deverá criar especificamente um caminho de rede entre o cluster do VPC e os recursos de dados. Para obter mais informações sobre a configuração necessária, consulte Roteamento aprimorado da VPC na documentação do Amazon Redshift.

Criar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift

É possível criptografar os dados de destino enviados ao Amazon S3 antes que sejam copiados no Amazon Redshift. Para isso, é possível criar e utilizar as chaves personalizadas do AWS KMS. É possível utilizar a chave criada para criptografar os dados de destino utilizando um dos seguintes mecanismos ao criar o endpoint de destino do Amazon Redshift:

- Utilize a opção a seguir ao executar o comando `create-endpoint` utilizando a AWS CLI.

```
--redshift-settings '{"EncryptionMode": "SSE_KMS", "ServerSideEncryptionKmsKeyId":
 "your-kms-key-ARN"}'
```

  Aqui, *`your-kms-key-ARN`* é o nome de recurso da Amazon (ARN) de sua chave do KMS. Para obter mais informações, consulte Utilizar uma chave de criptografia de dados e um bucket do Amazon S3 como armazenamento intermediário.

- Defina o atributo de conexão adicional `encryptionMode` como o valor `SSE_KMS`, e o atributo de conexão adicional `serverSideEncryptionKmsKeyId` como o ARN de sua chave do KMS. Para obter mais informações, consulte Configurações de endpoint ao utilizar o Amazon Redshift como destino do AWS DMS.

Para criptografar os dados de destino do Amazon Redshift utilizando uma chave do KMS, você precisa de um perfil do AWS Identity and Access Management (IAM) que tenha permissões para acessar os dados do Amazon Redshift. Esse perfil do IAM é acessado em uma política (uma política de chaves) anexada à chave de criptografia criada. É possível fazer isso no console do IAM criando o seguinte:

- Um perfil do IAM com uma política gerenciada pela AWS.
- A chave do KMS com uma política de chaves que faz referência a esse perfil.

Os procedimentos a seguir descrevem como fazer isso.

Como criar um perfil do IAM com a política gerenciada pela AWS necessária

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Funções. A página Roles (Funções) é aberta.
3. Selecione Criar perfil. A página Create role (Criar função) é aberta.
4. Com o Serviço da AWS escolhido como uma entidade confiável, escolha DMS como o serviço que utilizará esse perfil.
5. Escolha Próximo: permissões. A página Attach permissions policies (Anexar políticas de permissões) é exibida.
6. Encontre e selecione a política `AmazonDMSRedshiftS3Role`.
7. Escolha Próximo: etiquetas. A página Adicionar tags é exibida. Aqui, você pode adicionar todas as tags desejadas.
8. Escolha Next: Review (Próximo: revisar) e reveja os resultados.
9. Se as configurações forem o que você precisa, insira um nome para o perfil (por exemplo, `DMS-Redshift-endpoint-access-role`), e qualquer descrição adicional e escolha Criar função. A página Roles (Funções) é aberta com uma mensagem indicando que o perfil foi criado.

Agora, você criou o perfil para acessar os recursos do Amazon Redshift para criptografia com um nome especificado, por exemplo `DMS-Redshift-endpoint-access-role`.

Como criar uma chave de criptografia do AWS KMS com uma política de chave que faz referência ao seu perfil do IAM

> **Note**
>
> Para obter mais informações sobre como AWS DMS funciona com chaves de criptografia do AWS KMS, consulte Configurando uma chave de criptografia e especificando permissões AWS KMS.

1. Faça login no AWS Management Console e abra o console do AWS Key Management Service (AWS KMS) em https://console.aws.amazon.com/kms.
2. Para alterar a Região da AWS, use o Region selector (Seletor de regiões) no canto superior direito da página.
3. No painel de navegação, escolha Customer managed keys (Chaves gerenciadas pelo cliente).
4. Escolha Create key (Criar chave). A página Configure key (Configurar chave) é aberta.
5. Para Key type (Tipo de chave), escolha Symmetric (Simétrica).

   > **Note**
   >
   > Ao criar essa chave, você só pode criar uma chave simétrica, pois todos os serviços da AWS, como o Amazon Redshift, funcionam somente com chaves de criptografia simétricas.

6. Escolha Opções avançadas. Para Key material origin (Origem do material da chave), certifique-se de que o KMS está escolhido e escolha Next (Próximo). A página Add labels (Adicionar rótulos) é aberta.
7. Em Create alias and description (Criar alias e descrição), insira um alias para a chave (por exemplo, `DMS-Redshift-endpoint-encryption-key`) e qualquer descrição adicional.
8. Em Tags, adicione todas as tags desejadas para ajudar a identificar a chave e controlar seu uso e escolha Next (Próximo). A página Define key administrative permissions (Definir permissões administrativas de chaves) é aberta mostrando uma lista de usuários e funções que podem ser escolhidos.
9. Adicione os usuários e as funções desejados para gerenciar a chave. Certifique-se de que esses usuários e funções tenham as permissões necessárias para gerenciar a chave.

10. Em Key deletion (Exclusão de chaves), escolha se os administradores de chaves podem excluir a chave e escolha Next (Próximo). A página Define key usage permissions (Definir permissões de uso de chaves) é aberta mostrando uma lista adicional de usuários e funções que podem ser escolhidos.
11. Para Esta conta, escolha os usuários disponíveis que deseja que executem operações criptográficas em destinos do Amazon Redshift. Escolha também o perfil criado anteriormente em Perfis para ativar o acesso à criptografia dos objetos de destino do Amazon Redshift, por exemplo, `DMS-Redshift-endpoint-access-role`).
12. Se quiser adicionar outras contas não listadas para ter esse mesmo acesso, em Outras contas da AWS, escolha Adicionar outra conta da AWS e escolha Próximo. A página Review and edit key policy (Rever e editar política de chave) é aberta mostrando o JSON da política de chave que você pode revisar e editar digitando no JSON existente. Aqui, a política de chave que faz referência à função e aos usuários é mostrada (por exemplo, `Admin` e `User1`) que você escolheu na etapa anterior. Você também pode ver as diferentes ações de chaves permitidas para as várias entidades principais (usuários e perfis), conforme mostrado no exemplo a seguir.

```
{
  "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. Escolha Terminar. A página Chaves de criptografia é aberta com uma mensagem indicando que a AWS KMS key foi criada.

Agora, você criou uma nova chave do KMS com um alias especificado (por exemplo, `DMS-Redshift-endpoint-encryption-key`). Essa chave permite que o AWS DMS criptografe os dados de destino do Amazon Redshift.

Configurações de endpoint ao utilizar o Amazon Redshift como destino do AWS DMS

É possível utilizar as configurações de endpoint para configurar o destino do Amazon Redshift de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--redshift-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o Amazon Redshift como destino.

Nome	Descrição
`MaxFileSize`	Especifica o tamanho máximo (em KB) de um arquivo .csv utilizado para transferir dados para o Amazon Redshift. Valor padrão: 32768 KB (32 MB)

Nome	Descrição
	Valores válidos: 1 a 1.048.576 Exemplo: `--redshift-settings '{"MaxFileSize": 512}'`
`FileTransferUploadStreams`	Especifica o número de threads utilizados para fazer upload de um único arquivo. Valor padrão: 10 Valores válidos: 1 a 64 Exemplo: `--redshift-settings '{"FileTransferUploadStreams": 20}'`
`Acceptanydate`	Especifica se qualquer formato de data será aceito incluindo formatos de data inválidos, como 0000-00-00. Valor booleano. Valor padrão: falso Valores válidos: true | false Exemplo: `--redshift-settings '{"Acceptanydate": true}'`

Nome	Descrição
`Dateformat`	Especifica o formato de data. Esta é uma sequência de entrada e, por padrão, está vazia. O formato padrão é AAAA-MM-DD, mas é possível alterá-lo para, por exemplo, DD-MM-AAAA. Se os valores de data ou hora utilizarem outros formatos, utilize o argumento `auto` com o parâmetro `Dateformat` . O argumento `auto` reconhece vários formatos não compatíveis ao utilizar uma string `Dateformat` . A palavra-chave `auto` diferencia maiúsculas de minúsculas. Valor padrão: vazio Valores válidos: "*`dateformat_string`* " ou `auto` Exemplo:`--redshift-settings '{"Datefo rmat": "auto"}'`
`Timeformat`	Especifica o formato de hora. Esta é uma sequência de entrada e, por padrão, está vazia. O argumento `auto` reconhece vários formatos que não são compatíveis com a utilização de uma string `Timeformat` Se os valores de data e hora utilizarem formatos diferentes um do outro, utilize o argumento `auto` com o parâmetro `Timeformat` . Valor padrão: 10 Valores válidos: "*`Timeformat_string`* " | "auto" | "epochsecs" | "epochmillisecs" Exemplo:`--redshift-settings '{"Timefo rmat": "auto"}'`

Nome	Descrição
`Emptyasnull`	Especifica se o AWS DMS deve migrar campos CHAR e VARCHAR vazios como nulos. Um valor de true define os campos CHAR e VARCHAR vazios como nulos. Valor padrão: falso Valores válidos: true | false Exemplo: `--redshift-settings '{"Emptyasnull": true}'`
`TruncateColumns`	Trunca dados em colunas no número apropriado de caracteres, de maneira que ele caiba na especificação da coluna. Aplica-se somente a colunas com um tipo de dados VARCHAR ou CHAR e linhas com 4 MB ou menos. Valor padrão: falso Valores válidos: true | false Exemplo: `--redshift-settings '{"TruncateColumns": true}'`
`RemoveQuotes`	Remove as aspas de strings nos dados recebidos. Todos os caracteres entre aspas, inclusive delimitadores, são mantidos. Para obter mais informações sobre como remover cotas de um destino do Amazon Redshift, consulte o Guia do desenvolvedor do Amazon Redshift. Valor padrão: falso Valores válidos: true | false Exemplo: `--redshift-settings '{"RemoveQuotes": true}'`

Nome	Descrição
`TrimBlanks`	Remove os caracteres de espaço em branco à direita de uma string VARCHAR. Esse parâmetro se aplica somente a colunas com um tipo de dados VARCHAR. Valor padrão: falso Valores válidos: true | false Exemplo: `--redshift-settings '{"TrimBlanks": true}'`

Nome	Descrição
`EncryptionMode`	Especifica o modo de criptografia do lado do servidor que você deseja utilizar para enviar os dados para o S3 antes que sejam copiados no Amazon Redshift. Os valores válidos são `SSE_S3` (criptografia no lado do servidor do S3) ou `SSE_KMS` (criptografia da chave do KMS). Se você escolher `SSE_KMS`, defina o parâmetro `ServerSideEncryptionKmsKeyId` como o Nome do recurso da Amazon (ARN) para a chave do KMS a ser usada para a criptografia. **Note** Você também pode utilizar o comando `modify-endpoint` da CLI para alterar o valor da configuração de `EncryptionMode` um endpoint existente de `SSE_KMS` para `SSE_S3`. Mas não é possível alterar o valor de `EncryptionMode` de `SSE_S3` para `SSE_KMS`. Valor padrão: `SSE_S3` Valores válidos: `SSE_S3` ou `SSE_KMS` Exemplo:`--redshift-settings '{"EncryptionMode": "SSE_S3"}'`

Nome	Descrição
`ServerSideEncryptionKmsKeyId`	Se você definir `EncryptionMode` como `SSE_KMS`, defina esse parâmetro como o ARN da chave do KMS. É possível encontrar esse ARN selecionando o alias da chave na lista de chaves do AWS KMS criada para a sua conta. Ao criar a chave, você deve associar políticas e perfis específicos a ela. Para obter mais informações, consulte Criar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift. Exemplo: `--redshift-settings '{"ServerSideEncryptionKmsKeyId":"arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'`
`EnableParallelBatchInMemoryCSVFiles`	A configuração de `EnableParallelBatchInMemoryCSVFiles` melhora o desempenho de tarefas de carga máxima maiores com vários threads, fazendo com que o DMS grave em disco em vez de na memória. O valor padrão é `false`.
`CompressCsvFiles`	Utilize esse atributo para compactar dados enviados para um destino do Amazon Redshift durante a migração. O valor padrão é `true`, e a compactação é ativada por padrão.

Utilizar uma chave de criptografia de dados e um bucket do Amazon S3 como armazenamento intermediário

É possível utilizar as configurações de endpoint do Amazon Redshift para configurar o seguinte:

- A chave de criptografia de dados personalizada do AWS KMS. É possível utilizar essa chave para criptografar os dados enviados ao Amazon S3 antes que sejam copiados no Amazon Redshift.
- Um bucket do S3 personalizado como armazenamento intermediário para dados migrados para o Amazon Redshift.

- Mapeie um booleano como booleano de uma origem do PostgreSQL. Por padrão, um tipo BOOLEAN é migrado como varchar(1). É possível especificar `MapBooleanAsBoolean` para permitir que o destino do Redshift migre o tipo booleano como booleano, conforme mostrado no exemplo a seguir.

  ```
  --redshift-settings '{"MapBooleanAsBoolean": true}'
  ```

  Observe que você deve definir essa configuração nos endpoints de origem e de destino para que ela tenha efeito.

Configurações de chave do KMS para criptografia de dados

Os exemplos a seguir mostram como configurar uma chave personalizada do KMS para criptografar os dados enviados por push ao S3. Para começar, é possível fazer a seguinte chamada de `create-endpoint` utilizando a AWS CLI.

```
aws dms create-endpoint --endpoint-identifier redshift-target-endpoint --endpoint-type
 target
--engine-name redshift --username your-username --password your-password
--server-name your-server-name --port 5439 --database-name your-db-name
--redshift-settings '{"EncryptionMode": "SSE_KMS",
"ServerSideEncryptionKmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/24c3c5a1-
f34a-4519-a85b-2debbef226d1"}'
```

Aqui, o objeto JSON especificado pela opção `--redshift-settings` define dois parâmetros. Um é um parâmetro `EncryptionMode` com o valor `SSE_KMS`. O outro é um parâmetro `ServerSideEncryptionKmsKeyId` com o valor `arn:aws:kms:us-east-1:111122223333:key/24c3c5a1-f34a-4519-a85b-2debbef226d1`. Esse valor é um Nome de recurso da Amazon (ARN) para a chave personalizada do KMS.

Por padrão, a criptografia dos dados do S3 ocorre usando a criptografia do lado do servidor do S3. Para o destino do Amazon Redshift do exemplo anterior, isso também é equivalente a especificar as configurações de endpoint, como no exemplo a seguir.

```
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"}'
```

Para obter mais informações sobre como trabalhar com a criptografia do lado do servidor do S3, consulte Proteger dados utilizando criptografia do lado do servidor no Guia do usuário do Amazon Simple Storage Service.

> **Note**
>
> Também é possível utilizar o comando `modify-endpoint` na CLI para alterar o valor do parâmetro `EncryptionMode` para um endpoint existente de `SSE_KMS` para `SSE_S3`. Mas não é possível alterar o valor de `EncryptionMode` de `SSE_S3` para `SSE_KMS`.

Configurações do bucket do Amazon S3

Ao migrar dados para um endpoint de destino do Amazon Redshift, o AWS DMS utiliza um bucket padrão do Amazon S3 como armazenamento intermediário de tarefa antes de copiar os dados migrados para o Amazon Redshift. Por exemplo, os exemplos mostrados para a criação de um endpoint de destino do Amazon Redshift com uma chave de criptografia de dados do AWS KMS utilizam esse bucket padrão do S3 (consulte Configurações de chave do KMS para criptografia de dados).

É possível, em vez disso, especificar um bucket do S3 personalizado para esse armazenamento intermediário incluindo os seguintes parâmetros no valor da opção `--redshift-settings` no comando `create-endpoint` da AWS CLI:

- `BucketName`: uma string que você especifica como o nome do armazenamento do bucket do S3. Se o perfil de acesso ao serviço for baseado na política `AmazonDMSRedshiftS3Role`, esse valor deverá ter um prefixo de `dms-`, por exemplo, `dms-my-bucket-name`.
- `BucketFolder`: (opcional) uma string que é possível especificar como o nome da pasta de armazenamento no bucket do S3 especificado.
- `ServiceAccessRoleArn`: o ARN de um perfil do IAM que permite acesso administrativo ao bucket do S3. Normalmente, você cria esse perfil com base na política `AmazonDMSRedshiftS3Role`. Para obter um exemplo, consulte o procedimento para criar um perfil do IAM com a política gerenciada pela AWS necessária em Criar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift.

> **Note**
>
> Se você especificar o ARN de outro perfil do IAM utilizando a opção `--service-access-role-arn` do comando `create-endpoint`, essa opção de perfil do IAM terá precedência.

O exemplo a seguir mostra como é possível utilizar esses parâmetros para especificar um bucket personalizado do Amazon S3 na seguinte chamada de `create-endpoint` utilizando a AWS CLI.

```
aws dms create-endpoint --endpoint-identifier redshift-target-endpoint --endpoint-type
 target
--engine-name redshift --username your-username --password your-password
--server-name your-server-name --port 5439 --database-name your-db-name
--redshift-settings '{"ServiceAccessRoleArn": "your-service-access-ARN",
"BucketName": "your-bucket-name", "BucketFolder": "your-bucket-folder-name"}'
```

Configurações de tarefas de vários threads para o Amazon Redshift

É possível melhorar o desempenho de tarefas de carga máxima e captura de dados de alteração (CDC) para um endpoint de destino do Amazon Redshift utilizando configurações de tarefas de vários threads. Elas permitem especificar os threads simultâneos e o número de registros a serem armazenados em um buffer.

Configurações de tarefas de vários threads para o Amazon Redshift

Para promover o desempenho da carga máxima, é possível utilizar as seguintes configurações da tarefa `ParallelLoad*`:

- `ParallelLoadThreads`: especifica o número de threads simultâneos que o DMS utiliza durante uma carga máxima para enviar registros de dados para um endpoint de destino do Amazon Redshift. O valor padrão é zero (0) e o valor máximo é 32. Para obter mais informações, consulte Configurações de tarefa de carregamento completo.

  É possível utilizar o atributo `enableParallelBatchInMemoryCSVFiles` definido como `false` ao utilizar a configuração da tarefa `ParallelLoadThreads`. O atributo melhora o desempenho de tarefas de carga máxima maiores com vários threads, fazendo com que o DMS grave em disco em vez de na memória. O valor padrão é `true`.

- `ParallelLoadBufferSize`: especifica o máximo de solicitações de registros de dados ao utilizar threads de carga paralela com o destino do Redshift. O valor padrão é 100 e o valor máximo é 1.000. É recomendável utilizar essa opção quando ParallelLoadThreads > 1 (maior que um).

> **Note**
>
> Compatibilidade com a utilização de configurações da tarefa `ParallelLoad*` durante FULL LOAD para endpoints de destino do Amazon Redshift está disponível nas versões 3.4.5 e superiores do AWS DMS.
> A configuração de `ReplaceInvalidChars` do endpoint do Redshift não é compatível para utilização durante a captura de dados de alteração (CDC) ou durante uma tarefa de migração FULL LOAD ativada para carga paralela. Ela é compatível com a migração FULL LOAD quando a carga paralela não está ativada. Para obter mais informações, consulte RedshiftSettings na Referência da API do AWS Database Migration Service.

Configurações de tarefas de CDC de vários threads para o Amazon Redshift

Para promover o desempenho da CDC, é possível utilizar as seguintes configurações da tarefa `ParallelApply*`:

- `ParallelApplyThreads`: especifica o número de threads simultâneos que o AWS DMS utiliza durante uma carga de CDC para enviar registros de dados para um endpoint de destino do Amazon Redshift. O valor padrão é zero (0) e o valor máximo é 32. O valor mínimo recomendado é igual ao número de fatias no cluster.
- `ParallelApplyBufferSize`: especifica o máximo de solicitações de registro de dados ao utilizar threads de aplicação paralelos com o destino do Redshift. O valor padrão é 100 e o valor máximo é 1.000. É recomendável utilizar essa opção quando ParallelApplyThreads > 1 (maior que um).

  Para obter o máximo benefício do Redshift como destino, é recomendável que o valor de `ParallelApplyBufferSize` seja pelo menos duas vezes (o dobro ou mais) do número de `ParallelApplyThreads`.

> **Note**
>
> Compatibilidade com a utilização de configurações da tarefa `ParallelApply*` durante a CDC para endpoints de destino do Amazon Redshift está disponível nas versões 3.4.3 e superiores do AWS DMS.

O nível de paralelismo aplicado depende da correlação entre o tamanho do lote e o tamanho máximo do arquivo utilizado para transferir dados. Ao utilizar configurações de tarefas CDC de vários threads com um destino do Redshift, os benefícios são obtidos quando o tamanho do lote é grande em relação ao tamanho máximo do arquivo. Por exemplo, é possível utilizar a seguinte combinação de configurações de endpoint e de tarefa para ajustar o desempenho ideal.

```
// Redshift endpoint setting

        MaxFileSize=250000;

// Task settings

        BatchApplyEnabled=true;
        BatchSplitSize =8000;
        BatchApplyTimeoutMax =1800;
        BatchApplyTimeoutMin =1800;
        ParallelApplyThreads=32;
        ParallelApplyBufferSize=100;
```

Utilizando as configurações do exemplo anterior, um cliente com uma workload transacional pesada se beneficia com seu buffer de lote, contendo 8.000 registros, sendo preenchido em 1800 segundos, utilizando 32 threads paralelos com um tamanho máximo de arquivo de 250 MB.

Para obter mais informações, consulte Configurações de ajuste de processamento de alterações.

> **Note**
>
> As consultas do DMS executadas durante a replicação contínua em um cluster do Redshift podem compartilhar a mesma fila do WLM (gerenciamento da workload) com outras consultas de aplicações em execução. Portanto, considere configurar adequadamente as propriedades do WLM para influenciar o desempenho durante a replicação contínua para um

destino do Redshift. Por exemplo, se outras consultas ETL paralelas estiverem em execução, o DMS será executado mais lentamente e os ganhos de desempenho serão perdidos.

Tipos de dados de destino do Amazon Redshift

O endpoint do Amazon Redshift para o AWS DMS é compatível com a maioria dos tipos de dados do Amazon Redshift. A tabela a seguir mostra os tipos de dados de destino compatíveis do Amazon Redshift ao utilizar o AWS DMS e o mapeamento padrão em tipos de dados do AWS DMS.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipos de dados do AWS DMS	Tipos de dados do Amazon Redshift
BOOLEAN	BOOL
BYTES	VARCHAR (tamanho)
DATA	DATA
TIME	VARCHAR(20)
DATETIME	Se a escala for => 0 e =< 6, dependendo do tipo de coluna de destino do Redshift, uma das seguintes opções: TIMESTAMP (s) TIMESTAMPTZ (s): se o timestamp de origem contiver um deslocamento de zona (como no SQL Server ou Oracle), ele será convertid o em UTC na inserção/atualização. Se ele não contiver um deslocamento, a hora já será considerada em UTC. Se a escala for => 7 e =< 9, use: VARCHAR (37)

Tipos de dados do AWS DMS	Tipos de dados do Amazon Redshift
INT1	INT2
INT2	INT2
INT4	INT4
INT8	INT8
NUMERIC	Se a escala for => 0 e =< 37, use: NUMERIC (p,s) Se a escala for => 38 e =< 127, use: VARCHAR (tamanho)
REAL4	FLOAT4
REAL8	FLOAT8
STRING	Se o tamanho for de 1 a 65.535, utilize VARCHAR (tamanho em bytes) Se o tamanho for de 65.536 a 2.147.483.647, utilize VARCHAR (65535)
UINT1	INT2
UINT2	INT2
UINT4	INT4
UINT8	NUMERIC (20,0)
WSTRING	Se o tamanho for de 1 a 65.535, utilize NVARCHAR (tamanho em bytes) Se o tamanho for de 65.536 a 2.147.483.647, utilize NVARCHAR (65535)

Tipos de dados do AWS DMS	Tipos de dados do Amazon Redshift
BLOB	VARCHAR (tamanho máximo de LOB *2) O tamanho máximo de LOB não pode exceder 31 KB. O Amazon Redshift não oferece suporte a VARCHARs maiores que 64 KB.
NCLOB	NVARCHAR (tamanho máximo de LOB) O tamanho máximo de LOB não pode exceder 63 KB. O Amazon Redshift não oferece suporte a VARCHARs maiores que 64 KB.
CLOB	VARCHAR (tamanho máximo de LOB) O tamanho máximo de LOB não pode exceder 63 KB. O Amazon Redshift não oferece suporte a VARCHARs maiores que 64 KB.

Utilizar o AWS DMS com o Amazon Redshift sem servidor como destino

O AWS DMS é compatível com a utilização do Amazon Redshift sem servidor como endpoint de destino. Para obter informações sobre como utilizar o Amazon Redshift sem servidor, consulte Amazon Redshift sem servidor no Guia de gerenciamento do Amazon Redshift.

Este tópico descreve como utilizar um endpoint do Amazon Redshift sem servidor com o AWS DMS.

> **Note**
>
> Ao criar um endpoint do Amazon Redshift sem servidor, para o campo DatabaseName da configuração de endpoint RedshiftSettings, utilize o nome do data warehouse do Amazon Redshift ou o nome do endpoint do grupo de trabalho. Para o campo ServerName, utilize o valor de endpoint exibido na página Grupo de trabalho do cluster com tecnologia sem servidor (por exemplo, `default-workgroup.093291321484.us-east-1.redshift-serverless.amazonaws.com`). Para obter informações sobre como criar um endpoint, consulte Criar endpoints de origem e de destino. Para obter informações sobre o endpoint do grupo de trabalho, consulte Conexão com o Amazon Redshift sem servidor.

Política de confiança com o Amazon Redshift sem servidor como destino

Ao utilizar o Amazon Redshift sem servidor como um endpoint de destino, adicione a seguinte seção destacada à política de confiança. Essa política de confiança é anexada ao perfil `dms-access-for-endpoint`.

```
{
    "PolicyVersion": {
        "CreateDate": "2016-05-23T16:29:57Z",
        "VersionId": "v3",
        "Document": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "ec2:CreateNetworkInterface",
                    "ec2:DescribeAvailabilityZones",
                    "ec2:DescribeInternetGateways",
                    "ec2:DescribeSecurityGroups",
                    "ec2:DescribeSubnets",
                    "ec2:DescribeVpcs",
                    "ec2:DeleteNetworkInterface",
                    "ec2:ModifyNetworkInterfaceAttribute"
                ],
                "Resource": "arn:aws:service:region:account:resourcetype/id",
                "Effect": "Allow"
            },
            {
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "redshift-serverless.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "IsDefaultVersion": true
    }
}
```

Para obter mais informações sobre como utilizar as políticas de confiança com o AWS DMS, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

Limitações ao utilizar o Amazon Redshift sem servidor como destino

A utilização do Redshift sem servidor como destino apresenta as seguintes limitações:

- O AWS DMS só é compatível com o Amazon Redshift sem servidor como endpoint em regiões compatíveis com o Amazon Redshift sem servidor. Para obter informações sobre quais regiões são compatíveis com o Amazon Redshift sem servidor, consulte API do Redshift sem servidor no tópico Endpoints e cotas do Amazon Redshift na Referência geral do AWS.
- Ao utilizar o roteamento aprimorado da VPC, crie um endpoint do Amazon S3 na mesma VPC do Redshift sem servidor ou no cluster provisionado pelo Redshift. Para obter mais informações, consulte Utilizar o roteamento aprimorado da VPC com o Amazon Redshift como destino do AWS Database Migration Service.
- O AWS DMS com Tecnologia Sem Servidor não é compatível com o Amazon Redshift sem servidor.

Utilizar um banco de dados SAP ASE como destino do AWS Database Migration Service

É possível migrar dados para bancos de dados SAP Adaptive Server Enterprise (ASE), anteriormente conhecido como Sybase-databases, utilizando o AWS DMS, de qualquer banco de dados de origem compatível.

Para obter informações sobre as versões do SAP ASE compatíveis com o AWS DMS como destino, consulte Metas para AWS DMS.

Pré-requisitos para a utilização de um banco de dados SAP ASE como destino do AWS Database Migration Service

Antes de começar a trabalhar com um banco de dados SAP ASE como destino do AWS DMS, confirme se você tem os seguintes pré-requisitos:

- Forneça acesso à conta ao SAP ASE ao usuário do AWS DMS. Esse usuário deve ter privilégios de leitura/gravação no banco de dados SAP ASE.
- Em alguns casos, é possível replicar para o SAP ASE versão 15.7 instalado em uma instância do Amazon EC2 no Microsoft Windows configurada com caracteres não latinos (por exemplo, chinês). Nesses casos, o AWS DMS exigirá que o SAP ASE 15.7 SP121 seja instalado na máquina SAP ASE de destino.

Limitações ao utilizar um banco de dados SAP ASE como destino do AWS DMS

As seguintes limitações se aplicam ao utilizar um banco de dados SAP ASE como destino do AWS DMS:

- O AWS DMS não é compatível com tabelas que incluem campos com os tipos de dados a seguir. As colunas replicadas com esses tipos de dados são exibidas com valores nulos.
  - Tipos definidos pelo usuário (UDT)

Configurações de endpoint ao utilizar o SAP ASE como destino para o AWS DMS

É possível utilizar as configurações de endpoint para configurar o destino do SAP ASE de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--sybase-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o SAP ASE como destino.

Nome	Descrição
`Driver`	Defina esse atributo se quiser utilizar TLS para as versões do ASE 15.7 e superiores. Valor padrão: `Adaptive Server Enterprise` Exemplo: `driver=Adaptive Server Enterprise 16.03.06;` Valores válidos: `Adaptive Server Enterprise 16.03.06`
`AdditionalConnectionProperties`	Quaisquer parâmetros de conexão de ODBC adicionais que deseja especificar.

Tipos de dados de destino do SAP ASE

A tabela a seguir mostra os tipos de dados de destino do banco de dados SAP ASE compatíveis com o AWS DMS e o mapeamento padrão relativo aos tipos de dados do AWS DMS.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipos de dados do AWS DMS	Tipos de dados do SAP ASE
BOOLEAN	BIT
BYTES	VARBINARY (tamanho)
DATE	DATE
TIME	TIME
TIMESTAMP	Se a escala for => 0 e =< 6, use: BIGDATETIME Se a escala for => 7 e =< 9, use: VARCHAR (37)
INT1	TINYINT
INT2	SMALLINT
INT4	INTEGER
INT8	BIGINT
NUMERIC	NUMERIC (p,s)
REAL4	REAL
REAL8	DOUBLE PRECISION
STRING	VARCHAR (tamanho)
UINT1	TINYINT

Tipos de dados do AWS DMS	Tipos de dados do SAP ASE
UINT2	UNSIGNED SMALLINT
UINT4	UNSIGNED INTEGER
UINT8	UNSIGNED BIGINT
WSTRING	VARCHAR (tamanho)
BLOB	IMAGE
CLOB	UNITEXT
NCLOB	TEXT

Utilizar o Amazon S3 como destino de dados do AWS Database Migration Service

É possível migrar dados para o Amazon S3 utilizando o AWS DMS em qualquer uma das origens de bancos de dados compatíveis. Ao utilizar o Amazon S3 como destino em uma tarefa do AWS DMS, os dados da carga máxima e da captura de dados de alteração (CDC) são gravados no formato de valores separados por vírgulas (.csv) por padrão. Para obter mais opções de armazenamento compacto e de consultas mais rápidas, você também tem a opção de ter os dados gravados no formato do Apache Parquet (.parquet).

O AWS DMS nomeia os arquivos criados durante um carga máxima utilizando um contador hexadecimal incremental, por exemplo, LOAD00001.csv, LOAD00002..., LOAD00009, LOAD0000A e assim por diante para arquivos .csv. O AWS DMS nomeia arquivos da CDC utilizando timestamps, por exemplo, 20141029-1134010000.csv. Para cada tabela de origem que contém registros, o AWS DMS cria uma pasta na pasta de destino especificada (se a tabela de destino não estiver vazia). O AWS DMS grava todos os arquivos da carga máxima e da CDC no bucket do Amazon S3 especificado. É possível controlar o tamanho dos arquivos criados pelo AWS DMS utilizando a configuração de endpoint MaxFileSize.

O parâmetro `bucketFolder` contém o local em que arquivos .csv ou .parquet são armazenados antes de serem carregados no bucket do S3. Com arquivos .csv, os dados de tabela são armazenados no seguinte formato no bucket do S3, mostrado com arquivos da carga máxima.

```
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
...
```

É possível especificar o delimitador de coluna, o delimitador de linha e outros parâmetros utilizando os atributos de conexão adicionais. Para obter mais informações sobre os atributos de conexão extra, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS, no final desta seção.

É possível especificar o proprietário do bucket e evitar o corte utilizando a configuração `ExpectedBucketOwner` do endpoint do Amazon S3, conforme mostrado a seguir. Ao fazer uma solicitação para testar uma conexão ou executar uma migração, o S3 verifica o ID da conta do proprietário do bucket em relação ao parâmetro especificado.

```
--s3-settings='{"ExpectedBucketOwner": "AWS_Account_ID"}'
```

Ao utilizar o AWS DMS para replicar alterações de dados utilizando uma tarefa de CDC, a primeira coluna do arquivo de saída .csv ou .parquet indica como os dados da linha foram alterados para o seguinte arquivo .csv.

```
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
```

Para este exemplo, suponha que haja uma tabela `EMPLOYEE` no banco de dados de origem. O AWS DMS grava dados no arquivo .csv ou .parquet, em resposta aos seguintes eventos:

- Um novo funcionário (Bob Smith, ID de funcionário 101) é contratado em 4 de junho de 2014 no escritório de Nova York. No arquivo .csv ou .parquet, o `I` na primeira coluna indica que uma nova linha foi `INSERT` (inserida) na tabela EMPLOYEE no banco de dados de origem.

- Em 8 de outubro de 2015, Bob é transferido para o escritório de Los Angeles. No arquivo .csv ou .parquet, o U indica que a linha correspondente na tabela EMPLOYEE foi UPDATE (atualizada) para refletir o local do novo escritório de Bob. O restante da linha reflete a linha na tabela EMPLOYEE conforme ela aparece após UPDATE.
- Em 13 de março de 2017, Bob é transferido novamente para o escritório de Dallas. No arquivo .csv ou .parquet, o U indica que essa linha foi UPDATE (atualizada) novamente. O restante da linha reflete a linha na tabela EMPLOYEE conforme ela aparece após UPDATE.
- Depois de um tempo trabalhando em Dallas, Bob deixa a empresa. No arquivo .csv ou .parquet, o D indica que a linha foi DELETE (excluída) da tabela de origem. O restante da linha reflete como a linha na tabela EMPLOYEE aparecia antes de ser excluída.

Observe que, por padrão, para a CDC, o AWS DMS armazena as alterações de linha para cada tabela do banco de dados, independentemente da ordem da transação. Para armazenar as alterações de linha nos arquivos de CDC de acordo com a ordem da transação, utilize as configurações do endpoint do S3 para especificar isso e o caminho da pasta em que deseja que os arquivos de transações da CDC sejam armazenados no destino do S3. Para obter mais informações, consulte Captura de dados de alteração (CDC), incluindo a ordem de transações no destino do S3..

Para controlar a frequência de gravações em um destino do Amazon S3 durante uma tarefa de replicação de dados, é possível configurar os atributos de conexão adicionais `cdcMaxBatchInterval` e `cdcMinFileSize`. Isso pode resultar em melhor desempenho ao analisar os dados sem operações adicionais de sobrecarga. Para obter mais informações, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS.

Tópicos

- Utilizar o AWS Glue Data Catalog com um destino do Amazon S3 do AWS DMS
- Utilizar criptografia de dados, arquivos de parquet e de CDC no destino do Amazon S3
- Indicar operações de banco de dados de origem em dados migrados do S3
- Tipos de dados de destino do S3 Parquet

Pré-requisitos da utilização do Amazon S3 como destino

Antes de utilizar o Amazon S3 como destino, verifique se o seguinte é verdadeiro:

- O bucket do S3 que você está utilizando como destino está na mesma região da AWS que a instância de replicação do DMS que você está utilizando para migrar os dados.
- A conta da AWS que você utiliza para a migração tem um perfil do IAM com acesso de gravação e exclusão no bucket do S3 que você está utilizando como destino.
- Esse perfil tem acesso à marcação para que você possa marcar todos os objetos do S3 gravados no bucket de destino.
- O perfil do IAM tem o DMS (dms.amazonaws.com) adicionado como Entidade confiável.

Para configurar esse acesso à conta, verifique se o perfil atribuído à conta de usuário utilizada para criar a tarefa de migração tem o seguinte conjunto de permissões.

```
{
    "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"
            ]
        }
    ]
}
```

Para obter os pré-requisitos para utilizar a validação com o S3 como destino, consulte Pré-requisitos da validação do S3 de destino.

Limitações da utilização do Amazon S3 como destino

Aplicam-se as seguintes limitações ao utilizar o Amazon S3 como destino:

- Não ative o versionamento para o S3. Se o versionamento do S3 for necessário, utilize políticas de ciclo de vida para excluir ativamente as versões antigas. Caso contrário, é possível encontrar falhas na conexão de teste de endpoint devido ao tempo limite de uma chamada `list-object` do S3. Para criar uma política de ciclo de vida para um bucket do S3, consulte Gerenciar o ciclo de vida do armazenamento. Para excluir a versão de um objeto do S3, consulte Excluir versões de objetos de um bucket com versionamento ativado.
- Um bucket do S3 ativado para VPC (VPC do gateway) é compatível com as versões 3.4.7 e superiores.
- Os seguintes comandos da linguagem de definição de dados (DDL) são compatíveis com a captura de dados de alteração (CDC): truncar tabela, descartar tabela, criar tabela, renomear tabela, adicionar coluna, descartar coluna, renomear coluna e alterar o tipo de dados de coluna. Observe que quando uma coluna é adicionada, descartada ou renomeada no banco de dados de origem, nenhuma instrução ALTER é registrada no bucket do S3 de destino e o AWS DMS não altera os registros criados anteriormente para corresponder à nova estrutura. Após a alteração, o AWS DMS cria todos os novos registros utilizando a nova estrutura da tabela.

> **Note**
>
> Uma operação de truncamento da DDL remove todos os arquivos e pastas de tabelas correspondentes em um bucket do S3. É possível utilizar as configurações de tarefas para desativar esse comportamento e configurar a forma como o DMS trata o comportamento da DDL durante a captura de dados de alteração (CDC). Para obter mais informações,

consulte Configurações de tarefa para processamento de DDL de processamento de alterações.

- O modo Full LOB não é compatível.
- As alterações na estrutura da tabela de origem durante a carga máxima não são compatíveis. As alterações nos dados são compatíveis durante a carga máxima.
- Várias tarefas que replicam dados da mesma tabela de origem para o mesmo bucket de endpoint do S3 de destino resultam em tarefas sendo gravadas no mesmo arquivo. Recomendamos que você especifique diferentes endpoints de destino (buckets) se sua fonte de dados estiver na mesma tabela.
- `BatchApply` não é compatível com um endpoint do S3. A utilização da aplicação em lote (por exemplo, a configuração da tarefa de metadados de destino `BatchApplyEnabled`) para um destino do S3 pode resultar em perda de dados.
- Não é possível utilizar `DatePartitionEnabled` ou `addColumnName` em conjunto com `PreserveTransactions` ou `CdcPath`.
- AWS DMS não é compatível com a renomeação de várias tabelas de origem para a mesma pasta de destino utilizando regras de transformação.
- Se houver gravação intensa na tabela de origem durante a fase de carregamento completo, o DMS poderá gravar registros duplicados no bucket do S3 ou nas alterações armazenadas em cache.
- Se você configurar a tarefa com um `TargetTablePrepMode` definido como `DO_NOTHING`, o DMS poderá gravar registros duplicados no bucket do S3 se a tarefa for interrompida e retomada abruptamente durante a fase de carregamento completo.
- Se você configurar o endpoint de destino com a configuração `PreserveTransactions` definida como `true`, o recarregamento de uma tabela não limpará os arquivos de CDC gerados anteriormente. Para obter mais informações, consulte Captura de dados de alteração (CDC), incluindo a ordem de transações no destino do S3..

Para obter as limitações da utilização da validação com o S3 como destino, consulte Limitações da utilização da validação de destino do S3..

Segurança

Para utilizar o Amazon S3 como destino, a conta utilizada para a migração deve ter acesso de gravação e exclusão ao bucket do Amazon S3 utilizado como o destino. Especifique o nome de

recurso da Amazon (ARN) de um perfil do IAM que tem as permissões necessárias para acessar o Amazon S3.

O AWS DMS é compatível com um conjunto de concessões predefinidas do Amazon S3, conhecidas como listas de controle de acesso (ACLs) pré-configuradas padrão. Cada ACL pré-configurada tem um conjunto de concessões e permissões que podem ser utilizadas para definir permissões para o bucket do Amazon S3. É possível especificar uma ACL pré-configurada utilizando `cannedAclForObjects` no atributo da string de conexão para o endpoint de destino do S3. Para obter mais informações sobre como utilizar o atributo de conexão adicional `cannedAclForObjects`, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS. Para obter mais informações sobre ACLs pré-configuradas do Amazon S3, consulte ACL pré-configurada.

O perfil do IAM que você utiliza para a migração deve ser capaz de executar a operação da API `s3:PutObjectAcl`.

Utilizar o Apache Parquet para armazenar objetos do Amazon S3

O formato de valores separados por vírgulas (.csv) é o formato do armazenamento padrão para objetos de destino do Amazon S3. Para obter armazenamento compacto e consultas mais rápidas, é possível utilizar o Apache Parquet (.parquet) como o formato de armazenamento.

O Apache Parquet é um formato de armazenamento de arquivos de código aberto originalmente projetado para o Hadoop. Para obter mais informações sobre o Apache Parquet, consulte https://parquet.apache.org/.

Para definir o .parquet como o formato de armazenamento para os objetos de destino do S3, é possível utilizar os seguintes mecanismos:

- Configurações de endpoint que você fornece como parâmetros de um objeto JSON ao criar o endpoint utilizando a AWS CLI ou a API do AWS DMS. Para obter mais informações, consulte Utilizar criptografia de dados, arquivos de parquet e de CDC no destino do Amazon S3.
- Atributos de conexão adicionais que você fornece como uma lista separada por ponto-e-vírgula ao criar o endpoint. Para obter mais informações, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS.

Marcação de objetos do Amazon S3

É possível marcar objetos do Amazon S3 criados por uma instância de replicação especificando objetos JSON adequados como parte das regras de mapeamento de tarefa-tabela. Para obter mais informações sobre os requisitos e as opções de marcação de objetos do S3, incluindo nomes de tag válidos, consulte Marcação de objetos no Guia do usuário do Amazon Simple Storage Service. Para obter mais informações sobre o mapeamento de tabelas utilizando JSON, consulte Especificar a seleção de tabelas e as regras de transformação utilizando JSON.

Você marca objetos do S3 criados para tabelas e esquemas especificados utilizando um ou mais objetos JSON do tipo de regra `selection`. Então, você segue esse objeto (ou objetos) `selection` por um ou mais objetos JSON do tipo de regra `post-processing` com a ação `add-tag`. Essas regras de pós-processamento identificam os objetos do S3 que você deseja marcar e especificar os nomes e valores das tags que você deseja adicionar a esses objetos do S3.

É possível encontrar os parâmetros para especificar em objetos JSON do tipo de regra `post-processing` na tabela a seguir.

Parâmetro	Possíveis valores	Descrição
`rule-type`	`post-processing`	Um valor que aplica ações de pós-processamento aos objetos de destino gerados. É possível especific ar uma ou mais regras de pós-proce ssamento para marcar objetos do S3 selecionados.
`rule-id`	Um valor numérico.	Um valor numérico exclusivo para identificar a regra.
`rule-name`	Um valor alfanumérico.	Um nome exclusivo para identificar a regra.
`rule-action`	`add-tag`	A ação de pós-processamento que você deseja aplicar ao objeto do S3. É possível adicionar uma ou mais tags utilizando um único objeto fr

Parâmetro	Possíveis valores	Descrição
		pós-processamento do JSON para a ação `add-tag`.
`object-locator`	`schema-name` : o nome do esquema da tabela. `table-name` : o nome da tabela.	O nome de cada esquema e tabela aos quais a regra se aplica. É possível utilizar o sinal de porcentag em "%" como um curinga para todo ou parte do valor de cada parâmetro `object-locator` . Assim, você pode corresponder estes itens: • Uma única tabela em um único esquema • Uma única tabela em alguns ou em todos os esquemas • Algumas ou todas as tabelas em um único esquema. • Algumas ou todas as tabelas em alguns ou em todos os esquemas

Parâmetro	Possíveis valores	Descrição
`tag-set`	`key`: qualquer nome válido para uma única tag. `value`: qualquer valor JSON válido para esta tag.	Os nomes e os valores de uma ou mais tags que você deseja definir em cada objeto do S3 criado, que corresponde ao `object-locator` especificado. É possível especific ar até 10 pares de chave-valor em um único objeto de parâmetro `tag-set`. Para obter mais informações sobre a marcação de objetos do S3, consulte Marcação de objetos no Guia do usuário do Amazon Simple Storage Service. Também é possível especificar um valor dinâmico para todo ou parte do valor para os parâmetros `value` e `key` de uma tag utilizand o `${dyn-value }`. Aqui, `${dyn-value }` pode ser `${schema-name}` ou `${table-name}` . Portanto, é possível inserir o nome do esquema ou da tabela seleciona da no momento como todo ou qualquer parte do valor do parâmetro . **Note** **Important** Se você inserir um valor dinâmico para o parâmetro `key`, poderá gerar tags com nomes

Parâmetro	Possíveis valores	Descrição
		duplicados para um objeto do S3, dependendo de como usá-lo. Nesse caso, apenas uma das configurações de tag duplicadas é adicionada ao objeto.

Ao especificar vários tipos de regra `post-processing` para marcar uma seleção de objetos do S3, cada objeto do S3 é marcado utilizando apenas um objeto `tag-set` de uma regra de pós-processamento. O conjunto de tags específico utilizado para marcar um determinado objeto do S3 é aquele da regra de pós-processamento cujo localizador de objeto associado corresponde melhor ao objeto do S3.

Por exemplo, suponha que duas regras de pós-processamento identificam o mesmo objeto do S3. Suponha também que o localizador de objetos de uma regra use curingas, e que o localizador de objetos de outra regra use uma correspondência exata para identificar o objeto do S3 (sem curingas). Nesse caso, o conjunto de tags associado à regra pós-processamento com a correspondência exata é utilizado para marcar o objeto do S3. Se várias regras de pós-processamento corresponderem a um determinado objeto do S3 igualmente bem, o conjunto de tags associado à primeira regra de pós-processamento será utilizado para marcar o objeto.

Example Adicionar tags estáticas a um objeto do S3 criado para uma única tabela e esquema

As seguintes regras de seleção e pós-processamento adicionam três tags (`tag_1`, `tag_2` e `tag_3` com valores estáticos correspondentes, `value_1`, `value_2` e `value_3`) a um objeto do S3 criado. Esse objeto do S3 corresponde a uma única tabela na origem chamada `STOCK` com um esquema chamado `aat2`.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "5",
```

```
            "rule-name": "5",
            "object-locator": {
                "schema-name": "aat2",
                "table-name": "STOCK"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "post-processing",
            "rule-id": "41",
            "rule-name": "41",
            "rule-action": "add-tag",
            "object-locator": {
                "schema-name": "aat2",
                "table-name": "STOCK"
            },
            "tag-set": [
              {
                "key": "tag_1",
                "value": "value_1"
              },
              {
                "key": "tag_2",
                "value": "value_2"
              },
              {
                "key": "tag_3",
                "value": "value_3"
              }
           ]
        }
    ]
}
```

Example Adicionar tags estáticas e dinâmicas a objetos do S3 criados para várias tabelas e esquemas

O exemplo a seguir tem uma seleção e duas regras de pós-processamento, em que a entrada da origem inclui todas as tabelas e todos os seus esquemas.

```
{
    "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"
              }           ]
        }
    ]
}
```

A primeira regra de pós-processamento adiciona duas tags (`dw-schema-name` e `dw-schema-table`) com valores dinâmicos correspondentes (`${schema-name}` e `my_prefix_${table-name}`) a quase todos os objetos do S3 criados no destino. A exceção é o objeto do S3 identificado e marcado com a segunda regra de pós-processamento. Assim, cada objeto de destino do S3 identificado pelo localizador de objetos curinga é criado com tags que identificam o esquema e a tabela à qual ele corresponde na origem.

A segunda regra de pós-processamento adiciona `tag_1` e `tag_2` com valores estáticos correspondentes `value_1` e `value_2` a um objeto do S3 criado, que é identificado por um localizador de objetos de correspondência exata. Esse objeto do S3 criado, portanto, corresponde à uma única tabela na origem chamada `ITEM` com um esquema chamado `aat`. Devido à correspondência exata, essas tags substituem todas as tags nesse objeto adicionado na primeira regra de pós-processamento, que corresponde a objetos do S3 apenas por meio de curinga.

Example Adicionar nomes e valores de tags dinâmicas a objetos do S3

O exemplo a seguir tem duas regras de seleção e uma regra de pós-processamento. Aqui, a entrada da origem inclui apenas a tabela `ITEM` no esquema `retail` ou `wholesale`.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "retail",
                "table-name": "ITEM"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "selection",
            "rule-id": "1",
```

```
            "rule-name": "1",
            "object-locator": {
                "schema-name": "wholesale",
                "table-name": "ITEM"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "post-processing",
            "rule-id": "21",
            "rule-name": "21",
            "rule-action": "add-tag",
            "object-locator": {
                "schema-name": "%",
                "table-name": "ITEM",
            },
            "tag-set": [
              {
                "key": "dw-schema-name",
                "value":"${schema-name}"
              },
              {
                "key": "dw-schema-table",
                "value": "my_prefix_ITEM"
              },
              {
                "key": "${schema-name}_ITEM_tag_1",
                "value": "value_1"
              },
              {
                "key": "${schema-name}_ITEM_tag_2",
                "value": "value_2"
              }
            ]
    ]
}
```

O conjunto de tags da regra de pós-processamento adiciona duas tags (`dw-schema-name` e `dw-schema-table`) para todos os objetos do S3 criados para a tabela `ITEM` no destino. A primeira tag tem o valor dinâmico `"${schema-name}"`, e a segunda tag tem um valor estático, `"my_prefix_ITEM"`. Assim, cada objeto de destino do S3 é criado com tags que identificam o esquema e a tabela à qual ele corresponde na origem.

Além disso, o conjunto de tags adiciona duas tags adicionais com nomes dinâmicos (`${schema-name}_ITEM_tag_1` e `"${schema-name}_ITEM_tag_2"`). Essas têm valores estáticos correspondentes `value_1` e `value_2`. Portanto, cada uma dessas tags são nomeadas pelo esquema atual, `retail` ou `wholesale`. Não é possível criar um nome de tag dinâmico duplicado nesse objeto, porque cada objeto é criado para um único nome de esquema exclusivo. O nome do esquema é utilizado para criar um nome de tag exclusivo de outra forma.

Criar chaves do AWS KMS para criptografar objetos de destino do Amazon S3

É possível criar e utilizar chaves personalizadas do AWS KMS para criptografar os objetos de destino do Amazon S3. Depois de criar uma chave do KMS, é possível utilizá-la para criptografar objetos utilizando uma das seguintes abordagens ao criar o endpoint de destino do S3:

- Utilize as seguintes opções para objetos de destino do S3 (com o formato de armazenamento de arquivo .csv padrão) ao executar o comando `create-endpoint` utilizando a AWS CLI.

  ```
  --s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN",
  "CsvRowDelimiter": "\n", "CsvDelimiter": ",", "BucketFolder": "your-bucket-folder",
  "BucketName": "your-bucket-name", "EncryptionMode": "SSE_KMS",
  "ServerSideEncryptionKmsKeyId": "your-KMS-key-ARN"}'
  ```

  Aqui, *`your-KMS-key-ARN`* é o nome de recurso da Amazon (ARN) de sua chave do KMS. Para obter mais informações, consulte Utilizar criptografia de dados, arquivos de parquet e de CDC no destino do Amazon S3.

- Defina o atributo de conexão adicional `encryptionMode` como o valor `SSE_KMS`, e o atributo de conexão adicional `serverSideEncryptionKmsKeyId` como o ARN de sua chave do KMS. Para obter mais informações, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS.

Para criptografar os objetos de destino do Amazon S3 utilizando uma chave do KMS, você precisa de um perfil do IAM que tenha permissões para acessar o bucket do Amazon S3. Esse perfil do IAM é acessado em uma política (uma política de chaves) anexada à chave de criptografia criada. É possível fazer isso no console do IAM criando o seguinte:

- Uma política com permissões para acessar o bucket do Amazon S3.
- Um perfil do IAM com essa política.
- A chave de criptografia do KMS com uma política de chaves que faz referência a esse perfil.

Os procedimentos a seguir descrevem como fazer isso.

Para criar uma política do IAM com permissões para acessar o bucket do Amazon S3.

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies (Políticas). A página Policies (Políticas) é aberta.
3. Escolha Create policy (Criar política). A página Create policy (Criar política) é aberta.
4. Escolha Service (Serviço) e escolha S3. Uma lista de permissões de ação é exibida.
5. Escolha Expand all (Expandir tudo) para expandir a lista e escolha as seguintes permissões no mínimo:

   - ListBucket
   - PutObject
   - DeleteObject

   Escolha todas as outras permissões necessárias e escolha Collapse all (Recolher tudo) para recolher a lista.
6. Escolha Resouces (Recursos) para especificar os recursos que você deseja acessar. No mínimo, escolha Todos os recursos para fornecer acesso geral aos recursos do Amazon S3.
7. Adicione todas as outras condições ou permissões necessárias e escolha Review policy (Revisar política). Verifique os resultados na página Review policy (Revisar política).
8. Se as configurações forem o que você precisa, insira um nome para a política (por exemplo, `DMS-S3-endpoint-access`), e qualquer descrição adicional e escolha Criar política. A página Policies (Políticas) é aberta com uma mensagem indicando que sua política foi criada.
9. Pesquise e escolha o nome da política na lista Policies (Políticas). A página Summary (Resumo) é aberta exibindo o JSON da política, semelhante ao seguinte.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListBucket",
                "s3:DeleteObject"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Agora, você criou a política para acessar os recursos do Amazon S3 para a criptografia com um nome específico, por exemplo `DMS-S3-endpoint-access`.

Como criar um perfil do IAM com essa política

1. No console do IAM, escolha Perfis no painel de navegação. A página de detalhes de Perfis é aberta.
2. Selecione Criar perfil. A página Create role (Criar função) é aberta.
3. Com o serviço da AWS selecionado como a entidade confiável, escolha DMS como o serviço que utilizará o perfil do IAM.
4. Escolha Próximo: permissões. A visualização Attach permissions policies (Anexar políticas de permissões) é exibida na página Create role (Criar função).
5. Encontre e selecione a política do IAM para o perfil do IAM criado no procedimento anterior (`DMS-S3-endpoint-access`).
6. Escolha Próximo: etiquetas. A visualização Add tags (Adicionar tags) é exibida na página Create role (Criar função). Aqui, você pode adicionar todas as tags desejadas.
7. Escolha Próximo: revisar. A visualização Review (Revisão) é exibida na página Create role (Criar função). Aqui, é possível verificar os resultados.
8. Se as configurações forem o que você precisa, insira um nome para o perfil (obrigatório, por exemplo, `DMS-S3-endpoint-access-role`), e qualquer descrição adicional e escolha Criar função. A página de detalhes Funções é aberta com uma mensagem indicando que o perfil foi criado.

Agora, você criou o perfil para acessar os recursos do Amazon S3 para criptografia com um nome especificado, por exemplo, `DMS-S3-endpoint-access-role`.

Como criar uma chave de criptografia do KMS com uma política de chave que faz referência ao perfil do IAM

> **Note**
>
> Para obter mais informações sobre como AWS DMS funciona com chaves de criptografia do AWS KMS, consulte Configurando uma chave de criptografia e especificando permissões AWS KMS.

1. Faça login no AWS Management Console e abra o console do AWS Key Management Service (AWS KMS) em https://console.aws.amazon.com/kms.
2. Para alterar a Região da AWS, use o Region selector (Seletor de regiões) no canto superior direito da página.
3. No painel de navegação, escolha Customer managed keys (Chaves gerenciadas pelo cliente).
4. Escolha Create key (Criar chave). A página Configure key (Configurar chave) é aberta.
5. Para Key type (Tipo de chave), escolha Symmetric (Simétrica).

   > **Note**
   >
   > Ao criar essa chave, só é possível criar uma chave simétrica, pois todos os serviços da AWS, como o Amazon S3, funcionam somente com chaves de criptografia simétricas.

6. Escolha Opções avançadas. Para Key material origin (Origem do material da chave), certifique-se de que o KMS está escolhido e escolha Next (Próximo). A página Add labels (Adicionar rótulos) é aberta.
7. Em Create alias and description (Criar alias e descrição), insira um alias para a chave (por exemplo, `DMS-S3-endpoint-encryption-key`) e qualquer descrição adicional.
8. Em Tags, adicione todas as tags desejadas para ajudar a identificar a chave e controlar seu uso e escolha Next (Próximo). A página Define key administrative permissions (Definir permissões administrativas de chaves) é aberta mostrando uma lista de usuários e funções que podem ser escolhidos.
9. Adicione os usuários e as funções desejados para gerenciar a chave. Certifique-se de que esses usuários e funções tenham as permissões necessárias para gerenciar a chave.
10. Em Key deletion (Exclusão de chaves), escolha se os administradores de chaves podem excluir a chave e escolha Next (Próximo). A página Define key usage permissions (Definir permissões

de uso de chaves) é aberta mostrando uma lista adicional de usuários e funções que podem ser escolhidos.

11. Para Esta conta, escolha os usuários disponíveis que você deseja que executem operações criptográficas em destinos do Amazon S3. Escolha também o perfil criado anteriormente em Perfis para ativar o acesso à criptografia dos objetos de destino do Amazon S3, por exemplo, `DMS-S3-endpoint-access-role`).
12. Se quiser adicionar outras contas não listadas para ter esse mesmo acesso, em Outras contas da AWS, escolha Adicionar outra conta da AWS e escolha Próximo. A página Review and edit key policy (Rever e editar política de chave) é aberta mostrando o JSON da política de chave que você pode revisar e editar digitando no JSON existente. Aqui, a política de chave que faz referência à função e aos usuários é mostrada (por exemplo, `Admin` e `User1`) que você escolheu na etapa anterior. Também é possível ver as diferentes ações de chaves permitidas para as várias entidades principais (usuários e perfis), conforme mostrado no exemplo a seguir.

```
{
  "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. Escolha Terminar. A página Chaves de criptografia é aberta com uma mensagem indicando que a chave do KMS foi criada.

Agora, você criou uma nova chave do KMS com um alias especificado (por exemplo, `DMS-S3-endpoint-encryption-key`). Essa chave permite que o AWS DMS criptografe os objetos de destino do Amazon S3.

Utilizar o particionamento de pastas com base em data

O AWS DMS é compatível com partições de pastas do S3 com base na data de confirmação da transação ao utilizar o Amazon S3 como o endpoint de destino. Utilizando o particionamento de pastas com base em data, é possível gravar dados de uma única tabela de origem em uma estrutura de pastas com hierarquia temporal em um bucket do S3. Ao particionar pastas ao criar um endpoint de destino do S3, é possível fazer o seguinte:

- Gerenciar melhor os objetos do S3
- Limitar o tamanho de cada pasta do S3
- Otimizar consultas de data lake ou outras operações subsequentes

É possível ativar o particionamento de pastas com base em data ao criar um endpoint de destino do S3. É possível ativá-lo ao migrar dados existentes e ao replicar alterações em andamento (carga máxima + CDC) ou replicar somente alterações de dados (somente CDC). Utilize as seguintes configurações de endpoint de destino:

- `DatePartitionEnabled`: especifica o particionamento com base em datas. Defina esta opção Boolean como `true` para particionar pastas de bucket do S3 com base nas datas de confirmação da transação.

  Não é possível utilizar essa configuração com `PreserveTransactions` ou `CdcPath`.

  O valor padrão é `false`.
- `DatePartitionSequence`: identifica a sequência do formato de data a ser utilizado durante o particionamento de pastas. Defina esta opção ENUM como `YYYYMMDD`, `YYYYMMDDHH`, `YYYYMM`, `MMYYYYDD` ou `DDMMYYYY`. O valor padrão é `YYYYMMDD`. Utilize essa configuração quando `DatePartitionEnabled` estiver definido como `true`..
- `DatePartitionDelimiter`: especifica um delimitador de separação de datas a ser utilizado durante o particionamento de pastas. Defina esta opção ENUM como `SLASH`, `DASH`, `UNDERSCORE` ou `NONE`. O valor padrão é `SLASH`. Utilize essa configuração quando `DatePartitionEnabled` estiver definido como `true`.

O exemplo a seguir mostra como ativar o particionamento de pastas com base em data, com valores padrão para a sequência e o delimitador da partição de dados. Ele utiliza a opção `--s3-settings '{`*`json-settings`*`}'` AWS CLI. Comando da `create-endpoint`.

```
  --s3-settings '{"DatePartitionEnabled": true,"DatePartitionSequence":
"YYYYMMDD","DatePartitionDelimiter": "SLASH"}'
```

Carga paralela de origens particionadas ao utilizar o Amazon S3 como destino do AWS DMS

É possível configurar uma carga máxima paralela de fontes de dados particionadas para destinos do Amazon S3. Essa abordagem melhora os tempos de carga da migração de dados particionados nos mecanismos de banco de dados de origem compatíveis para o destino do S3. Para melhorar os tempos de carga dos dados de origem particionados, crie subpastas de destino do S3 mapeadas para as partições de cada tabela no banco de dados de origem. Essas subpastas vinculadas à partição permitem que o AWS DMS execute processos paralelos para preencher cada subpasta no destino.

Para configurar uma carga máxima paralela de um destino do S3, o S3 é compatível com os três tipos de regra `parallel-load` para a regra `table-settings` de mapeamento de tabela:

- `partitions-auto`
- `partitions-list`
- `ranges`

Para obter mais informações sobre esses tipos de regra de carga paralela, consulte Regras e operações de configurações de tabelas e coleções.

Para os tipos de regra `partitions-auto` e `partitions-list`, o AWS DMS utiliza o nome de cada partição do endpoint de origem para identificar a estrutura da subpasta de destino, da seguinte forma.

```
bucket_name/bucket_folder/database_schema_name/table_name/partition_name/
LOADseq_num.csv
```

Aqui, o caminho da subpasta em que os dados são migrados e armazenados no destino do S3 inclui uma subpasta *`partition_name`* adicional que corresponde a uma partição de origem com o mesmo nome. Essa subpasta *`partition_name`* armazena um ou mais arquivos `LOAD`*`seq_num`*`.csv` contendo os dados migrados da partição de origem especificada. Aqui, *`seq_num`* é o sufixo do número de sequência no nome do arquivo .csv, como `00000001` no arquivo .csv com o nome `LOAD00000001.csv`.

No entanto, alguns mecanismos de banco de dados, como MongoDB e o DocumentDB, não têm o conceito de partições. Para esses mecanismos de banco de dados, o AWS DMS adiciona o índice do segmento de origem em execução como um prefixo para o nome do arquivo .csv de destino, da seguinte maneira.

```
.../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
```

Aqui, os arquivos `SEGMENT1_LOAD00000001.csv` e `SEGMENT1_LOAD00000002.csv` são nomeados com o mesmo prefixo de índice do segmento de origem em execução, `SEGMENT1`. Eles são nomeados assim porque os dados de origem migrados para esses dois arquivos .csv estão associados ao mesmo índice de segmento de origem em execução. Por outro lado, os dados migrados armazenados em cada um dos arquivos `SEGMENT2_LOAD00000009.csv` e `SEGMENT3_LOAD0000000A.csv` de destino estão associados a diferentes índices de segmentos de

origem em execução. Cada arquivo tem seu nome prefixado com o nome de seu índice de segmento em execução, `SEGMENT2` e `SEGMENT3`.

Para o tipo de carga paralela `ranges`, defina os nomes e valores das colunas utilizando as configurações `columns` e `boundaries` das regras `table-settings`. Com essas regras, é possível especificar partições correspondentes aos nomes dos segmentos, da seguinte maneira.

```
"parallel-load": {
    "type": "ranges",
    "columns": [
         "region",
         "sale"
    ],
    "boundaries": [
          [
               "NORTH",
               "1000"
          ],
          [
               "WEST",
               "3000"
          ]
    ],
    "segment-names": [
          "custom_segment1",
          "custom_segment2",
          "custom_segment3"
    ]
}
```

Aqui, a configuração de `segment-names` define nomes de três partições para migrar dados em paralelo no destino do S3. Os dados migrados são carregados paralelamente e armazenados em arquivos .csv nas subpastas da partição em ordem, da seguinte maneira.

```
.../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
```

Aqui, o AWS DMS armazena uma série de arquivos .csv em cada uma das três subpastas de partição. A série de arquivos .csv em cada subpasta de partição é nomeada de forma incremental, começando com `LOAD00000001.csv` até que todos os dados sejam migrados.

Em alguns casos, é possível não nomear explicitamente as subpastas de partição para um tipo de carga paralela `ranges` utilizando a configuração `segment-names`. Nesse caso, o AWS DMS aplica o padrão de criação de cada série de arquivos .csv em sua subpasta *`table_name`*. Aqui, o AWS DMS prefixa os nomes dos arquivos de cada série de arquivos .csv com o nome do índice do segmento de origem em execução, da seguinte forma.

```
.../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
```

Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados de destino do Amazon S3 de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--s3-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o Amazon S3 como destino.

Opção	Descrição
`CsvNullValue`	Um parâmetro opcional que especifica como o AWS DMS trata valores nulos. Ao tratar o valor nulo, é possível utilizar esse parâmetro para passar uma string definida pelo usuário como nula ao gravar no destino. Por exemplo, quando as colunas de destino forem anuláveis, é possível utilizar essa opção para diferenciar entre o valor de string vazia e o valor nulo. Portanto, se você definir o valor desse parâmetro como a string vazia (" " ou "), o AWS DMS tratará a string vazia como o valor nulo em vez de `NULL`. Valor padrão: `NULL` Valores válidos: qualquer string válida. Exemplo: `--s3-settings '{"CsvNullValue": " "}'`

Opção	Descrição
`AddColumnName`	Um parâmetro opcional que, quando definido como `true` ou `y`, pode ser utilizado para adicionar informações de nome de coluna ao arquivo .csv de saída. Não é possível utilizar esse parâmetro com `PreserveTransactions` ou `CdcPath`. Valor padrão: `false` Valores válidos: `true`, `false`, `y`, `n` Exemplo: `--s3-settings '{"AddColumnName": true}'`
`AddTrailingPaddingCharacter`	Utilize a configuração `AddTrailingPaddingCharacter` do endpoint de destino do S3 para adicionar preenchimento aos dados da string. O valor padrão é `false`. Tipo: booliano Exemplo: `--s3-settings '{"AddTrailingPaddingCharacter": true}'`
`BucketFolder`	Um parâmetro opcional para definir um nome de pasta no bucket do S3. Se fornecidos, os objetos de destino serão criados como arquivos .csv ou .parquet no caminho *`BucketFolder`* `/`*`schema_name`* `/`*`table_name`* `/`. Se esse parâmetro não for especificado, o caminho utilizado será *`schema_name`* `/`*`table_name`* `/`. Exemplo: `--s3-settings '{"BucketFolder": "testFolder"}'`
`BucketName`	O nome do bucket do S3 no qual os objetos de destino do S3 são criados como arquivos .csv ou .parquet. Exemplo: `--s3-settings '{"BucketName": "buckettest"}'`

Opção	Descrição
`CannedAcl ForObjects`	Permite que o AWS DMS especifique uma lista de controle de acesso (pré-configurada) predefinida no bucket do S3 como arquivos .csv ou .parquet. Para obter mais informações sobre ACLs pré-configuradas do Amazon S3, consulte ACL pré-configurada no Guia do desenvolvedor do Amazon S3. Valor padrão: NONE Os valores válidos para este atributo são: NONE; PRIVATE; PUBLIC_READ; PUBLIC_READ_WRITE; AUTHENTICATED_READ; AWS_EXEC_READ; BUCKET_OWNER_READ; BUCKET_OWNER_FULL_CONTROL. Exemplo: `--s3-settings '{"CannedAclForObjects": "PUBLIC_READ"}'`

Opção	Descrição
`CdcInsertsOnly`	Um parâmetro opcional durante um carga de captura de dados de alteração (CDC) para gravar apenas operações INSERT nos arquivos de saída de valores separados por vírgulas (.csv) ou de armazenamento colunar (.parquet). Por padrão (a configuração `false`), o primeiro campo em um registro .csv ou .parquet contém a letra I (INSERT), U (UPDATE) ou D (DELETE). Esta carta indica se a linha foi inserida, atualizada ou excluída no banco de dados de origem para um carregamento CDC no destino. Se `cdcInsertsOnly` estiver definido como `true` ou `y`, somente os INSERTs do banco de dados de origem serão migrados para o arquivo .csv ou .parquet. Somente no caso do formato .csv, a maneira como esses INSERTS são registrados depende do valor `IncludeOpForFullLoad`. Se `IncludeOpForFullLoad` estiver definido como `true`, o primeiro campo de cada registro CDC será definido como I para indicar a operação INSERT na origem. Se `IncludeOpForFullLoad` estiver definido como `false`, os registros CDC serão gravados sem um primeiro campo indicando a operação INSERT na origem. Para obter mais informações sobre como esses parâmetros funcionam juntos, consulte Indicar operações de banco de dados de origem em dados migrados do S3. Valor padrão: `false` Valores válidos: `true`, `false`, `y`, `n` Exemplo: `--s3-settings '{"CdcInsertsOnly": true}'`

Opção	Descrição
`CdcInsertsAndUpdates`	Habilita uma carga de captura de dados de alteração (CDC) para gravar operações INSERT e UPDATE em arquivos de saída .csv ou .parquet (armazenamento colunar). A configuração padrão é `false`, mas quando `cdcInsertsAndUpdates` é definido como `true` ou `y`, INSERTs e UPDATEs no banco de dados de origem são migrados para o arquivo .csv ou .parquet. Somente para o formato de arquivo .csv, a maneira como esses INSERTs e UPDATEs são registrados depende do valor do parâmetro `includeOpForFullLoad` . Se `includeOpForFullLoad` estiver definido como `true`, o primeiro campo de cada registro CDC será definido como `I` ou `U` para indicar operações INSERT e UPDATE na origem. Mas se `includeOpForFullLoad` for definido como `false`, os registros CDC serão gravados sem uma indicação de operações INSERT ou UPDATE na origem. Para obter mais informações sobre como esses parâmetros funcionam juntos, consulte Indicar operações de banco de dados de origem em dados migrados do S3. **Note** `CdcInsertsOnly` e `cdcInsertsAndUpdates` não podem ambos serem definidos como true para o mesmo endpoint. Defina `cdcInsertsOnly` ou `cdcInsertsAndUpdates` `true` para o mesmo endpoint, mas não os dois. Valor padrão: `false` Valores válidos: `true`, `false`, `y`, `n` Exemplo: `--s3-settings '{"CdcInsertsAndUpdates": true}'`

Opção	Descrição
`CdcPath`	Especifica o caminho da pasta dos arquivos CDC. Para uma origem do S3, essa configuração é necessária se uma tarefa captura os dados de alterações; caso contrário, ela é opcional. Se `CdcPath` estiver definido, o DMS lerá os arquivos da CDC nesse caminho e replicará as alterações de dados no endpoint de destino. Para um destino do S3, se você definir `PreserveTransactions` como verdadeiro, o DMS verificará se você definiu esse parâmetro para um caminho de pasta no destino do S3, em que o DMS pode salvar a ordem das transações na carga da CDC. O DMS cria esse caminho de pasta CDC no diretório de trabalho de destino do S3 ou no local de destino do S3 especificado por `BucketFolder` e `BucketName` . Não é possível utilizar esse parâmetro com `DatePartitionEnabled` ou `AddColumnName` . Tipo: string Por exemplo, se você especificar `CdcPath` como `MyChangedData` e especificar `BucketName` como `MyTargetBucket` , mas não especificar `BucketFolder` , o DMS criará o seguinte caminho da pasta CDC: `MyTargetBucket/MyChangedData` . Se você especificar o mesmo `CdcPath` e especificar `BucketName` como `MyTargetBucket` e `BucketFolder` como `MyTargetData` , o DMS criará o seguinte caminho da pasta CDC: `MyTargetBucket/MyTargetData/MyChangedData` . **Note** Essa configuração é compatível no AWS DMS versões 3.4.2 e superiores. Ao capturar alterações de dados na ordem da transação, o DMS sempre armazena as alterações de linha em arquivos .csv, independentemente do valor da configuração de DataFormat

Opção	Descrição
	do S3 no destino. O DMS não salva as alterações de dados na ordem da transação utilizando arquivos .parquet.
`CdcMaxBatchInterval`	Condição de tamanho máximo do intervalo, definido em segundos, para a saída de um arquivo para o Amazon S3. Valor padrão: 60 segundos Quando `CdcMaxBatchInterval` e `CdcMinFileSize` forem especificados, a gravação do arquivo será acionada por qualquer condição de parâmetro que seja atendida primeiro.
`CdcMinFileSize`	Condição de tamanho mínimo do arquivo, definido em kilobytes, para a saída de um arquivo para o Amazon S3. Valor padrão: 32000 KB Quando `CdcMinFileSize` e `CdcMaxBatchInterval` forem especificados, a gravação do arquivo será acionada por qualquer condição de parâmetro que seja atendida primeiro.

Opção	Descrição
`PreserveTransactions`	Se definido como `true`, o DMS salvará a ordem da transação para a captura de dados de alteração (CDC) no destino do Amazon S3 especificado por `CdcPath`. Não é possível utilizar esse parâmetro com `DatePartitionEnabled` ou `AddColumnName` . Tipo: booliano Ao capturar alterações de dados na ordem da transação, o DMS sempre armazena as alterações de linha em arquivos .csv, independentemente do valor da configuração de DataFormat do S3 no destino. O DMS não salva as alterações de dados na ordem da transação utilizando arquivos .parquet. **Note** Essa configuração é compatível no AWS DMS versões 3.4.2 e superiores.

Opção	Descrição
`IncludeOp ForFullLoad`	Um parâmetro opcional durante um carga máxima para gravar operações INSERT só nos arquivos de saída de valores separados por vírgulas (.csv). Em cargas máximas, os registros só podem ser inseridos. Por padrão (a configuração `false`), não há informações gravadas nesses arquivos de saída para um carga máxima indicando que as linhas foram inseridas no banco de dados de origem. Se `IncludeOpForFullLoad` estiver definido como `true` ou `y`, INSERT será registrado utilizando a letra I no primeiro campo do arquivo .csv. **Note** Esse parâmetro funciona em conjunto com `CdcInsertsOnly` ou `CdcInsertsAndUpdates` somente para arquivos .csv de saída. Para obter mais informações sobre como esses parâmetros funcionam juntos, consulte Indicar operações de banco de dados de origem em dados migrados do S3. Valor padrão: `false` Valores válidos: `true`, `false`, `y`, `n` Exemplo: `--s3-settings '{"IncludeOpForFullLoad": true}'`
`CompressionType`	Um parâmetro opcional ao definir como `GZIP` utiliza o GZIP para compactar os arquivos .csv ou .parquet de destino. Quando esse parâmetro é definido como o padrão, ele deixa os arquivos descompac tados. Valor padrão: `NONE` Valores válidos: `GZIP` ou `NONE` Exemplo: `--s3-settings '{"CompressionType": "GZIP"}'`

Opção	Descrição
`CsvDelimiter`	O delimitador utilizado para separar colunas nos arquivos .csv de origem. O padrão é uma vírgula (,). Exemplo: `--s3-settings '{"CsvDelimiter": ","}'`
`CsvRowDelimiter`	O delimitador utilizado para separar linhas nos arquivos .csv de origem. O padrão é uma nova linha (\n). Exemplo: `--s3-settings '{"CsvRowDelimiter": "\n"}'`
`MaxFileSize`	Um valor que especifica o tamanho máximo (em KB) de qualquer arquivo .csv a ser criado ao migrar para um destino do S3 durante a carga máxima. Valor padrão: 1.048.576 KB (1 GB) Valores válidos: 1 a 1.048.576 Exemplo: `--s3-settings '{"MaxFileSize": 512}'`
`Rfc4180`	Um parâmetro opcional utilizado para definir o comportamento com o propósito de determinar a conformidade com RFC para os dados migrados para o Amazon S3 utilizando somente o formato de arquivo .csv. Quando esse valor é definido como `true` ou y utilizando o Amazon S3 como destino, se os dados tiverem aspas, vírgulas ou caracteres de nova linha, o AWS DMS envolverá toda a coluna com um par de aspas duplas adicionais ("). Cada aspa dentro dos dados é repetida duas vezes. Essa formatação está em conformidade com RFC 4180. Valor padrão: `true` Valores válidos: `true`, `false`, `y`, `n` Exemplo: `--s3-settings '{"Rfc4180": false}'`

Opção	Descrição
`EncryptionMode`	O modo de criptografia do lado do servidor que você deseja utilizar para criptografar os arquivos de objetos .csv ou .parquet copiados no S3. Os valores válidos são `SSE_S3` (criptografia no lado do servidor do S3) ou `SSE_KMS` (criptografia da chave do KMS). Se você escolher `SSE_KMS`, defina o parâmetro `ServerSideEncryptionKmsKeyId` como o Nome do recurso da Amazon (ARN) para a chave do KMS a ser usada para a criptografia. **Note** Também é possível utilizar o comando `modify-endpoint` da CLI para alterar o valor do atributo `EncryptionMode` de um endpoint existente de `SSE_KMS` para `SSE_S3`. Mas não é possível alterar o valor de `EncryptionMode` de `SSE_S3` para `SSE_KMS`. Valor padrão: `SSE_S3` Valores válidos: `SSE_S3` ou `SSE_KMS` Exemplo: `--s3-settings '{"EncryptionMode": SSE_S3}'`
`ServerSideEncryptionKmsKeyId`	Se você definir `EncryptionMode` como `SSE_KMS`, defina esse parâmetro como o nome do recurso da Amazon (ARN) da chave do KMS. É possível encontrar esse ARN selecionando o alias da chave na lista de chaves do AWS KMS criada para a sua conta. Ao criar a chave, você deve associar políticas e perfis específicos associadas a essa chave do KMS. Para obter mais informações, consulte Criar chaves do AWS KMS para criptografar objetos de destino do Amazon S3. Exemplo: `--s3-settings '{"ServerSideEncryptionKmsKeyId":"arn:aws:kms:us-east-1:111122223333:key/11a1a1a1-aaaa-9999-abab-2bbbbbb222a2"}'`

Opção	Descrição
`DataFormat`	O formato de saída dos arquivos que o AWS DMS utiliza para criar objetos do S3. Para destinos do Amazon S3, o AWS DMS é compatível com arquivos .csv ou .parquet. Os arquivos .parquet têm um formato de armazenamento colunar binário com opções de compactação eficiente s e desempenho de consulta mais rápido. Para obter mais informações sobre os arquivos .parquet, consulte https://parquet.apache.org/. Valor padrão: `csv` Valores válidos: `csv` ou `parquet` Exemplo: `--s3-settings '{"DataFormat": "parquet"}'`
`EncodingType`	O tipo de codificação Parquet. As opções do tipo de compactação incluem as seguintes: • `rle-dictionary` : essa codificação de dicionário utiliza uma combinação de empacotamento de bits e de codificação de run-length para armazenar valores repetidos de forma mais eficiente. • `plain`: sem codificação. • `plain-dictionary` : essa codificação de dicionário compila um dicionário dos valores encontrados em uma determinada coluna. O dicionário é armazenado em uma página de dicionário para cada bloco da coluna. Valor padrão: `rle-dictionary` Valores válidos: `rle-dictionary` , `plain` ou `plain-dictionary` Exemplo: `--s3-settings '{"EncodingType": "plain-di ctionary"}'`

Opção	Descrição
`DictPageSizeLimit`	O tamanho máximo permitido, em bytes, para uma página de dicionário em um arquivo .parquet. Se uma página de dicionário exceder esse valor, a página usará a codificação simples. Valor padrão: 1.024.000 (1 MB) Valores válidos: qualquer valor inteiro válido Exemplo: `--s3-settings '{"DictPageSizeLimit": 2,048,000}'`
`RowGroupLength`	O número de linhas em um grupo de linhas de um arquivo .parquet. Valor padrão: 10.024 (10 KB) Valores válidos: qualquer número inteiro válido Exemplo: `--s3-settings '{"RowGroupLength": 20,048}'`
`DataPageSize`	O tamanho máximo permitido, em bytes, para uma página de dados em um arquivo .parquet. Valor padrão: 1.024.000 (1 MB) Valores válidos: qualquer número inteiro válido Exemplo: `--s3-settings '{"DataPageSize": 2,048,000}'`
`ParquetVersion`	A versão do formato do arquivo .parquet. Valor padrão: `PARQUET_1_0` Valores válidos: `PARQUET_1_0` ou `PARQUET_2_0` Exemplo: `--s3-settings '{"ParquetVersion": "PARQUET_2_0"}'`

Opção	Descrição
`EnableStatistics`	Defina como `true` ou `y` para ativar as estatísticas sobre páginas e grupos de linhas do arquivo .parquet. Valor padrão: `true` Valores válidos: `true`, `false`, `y`, `n` Exemplo: `--s3-settings '{"EnableStatistics": false}'`
`TimestampColumnName`	Um parâmetro opcional para incluir uma coluna de timestamp nos dados de endpoint de destino do S3. O AWS DMS inclui uma coluna `STRING` adicional nos arquivos de objeto .parquet ou .csv dos dados migrados quando você define `TimestampColumnName` como um valor não vazio. Em uma carga completa, cada linha da coluna de timestamp contém um timestamp para quando os dados foram transferidos da origem para o destino pelo DMS. Em um carregamento de CDC, cada linha da coluna de timestamp contém o timestamp da confirmação da linha no banco de dados de origem. O formato de string para esse valor de coluna de timestamp é `yyyy-MM-dd HH:mm:ss.SSSSSS`. Por padrão, a precisão desse valor é em microssegundos. Para uma carga de CDC, o arredondamento da precisão depende do carimbo de data e hora de confirmação compatível com o DMS para o banco de dados de origem. Quando o parâmetro `AddColumnName` for definido como `true`, o DMS também inclui o nome da coluna de timestamp definida como o valor não vazio de `TimestampColumnName`. Exemplo: `--s3-settings '{"TimestampColumnName": "TIMESTAMP"}'`

Opção	Descrição
`UseTaskStartTimeForFullLoadTimestamp`	Quando definido como `true`, esse parâmetro utiliza a hora de início da tarefa como o valor da coluna de timestamp em vez da hora em que os dados são gravados no destino. Para carga completa, quando o `UseTaskStartTimeForFullLoadTimestamp` for definido como `true`, cada linha da coluna de carimbo de data/hora mostrará a hora de início da tarefa. Para cargas de CDC, cada linha da coluna de data e hora contém o tempo de confirmação da transação. Quando `UseTaskStartTimeForFullLoadTimestamp` está definido como `false`, o timestamp da carga máxima na coluna de timestamp é incrementado com data e hora em que os dados chegam ao destino. Valor padrão: `false` Valores válidos: `true`, `false` Exemplo: `--s3-settings '{"UseTaskStartTimeForFullLoadTimestamp": true}'` `UseTaskStartTimeForFullLoadTimestamp: true` ajuda a tornar `TimestampColumnName` do destino do S3 de uma carga máxima classificável com `TimestampColumnName` para uma carga de CDC.

Opção	Descrição
`ParquetTimestampInMillisecond`	Um parâmetro opcional que especifica a precisão de qualquer valor de coluna `TIMESTAMP` gravado em um arquivo de objeto do S3 no formato .parquet. Quando esse atributo é definido como `true` ou `y`, o AWS DMS grava todas as colunas `TIMESTAMP` em um arquivo com formato .parquet com precisão de milissegundos. Caso contrário, o DMS grava com precisão de microssegundos. No momento, Amazon Athena e AWS Glue só podem lidar com precisão de milissegundos para valores `TIMESTAMP` . Defina esse atributo como verdadeiro para arquivos de objeto do endpoint do S3 formatados como .parquet somente se quiser consultar ou processar os dados com o Athena ou o AWS Glue. **Note** • O AWS DMS grava qualquer valor de coluna `TIMESTAMP` gravado em um arquivo do S3 no formato .csv com precisão de microssegundos. • A configuração desse atributo não tem efeito sobre o formato da string do valor da coluna de timestamp inserido definindo o atributo `TimestampColumnName` . Valor padrão: `false` Valores válidos: `true`, `false`, `y`, `n` Exemplo: `--s3-settings '{"ParquetTimestampInMillisecond": true}'`

Opção	Descrição
`GlueCatalogGeneration`	Para gerar um AWS Glue Data Catalog, defina essa configuração de endpoint como `true`. Valor padrão: `false` Valores válidos: `true`, `false`. Exemplo: `--s3-settings '{"GlueCatalogGeneration": true}'` Observação: não utilize `GlueCatalogGeneration` com `PreserveTransactions` e `CdcPath`.

Utilizar o AWS Glue Data Catalog com um destino do Amazon S3 do AWS DMS

O AWS Glue é um serviço que fornece maneiras simples de categorizar dados e consiste em um repositório de metadados conhecido como AWS Glue Data Catalog. É possível integrar o AWS Glue Data Catalog ao endpoint de destino do Amazon S3 e consultar os dados do Amazon S3 por meio de outros serviços da AWS, como o Amazon Athena. O Amazon Redshift funciona com o AWS Glue, mas o AWS DMS não é compatível com isso como uma opção pré-criada.

Para gerar o catálogo de dados, defina a configuração `GlueCatalogGeneration` do endpoint como `true`, conforme mostrado no exemplo da AWS CLI a seguir.

```
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}'
```

Para uma tarefa de replicação de carga máxima que inclua o tipo de dados `csv`, defina `IncludeOpForFullLoad` como `true`.

Não utilize `GlueCatalogGeneration` com `PreserveTransactions` e `CdcPath`. O crawler do AWS Glue não pode reconciliar os diferentes esquemas de arquivos armazenados sob o `CdcPath` especificado.

Para que o Amazon Athena indexe os dados do Amazon S3, e para você consultar os dados utilizando consultas SQL padrão por meio do Amazon Athena, o perfil do IAM anexado ao endpoint deve ter a seguinte política:

```
{
    "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_*"
        }
    ]
}
```

Referências

- Para obter mais informações sobre o AWS Glue, consulte Conceitos no Guia do desenvolvedor do AWS Glue.
- Para obter mais informações sobre o AWS Glue Data Catalog, consulte Componentes no Guia do desenvolvedor do AWS Glue.

Utilizar criptografia de dados, arquivos de parquet e de CDC no destino do Amazon S3

É possível utilizar as configurações do endpoint de destino do S3 para configurar o seguinte:

- Uma chave do KMS personalizada para criptografar os objetos de destino do S3.
- Arquivos parquet como o formato de armazenamento para objetos de destino do S3.
- Captura de dados de alteração (CDC), incluindo a ordem de transações no destino do S3.
- Integre o AWS Glue Data Catalog com o endpoint de destino do Amazon S3 e consulte os dados do Amazon S3 por meio de outros serviços, como o Amazon Athena.

Configurações de chaves do AWS KMS para a criptografia de dados

Os exemplos a seguir mostram como configurar uma chave do KMS personalizada para criptografar os objetos de destino do S3. Para iniciar, execute o seguinte comando `create-endpoint` na CLI,

```
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"}'
```

Aqui, o objeto JSON especificado pela opção `--s3-settings` define dois parâmetros. Um é um parâmetro `EncryptionMode` com o valor `SSE_KMS`. O outro é um parâmetro `ServerSideEncryptionKmsKeyId` com o valor de `arn:aws:kms:us-east-1:111122223333:key/72abb6fb-1e49-4ac1-9aed-c803dfcc0480`. Esse valor é um Nome de recurso da Amazon (ARN) para a chave personalizada do KMS. Para um destino do S3, você também pode especificar configurações adicionais. Esses identificam o perfil de acesso ao servidor, fornecem delimitadores para o formato de armazenamento de objetos CSV padrão e fornecem o local e o nome do bucket para armazenar objetos de destino do S3.

Por padrão, a criptografia dos dados do S3 ocorre usando a criptografia do lado do servidor do S3. Para o destino do S3 do exemplo anterior, isso também é equivalente a especificar suas configurações de endpoint, como no exemplo a seguir.

```
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"}'
```

Para obter mais informações sobre como trabalhar com a criptografia do lado do servidor do S3, consulte Proteger dados utilizando a criptografia do lado do servidor.

> **Note**
>
> Também é possível utilizar o comando `modify-endpoint` na CLI para alterar o valor do parâmetro `EncryptionMode` para um endpoint existente de `SSE_KMS` para `SSE_S3`. Mas não é possível alterar o valor de `EncryptionMode` de `SSE_S3` para `SSE_KMS`.

Configurações para utilizar arquivos .parquet para armazenar objetos de destino do S3

O formato padrão para criar objetos de destino do S3 é arquivos .csv. Os exemplos a seguir mostram algumas configurações de endpoint para especificar arquivos .parquet como o formato para criar objetos de destino do S3. É possível especificar o formato .parquet de arquivos com todos os padrões, como no exemplo a seguir.

```
aws dms create-endpoint --endpoint-identifier s3-target-endpoint --engine-name s3 --
endpoint-type target
--s3-settings '{"ServiceAccessRoleArn": "your-service-access-ARN", "DataFormat":
 "parquet"}'
```

Aqui, o parâmetro `DataFormat` é definido como `parquet` para ativar o formato com todos os padrões do S3. Esses padrões incluem a codificação de um dicionário (`"EncodingType: "rle-dictionary"`) que utiliza uma combinação de empacotamento de bits e de codificação de run-length para armazenar valores repetidos.

É possível adicionar outras configurações para opções diferentes do padrão, como no exemplo a seguir.

```
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 }'
```

Aqui, além dos parâmetros para várias opções padrão de bucket do S3 e do parâmetro `DataFormat`, os seguintes parâmetros adicionais de arquivo .parquet são definidos:

- `EncodingType`: defina uma codificação de dicionário (`plain-dictionary`) que armazena os valores encontrados em cada coluna em um bloco por coluna da página do dicionário.
- `DictPageSizeLimit`: defina um tamanho máximo de página do dicionário de 3 MB.
- `EnableStatistics`: desativa o padrão que ativa a coleção de estatísticas sobre páginas de arquivos Parquet e grupos de linhas.

Captura de dados de alteração (CDC), incluindo a ordem de transações no destino do S3.

Por padrão, quando o AWS DMS executa uma tarefa de CDC, ele armazena todas as alterações de linha registradas em log no banco de dados de origem (ou bancos de dados) em um ou mais arquivos para cada tabela. Cada conjunto de arquivos que contêm alterações para a mesma tabela reside em um único diretório de destino associado a essa tabela. O AWS DMS cria tantos diretórios de destino quanto as tabelas de banco de dados migradas para o endpoint de destino do Amazon S3. Os arquivos são armazenados no destino do S3 nesses diretórios, independentemente da ordem da transação. Para obter mais informações sobre as convenções de nomenclatura, o conteúdo e o formato dos dados, consulte Utilizar o Amazon S3 como destino de dados do AWS Database Migration Service.

Para capturar as alterações do banco de dados de origem de uma maneira que também capture a ordem da transação, é possível especificar as configurações do endpoint do S3 que direcionam o AWS DMS a armazenar as alterações de linhas de todas as tabelas do banco de dados em um ou mais arquivos .csv criados, dependendo do tamanho da transação. Esses arquivos de transação .csv contêm todas as alterações de linhas listadas sequencialmente na ordem da transação de todas as tabelas envolvidas em cada transação. Esses arquivos de transação residem juntos em um único diretório de transações que você também especifica no destino do S3. Em cada arquivo de transação, a operação da transação e a identidade do banco de dados e a tabela de origem de cada alteração de linha são armazenadas como parte dos dados da linha, da seguinte maneira.

```
operation,table_name,database_schema_name,field_value,...
```

Aqui, *operation* é a operação da transação na linha alterada, *table_name* é o nome da tabela do banco de dados em que a linha foi alterada, *database_schema_name* é o nome do esquema do banco de dados em que a tabela reside e *field_value* é o primeiro de um ou mais valores de campo que especificam os dados da linha.

O exemplo a seguir de um arquivo de transação mostra as linhas alteradas de uma ou mais transações que envolvem duas tabelas.

```
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
```

Aqui, a operação da transação em cada linha é indicada por `I` (inserir), `U` (atualizar) ou `D` (excluir) na primeira coluna. O nome da tabela é o valor da segunda coluna (por exemplo, `Names_03cdcad11a`). O nome do esquema do banco de dados é o valor da terceira coluna (por exemplo, `rdsTempsdb`). E as colunas restantes são preenchidas com seus próprios dados da linha (por exemplo, `13,Daniel`).

Além disso, o AWS DMS nomeia os arquivos de transação que cria no destino do Amazon S3 utilizando um timestamp de acordo com a seguinte convenção de nomenclatura.

```
CDC_TXN-timestamp.csv
```

Aqui, *`timestamp`* é a hora em que o arquivo de transação foi criado, como no exemplo a seguir.

```
CDC_TXN-20201117153046033.csv
```

Esse timestamp no nome do arquivo garante que os arquivos de transação sejam criados e listados na ordem da transação quando você os lista no diretório de transações.

> **Note**
>
> Ao capturar alterações de dados na ordem da transação, o AWS DMS sempre armazena as alterações de linhas em arquivos .csv, independentemente do valor da configuração de `DataFormat` do S3 no destino. O AWS DMS não salva as alterações de dados na ordem da transação utilizando arquivos .parquet.

Para controlar a frequência de gravações em um destino do Amazon S3 durante uma tarefa de replicação de dados, é possível definir as configurações `CdcMaxBatchInterval` e `CdcMinFileSize`. Isso pode resultar em melhor desempenho ao analisar os dados sem operações adicionais de sobrecarga. Para obter mais informações, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS.

Como informar ao AWS DMS para armazenar todas as alterações de linhas na ordem da transação

1. Defina a configuração `PreserveTransactions` do S3 no destino como `true`.
2. Defina a configuração `CdcPath` do S3 no destino para um caminho de pasta relativo em que você deseja que o AWS DMS armazene os arquivos de transação .csv.

O AWS DMS cria esse caminho no bucket de destino e no diretório de trabalho padrão do S3 ou no bucket e na pasta do bucket e no bucket que você especifica utilizando as configurações `BucketName` e `BucketFolder` do S3 no destino.

Indicar operações de banco de dados de origem em dados migrados do S3

Quando o AWS DMS migra os registros para um destino do S3, ele pode criar um campo adicional em cada registro migrado. Esse campo adicional indica a operação aplicada ao registro no banco de dados de origem. A forma como o AWS DMS cria e define esse primeiro campo depende do tipo da tarefa de migração e das configurações de `includeOpForFullLoad`, `cdcInsertsOnly` e `cdcInsertsAndUpdates`.

Em uma carga máxima, quando `includeOpForFullLoad` é `true`, o AWS DMS sempre cria um primeiro campo adicional em cada registro .csv. Esse campo contém a letra I (INSERT) para indicar que a linha foi inserida no banco de dados de origem. Em uma carga de CDC, quando `cdcInsertsOnly` é `false` (o padrão), o AWS DMS também sempre cria um primeiro campo adicional em cada registro .csv ou .parquet. Esse campo contém a letra I (INSERT), U (UPDATE) ou D (DELETE) para indicar se a linha foi inserida, atualizada ou excluída no banco de dados de origem.

Na tabela a seguir, é possível ver como as configurações dos atributos `includeOpForFullLoad` e `cdcInsertsOnly` funcionam em conjunto e afetam a configuração dos registros migrados.

Com essas configurações de parâmetros		O DMS define os registros de destino para a saída .csv e .parquet da seguinte maneira	
includeOpForFullLoad	cdcInsertsOnly	Para carga completa	Para carga de CDC
`true`	`true`	O valor do primeiro campo definido como `I` é adicionado	O valor do primeiro campo definido como `I` é adicionado
`false`	`false`	Nenhum campo é adicionado	O valor do primeiro campo definido como `I`, `U` ou `D` é adicionado

Com essas configurações de parâmetros		O DMS define os registros de destino para a saída .csv e .parquet da seguinte maneira	
includeOpForFullLoad	cdcInsertsOnly	Para carga completa	Para carga de CDC
`false`	`true`	Nenhum campo é adicionado	Nenhum campo é adicionado
`true`	`false`	O valor do primeiro campo definido como `I` é adicionado	O valor do primeiro campo definido como `I`, `U` ou `D` é adicionado

Quando `includeOpForFullLoad` e `cdcInsertsOnly` forem definidos com o mesmo valor, os registros de destino serão definidos de acordo com o atributo que controla as configurações de registro para o tipo de migração atual. Esse atributo é `includeOpForFullLoad` para a carga máxima e `cdcInsertsOnly` para a carga da CDC.

Quando `includeOpForFullLoad` e `cdcInsertsOnly` estão definidos com valores diferentes, o AWS DMS torna as configurações dos registros de destino consistentes para a CDC e para a carga máxima. Isso é feito fazendo com que as configurações de registro de uma carga de CDC com as configurações de qualquer carga máxima anterior especificada por `includeOpForFullLoad` estejam em conformidade.

Ou seja, suponha que uma carga máxima está configurada para adicionar um primeiro campo para indicar um registro inserido. Nesse caso, um carregamento de CDC seguinte será configurado para adicionar um primeiro campo que indica um registro inserido, atualizado ou excluído de acordo com a origem. Por outro lado, suponha que um carga máxima está configurada para não adicionar um primeiro campo indicando um registro inserido. Nesse caso, uma carga de CDC também será definida para não adicionar um primeiro campo para cada registro, independentemente das operações de registro correspondentes na origem.

Da mesma forma, a maneira como o DMS cria e define um primeiro campo adicional depende das configurações de `includeOpForFullLoad` e de `cdcInsertsAndUpdates`. Na tabela a seguir, é possível ver como as configurações dos atributos `includeOpForFullLoad` e `cdcInsertsAndUpdates` funcionam em conjunto e afetam a configuração dos registros migrados neste formato.

Com essas configurações de parâmetros		O DMS define os registros de destino para a saída .csv da seguinte forma	
includeOpForFullLoad	cdcInsertsAndUpdates	Para carga completa	Para carga de CDC
`true`	`true`	O valor do primeiro campo definido como `I` é adicionado	O valor do primeiro campo definido como `I` ou `U`
`false`	`false`	Nenhum campo é adicionado	O valor do primeiro campo definido como `I`, `U` ou `D` é adicionado
`false`	`true`	Nenhum campo é adicionado	O valor do primeiro campo definido como `I` ou `U`
`true`	`false`	O valor do primeiro campo definido como `I` é adicionado	O valor do primeiro campo definido como `I`, `U` ou `D` é adicionado

Tipos de dados de destino do S3 Parquet

A tabela a seguir mostra os tipos de dados de destino do Parquet compatíveis ao utilizar o AWS DMS e o mapeamento padrão de tipos de dados do AWS DMS.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipo de dados do AWS DMS	Tipo de dados parquet do S3
`BYTES`	`BINARY`
`DATE`	`DATE32`

Tipo de dados do AWS DMS	Tipo de dados parquet do S3
TIME	TIME32
DATETIME	TIMESTAMP
INT1	INT8
INT2	INT16
INT4	INT32
INT8	INT64
NUMERIC	DECIMAL
REAL4	FLOAT
REAL8	DOUBLE
STRING	STRING
UINT1	UINT8
UINT2	UINT16
UINT4	UINT32
UINT8	UINT64
WSTRING	STRING
BLOB	BINARY
NCLOB	STRING
CLOB	STRING
BOOLEAN	BOOL

Utilizar um banco de dados Amazon DynamoDB como destino do AWS Database Migration Service

É possível utilizar o AWS DMS para migrar dados para uma tabela do Amazon DynamoDB. O Amazon DynamoDB é um serviço de banco de dados NoSQL totalmente gerenciado que fornece desempenho rápido e previsível com escalabilidade contínua. O AWS DMS é compatível com a utilização de um banco de dados relacional ou MongoDB como origem.

No DynamoDB, tabelas, itens e atributos são os componentes principais com que você trabalha. Uma tabela é uma coleção de itens, e cada item é uma coleção de atributos. O DynamoDB utiliza chaves primárias, chamadas chaves de partição, para identificar de modo exclusivo cada item em uma tabela. Também é possível utilizar chaves e índices secundários para fornecer mais flexibilidade de consulta.

O mapeamento de objeto é utilizado para migrar os dados de um banco de dados de origem para uma tabela de destino do DynamoDB. O mapeamento de objetos permite que você determine onde os dados de origem estão localizados no destino.

Quando o AWS DMS cria tabelas em um endpoint de destino do DynamoDB, ele cria o mesmo número de tabelas que está no endpoint do banco de dados de origem. O AWS DMS também define vários valores de parâmetros do DynamoDB. O custo de criação da tabela depende da quantidade de dados e do número de tabelas a serem migradas.

> **Note**
>
> A opção Modo SSL no console ou na API do AWS DMS não se aplica a alguns serviços de streaming de dados e do NoSQL, como o Kinesis e o DynamoDB. Eles são seguros por padrão, portanto, o AWS DMS mostra que a configuração do modo SSL é igual a nenhum (Modo SSL=nenhum). Não é necessário fornecer nenhuma configuração adicional para que o endpoint utilize o SSL. Por exemplo, ao utilizar o DynamoDB como um endpoint de destino, ele é seguro por padrão. Todas as chamadas de API para o DynamoDB utilizam SSL, portanto, não há necessidade de uma opção adicional de SSL no endpoint do AWS DMS. É possível inserir e recuperar dados com segurança por meio de endpoints SSL utilizando o protocolo HTTPS, que o AWS DMS utiliza por padrão ao se conectar a um banco de dados DynamoDB.

Para ajudar a aumentar a velocidade da transferência, o AWS DMS é compatível com uma carga máxima multithread para uma instância de destino do DynamoDB. O DMS oferece suporte a esse multithreading com configurações de tarefa que incluem o seguinte:

- `MaxFullLoadSubTasks`: utilize esta opção para indicar o número máximo de tabelas de origem a serem carregadas em paralelo. O DMS carrega cada tabela na tabela de destino do DynamoDB correspondente utilizando uma subtarefa dedicada. O valor padrão é 8. O valor máximo é 49.
- `ParallelLoadThreads`: utilize essa opção para especificar o número de threads que o AWS DMS utiliza para carregar cada tabela na tabela de destino do DynamoDB. O valor padrão é 0 (segmento único). O valor máximo é 200. Você pode solicitar o aumento desse limite máximo.

> **Note**
>
> O DMS atribui cada segmento de uma tabela ao seu próprio thread para carregar. Portanto, defina `ParallelLoadThreads` como o número máximo de segmentos que você especifica para uma tabela na origem.

- `ParallelLoadBufferSize`: utilize essa opção para especificar o número máximo de registros a serem armazenados em buffer utilizado pelos threads paralelos para carregar dados no destino do DynamoDB. O valor padrão é 50. Valor máximo de 1.000. Use essa configuração com `ParallelLoadThreads`; `ParallelLoadBufferSize` é válido somente quando há mais de um thread.
- Configurações de mapeamento de tabela para tabelas individuais: utilize as regras `table-settings` para identificar tabelas individuais da origem que você deseja carregar em paralelo. Além disso, utilize essas regras para especificar como segmentar as linhas de cada tabela para carregamento multithread. Para obter mais informações, consulte Regras e operações de configurações de tabelas e coleções.

> **Note**
>
> Quando o AWS DMS define valores de parâmetros do DynamoDB para uma tarefa de migração, o valor do parâmetro de Unidades de capacidade de leitura (RCU) padrão é definido como 200.
> O valor do parâmetro das Unidades de capacidade de gravação (WCU) também é definido, mas o valor depende de várias outras configurações:
>
> - O valor padrão do parâmetro WCU é 200.

- Se a configuração de tarefa `ParallelLoadThreads` for definida com um valor maior que 1 (o padrão é 0), o parâmetro WCU será definido como 200 vezes o valor de `ParallelLoadThreads`.
- As taxas de utilização padrão do AWS DMS se aplicam aos recursos que você utiliza.

Migração de um banco de dados relacional para uma tabela do DynamoDB

O AWS DMS é compatível com a migração de dados para os tipos de dados escalares do DynamoDB. Para migrar de um banco de dados relacional, como o Oracle ou o MySQL, para o DynamoDB, reestruture a forma como você armazena esses dados.

Atualmente, o AWS DMS é compatível com tabela única para a reestruturação de tabela única para os atributos do tipo escalar do DynamoDB. Se você estivesse migrando dados para o DynamoDB a partir de uma tabela de banco de dados relacional, pegaria dados de uma tabela e os reformataria como atributos de tipo de dados escalares. Esses atributos podem aceitar dados de várias colunas, e é possível mapear uma coluna diretamente para um atributo.

O AWS DMS é compatível com os seguintes tipos de dados escalares do DynamoDB:

- String
- Número
- Booleano

> **Note**
>
> Os dados NULL da origem são ignorados no destino.

Pré-requisitos para a utilização do DynamoDB como destino do AWS Database Migration Service

Antes de começar a trabalhar com um banco de dados DynamoDB como destino do AWS DMS, crie um perfil do IAM. Esse perfil do IAM deve permitir que o AWS DMS pressuponha e conceda acesso às tabelas do DynamoDB que estão sendo migradas. O conjunto mínimo de permissões de acesso é mostrado na seguinte política do IAM.

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

O perfil utilizado para a migração para o DynamoDB deve ter as seguintes permissões:

```
{
    "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": "*"
```

```
    }
    ]
}
```

Limitações ao utilizar o DynamoDB como destino do AWS Database Migration Service

Aplicam-se as seguintes limitações ao utilizar o DynamoDB como destino:

- O DynamoDB limita a precisão do tipo de dados Número a 38 locais. Armazene todos os tipos de dados com mais precisão como uma string. É necessário especificar isso explicitamente utilizando o recurso de mapeamento de objetos.
- Como o DynamoDB não tem um tipo de dados Date, os dados que utilizam esse tipo Date são convertidos em strings.
- O DynamoDB não permite atualizações nos atributos da chave primária. Essa restrição é importante quando se usa a replicação contínua com captura de dados de alterações (CDC), pois ela pode gerar dados indesejados no destino. Dependendo de como é o seu mapeamento de objetos, uma operação de CDC que atualiza a chave primária pode fazer uma de duas coisas. Pode falhar ou inserir um novo item com a chave primária atualizada e dados incompletos.
- O AWS DMS só é compatível com a replicação de tabelas com chaves primárias não compostas. A exceção é se você especificar um mapeamento de objetos para a tabela de destino com uma chave de partição personalizada ou chave de classificação, ou ambas.
- O AWS DMS não é compatível com dados LOB, a menos que ele seja um CLOB. O AWS DMS converte dados CLOB em uma string do DynamoDB ao migrar dados.
- Ao utilizar o DynamoDB como destino, somente a tabela de controle Aplicar exceções (`dmslogs.awsdms_apply_exceptions`) será compatível. Para obter mais informações sobre tabelas de controle, consulte Configurações de tarefa de tabela de controle.
- O AWS DMS não é compatível com a configuração da tarefa `TargetTablePrepMode=TRUNCATE_BEFORE_LOAD` do DynamoDB como destino.
- O AWS DMS não é compatível com a configuração da tarefa `TaskRecoveryTableEnabled` do DynamoDB como destino.

Utilizar o mapeamento de objetos para migrar dados para o DynamoDB

O AWS DMS utiliza regras de mapeamento de tabela para mapear dados da origem para a tabela de destino do DynamoDB. Para mapear dados para um destino do DynamoDB, utilize um tipo de regra de mapeamento de tabela chamada object-mapping. O mapeamento de objetos permite definir

os nomes de atributos e os dados a serem migrados para eles. Você deve ter regras de seleção ao utilizar o mapeamento de objetos.

O DynamoDB não tem uma estrutura predefinida além de ter uma chave de partição e uma chave de classificação opcional. Se você tiver uma chave primária não composta, o AWS DMS a utilizará. Se você tiver uma chave primária composta ou quiser utilizar uma chave de classificação, defina as chaves e os outros atributos na tabela de destino do DynamoDB.

Para criar uma regra de mapeamento de objetos, especifique `rule-type` como object-mapping. Essa regra especifica o tipo de mapeamento de objeto que você deseja usar.

A estrutura da regra é a seguinte:

```
{ "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>"
    }
  ]
}
```

Atualmente, o AWS DMS oferece suporte a `map-record-to-record` e `map-record-to-document` como os únicos valores válidos para o parâmetro `rule-action`. Esses valores especificam o que o AWS DMS faz por padrão com registros que não são excluídos como parte da lista de atributos `exclude-columns`. Esses valores não afetam os mapeamentos de atributos de forma alguma.

- É possível utilizar `map-record-to-record` ao migrar de um banco de dados relacional para o DynamoDB. Ele utiliza a chave primária do banco de dados relacional como a chave de partição no DynamoDB e cria um atributo para cada coluna no banco de dados de origem. Ao utilizar `map-record-to-record` para qualquer coluna na tabela de origem que não aparece na lista de atributos `exclude-columns`, o AWS DMS cria um atributo correspondente na instância de destino do DynamoDB. Isso é feito, independentemente de a coluna de origem ser utilizada ou não em um mapeamento de atributos.

- Você utiliza `map-record-to-document` para colocar colunas de origem em um único mapa sem formatação do DynamoDB no destino, utilizando o nome de atributo "_doc". Ao utilizar `map-record-to-document`, o AWS DMS coloca os dados em um único atributo de mapa sem formatação do DynamoDB na origem. Esse atributo é chamado "_doc". Esse posicionamento se aplica a qualquer coluna na tabela de origem que não aparece na lista de atributos `exclude-columns`.

Uma maneira de compreender a diferença entre os parâmetros de `rule-action`, `map-record-to-record` e `map-record-to-document`, é ver os dois parâmetros em ação. Para este exemplo, suponha que você está começando com uma linha de tabela de banco de dados relacional com a seguinte estrutura e dados:

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

Para migrar essas informações para o DynamoDB, você criaria regras para mapear os dados para um item de tabela do DynamoDB. Observe as colunas listadas para o parâmetro `exclude-columns`. Essas colunas não são mapeadas diretamente para o destino. Em vez disso, o mapeamento de atributo é utilizado para combinar os dados em novos itens, como quando FirstName e LastName são agrupados para se tornarem CustomerName no destino do DynamoDB. NickName e renda não são excluídos.

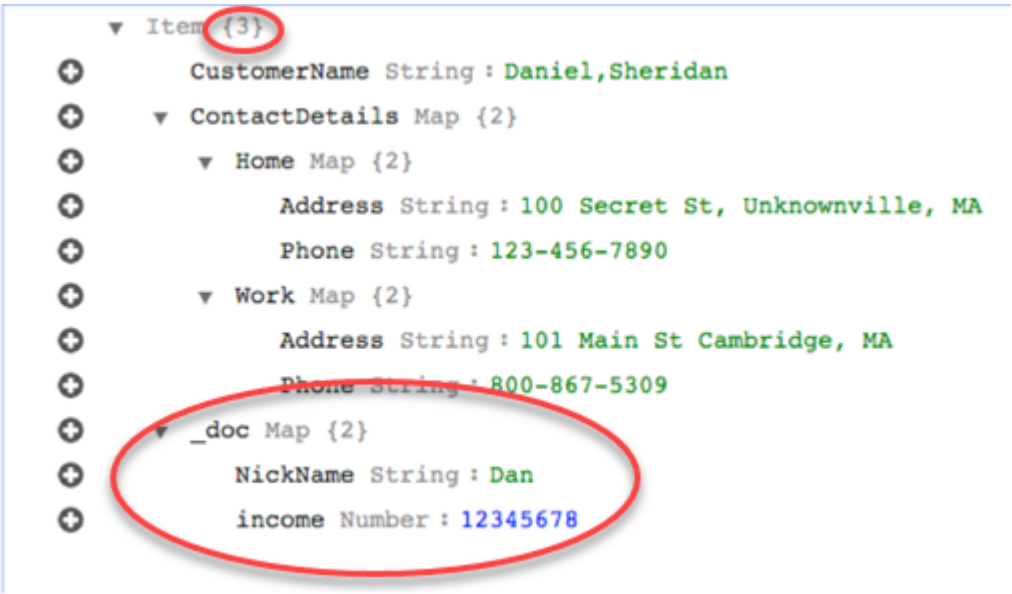
```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "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}"
```

```
                                    }
                                }
                            }
                        }
                    }
                }
            ]
        }
    }
]
}
```

Ao utilizar o parâmetro `rule-action` map-record-to-record, os dados de NickName e renda são mapeados para itens com o mesmo nome no destino do DynamoDB.

No entanto, suponhamos que você utilize as mesmas regras, mas altere o parâmetro `rule-action` para map-record-to-document. Nesse caso, as colunas não listadas no parâmetro `exclude-columns`, NickName e income, serão mapeadas para um item _doc.



Utilizar expressões de condição personalizadas com o mapeamento de objetos

É possível utilizar um recurso do DynamoDB chamado de expressões condicionais para manipular dados que estão sendo gravados em uma tabela do DynamoDB. Para obter mais informações sobre expressões condicionais no DynamoDB, consulte Expressões de condição.

Um membro de expressão condicional consiste em:

- uma expressão (obrigatória)
- valores de atributo de expressão (opcional). Especifica uma estrutura json do DynamoDB do valor de atributo
- nomes de atributo de expressão (opcional)
- opções para quando utilizar a expressão condicional (opcional). O padrão é apply-during-cdc = falso e apply-during-full-load = verdadeiro

A estrutura da regra é a seguinte:

```
"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>
        }
```

O exemplo a seguir destaca as seções utilizadas para expressão condicional.


```
{
  "rules": [
    {
      "rule-type": "object-mapping",
      "rule-id": "1",
      "rule-name": "TransformToDDB",
      "rule-action": "map-record-to-record",
      "object-locator": {
        "schema-name": "test",
        "table-name": "customer",

      },
      "target-table-name": "customer_t",
      "mapping-parameters": {
        "partition-key-name": "CustomerName",
        "condition-expression": {
          "expression":"attribute_not_exists(version) or version <= :record_version",
          "expression-attribute-values": [
              {
                "name":":record_version",
                "value":{"N":"${version}"}
              }
          ],
          "apply-during-cdc":true,
          "apply-during-full-load": true
        }
        "attribute-mappings": [
          {
            "target-attribute-name": "CustomerName",
            "attribute-type": "scalar",
            "attribute-sub-type": "string",
            "value": "${FirstName},${LastName}"
          }
        ]
      }
    }
  ]
}
```

Usar o mapeamento de atributo com mapeamento de objeto

O mapeamento de atributo permite especificar uma string de modelo utilizando nomes de colunas de origem para reestruturar dados no destino. Não há formatação feita além do que o usuário especifica no modelo.

O exemplo a seguir mostra a estrutura do banco de dados de origem e a estrutura desejada do destino do DynamoDB. Primeiramente, é mostrada a estrutura da origem: nesse caso um banco de

dados Oracle, e, depois, a estrutura desejada dos dados no DynamoDB. O exemplo termina com o JSON utilizado para criar a estrutura de destino desejada.

A estrutura dos dados do Oracle é a seguinte:

First	Las	St	Home/ ss	HomeF	WorkAddre ss	Work	DateOfBirth
Chave primária				N/D			
Ran	Mar	5	221B Baker Street	123456 0	31 Spooner Street, Quahog	9876 0	29/02/1988

A estrutura dos dados do DynamoDB é a seguinte:

Custor ame	StoreId	ContactDetails	DateOfBirth
Chave de partição	Chave de classific ação	N/D	
Randy sh	5	{ "Name": "Randy", "Home": { "Address": "221B Baker Street", "Phone": 1234567890 }, "Work": { "Address": "31 Spooner Street, Quahog", "Phone": 9876541230 }	02/29/1988

Custor ame	StoreId	ContactDetails	DateOfBirth
		}	

O JSON a seguir mostra o mapeamento de objetos e o mapeamento de colunas utilizados para alcançar a estrutura do DynamoDB:

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "TransformToDDB",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "test",
                "table-name": "customer"
            },
            "target-table-name": "customer_t",
            "mapping-parameters": {
                "partition-key-name": "CustomerName",
                "sort-key-name": "StoreId",
                "exclude-columns": [
                    "FirstName",
                    "LastName",
                    "HomeAddress",
                    "HomePhone",
                    "WorkAddress",
                    "WorkPhone"
                ],
```

```
                "attribute-mappings": [
                    {
                        "target-attribute-name": "CustomerName",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${FirstName},${LastName}"
                    },
                    {
                        "target-attribute-name": "StoreId",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${StoreId}"
                    },
                    {
                        "target-attribute-name": "ContactDetails",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "{\"Name\":\"${FirstName}\",\"Home\":{\"Address
\":\"${HomeAddress}\",\"Phone\":\"${HomePhone}\"}, \"Work\":{\"Address\":
\"${WorkAddress}\",\"Phone\":\"${WorkPhone}\"}}"
                    }
                ]
            }
        }
    ]
}
```

Outra maneira de utilizar o mapeamento de colunas é utilizar o formato do DynamoDB como o tipo de documento. O código a seguir utiliza dynamodb-map como `attribute-sub-type` para o mapeamento de atributo.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
```

```
            "rule-action": "include"
        },
        {
            "rule-type": "object-mapping",
            "rule-id": "2",
            "rule-name": "TransformToDDB",
            "rule-action": "map-record-to-record",
            "object-locator": {
                "schema-name": "test",
                "table-name": "customer"
            },
            "target-table-name": "customer_t",
            "mapping-parameters": {
                "partition-key-name": "CustomerName",
                "sort-key-name": "StoreId",
                "exclude-columns": [
                    "FirstName",
                    "LastName",
                    "HomeAddress",
                    "HomePhone",
                    "WorkAddress",
                    "WorkPhone"
                ],
                "attribute-mappings": [
                    {
                        "target-attribute-name": "CustomerName",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${FirstName},${LastName}"
                    },
                    {
                        "target-attribute-name": "StoreId",
                        "attribute-type": "scalar",
                        "attribute-sub-type": "string",
                        "value": "${StoreId}"
                    },
                    {
                        "target-attribute-name": "ContactDetails",
                        "attribute-type": "document",
                        "attribute-sub-type": "dynamodb-map",
                        "value": {
                          "M": {
                            "Name": {
                              "S": "${FirstName}"
```

```
                            },
                            "Home": {
                                    "M": {
                                        "Address": {
                                            "S": "${HomeAddress}"
                                        },
                                        "Phone": {
                                            "S": "${HomePhone}"
                                        }
                                    }
                                },
                                "Work": {
                                    "M": {
                                        "Address": {
                                            "S": "${WorkAddress}"
                                        },
                                        "Phone": {
                                            "S": "${WorkPhone}"
                                        }
                                    }
                                }
                            }
                        }
                    ]
                }
            }
        ]
}
```

Como alternativa a `dynamodb-map`, é possível utilizar `dynamodb-list` como o subtipo de atributo para mapeamento de atributos, conforme mostrado no exemplo a seguir.

```
{
"target-attribute-name": "ContactDetailsList",
"attribute-type": "document",
"attribute-sub-type": "dynamodb-list",
"value": {
    "L": [
            {
                "N": "${FirstName}"
```

```
            },
            {
                "N": "${HomeAddress}"
            },
            {
                "N": "${HomePhone}"
            },
            {
                "N": "${WorkAddress}"
            },
            {
                "N": "${WorkPhone}"
            }
        ]
    }
}
```

Exemplo 1: Usar o mapeamento de atributo com mapeamento de objeto

O exemplo a seguir migra dados de duas tabelas de banco de dados MySQL, nfl_data e sport_team, para duas tabelas do DynamoDBs chamadas NFLTeams e SportTeams. A estrutura das tabelas e do JSON utilizados para mapear os dados das tabelas do banco de dados MySQL para as tabelas do DynamoDB é mostrada a seguir.

A estrutura da tabela do banco de dados MySQL nfl_data é mostrada abaixo:

```
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    |       |
+---------------+-------------+------+-----+---------+-------+
```

A estrutura do banco de dados MySQL sport_team é mostrada abaixo:

```
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    |                |
```

As regras de mapeamento de tabela utilizadas para mapear as duas tabelas para as duas tabelas do DynamoDB são mostradas abaixo:

```
{
  "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}\"}"
          }
        ]
      }
    }
  ]
}
```

A amostra de saída da tabela NFLTeams do DynamoDB é mostrada abaixo:

```
 "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"
}
```

A amostra de saída da tabela SportsTeams do DynamoDB é mostrada abaixo:

```
{
  "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"
}
```

Tipos de dados de destino do DynamoDB

O endpoint do DynamoDB do AWS DMS é compatível com a maioria dos tipos de dados do DynamoDB. A tabela a seguir mostra os tipos de dados de destino do Amazon AWS DMS compatíveis com o AWS DMS e o mapeamento padrão relativo aos tipos de dados do AWS DMS.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Quando o AWS DMS migra dados de bancos de dados heterogêneos, mapeamos os tipos de dados do banco de dados de origem para tipos de dados intermediários chamados de tipos de dados AWS DMS. Em seguida, mapeamos os tipos de dados intermediários para os tipos de dados de destino.

A tabela a seguir mostra cada tipo de dados do AWS DMS e o tipo de dados que ele mapeia no DynamoDB:

Tipo de dados do AWS DMS	Tipo de dados do DynamoDB
String	String
WString	String
Booleano	Booleano
Data	String
DateTime	String
INT1	Número
INT2	Número
INT4	Número
INT8	Número
Numérico	Número
Real4	Número
Real8	Número
UINT1	Número
UINT2	Número
UINT4	Número
UINT8	Número
CLOB	String

Usando o Amazon Kinesis Data Streams como alvo para AWS Database Migration Service

Você pode usar AWS DMS para migrar dados para um stream de dados do Amazon Kinesis. O Amazon Kinesis Data Streams faz parte do serviço Amazon Kinesis Data Streams. É possível utilizar os fluxos de dados do Kinesis para coletar e processar grandes registros de dados em tempo real.

O fluxo de dados do Kinesis é composto por fragmentos. O estilhaço é uma sequência de registros de dados identificada de forma exclusiva em um stream. Para obter mais informações sobre fragmentos no Amazon Kinesis Data Streams, consulte o Fragmento no Guia do desenvolvedor do Amazon Kinesis Data Streams.

AWS Database Migration Service publica registros em um stream de dados do Kinesis usando JSON. Durante a conversão, o AWS DMS serializa cada registro do banco de dados de origem em um par atributo/valor no formato JSON ou em um formato de mensagem JSON_UNFORMATTED. Um formato de mensagem JSON_UNFORMATTED é uma string JSON de linha única com novo delimitador de linha. Ele permite que o Amazon Data Firehose entregue dados do Kinesis para um destino do Amazon S3 e, em seguida, consulte-os usando vários mecanismos de consulta, incluindo o Amazon Athena.

O mapeamento de objetos é utilizado para migrar os dados de uma fonte de dados compatível para um fluxo de destino. Com o mapeamento do objeto, você determina como estruturar os registros de dados no stream. Também é possível definir uma chave de partição para cada tabela, que será utilizada pelo Kinesis Data Streams para agrupar os dados em fragmentos.

Ao AWS DMS criar tabelas em um endpoint de destino do Kinesis Data Streams, ele cria tantas tabelas quanto no endpoint do banco de dados de origem. AWS DMS também define vários valores de parâmetros do Kinesis Data Streams. O custo de criação da tabela depende da quantidade de dados e do número de tabelas a serem migradas.

> **Note**
>
> A opção Modo SSL no AWS DMS console ou na API não se aplica a alguns serviços de streaming de dados e NoSQL, como Kinesis e DynamoDB. Eles são seguros por padrão, então AWS DMS mostra que a configuração do modo SSL é igual a nenhuma (Modo SSL = Nenhuma). Não é necessário fornecer nenhuma configuração adicional para que o endpoint utilize o SSL. Por exemplo, ao utilizar o Kinesis como um endpoint de destino, ele é seguro por padrão. Todas as chamadas de API para o Kinesis usam SSL, portanto, não há necessidade de uma opção adicional de SSL no endpoint. AWS DMS É possível colocar e

recuperar dados com segurança por meio de endpoints SSL utilizando o protocolo HTTPS, que o AWS DMS utiliza por padrão ao se conectar a um fluxo de dados Kinesis.

Configurações do endpoint do Kinesis Data Streams

Ao usar os endpoints de destino do Kinesis Data Streams, você pode obter detalhes da transação e do controle `KinesisSettings` usando a opção na API. AWS DMS

É possível definir as configurações da conexão de uma das seguintes maneiras:

- No AWS DMS console, usando as configurações do endpoint.
- Na CLI, usando a `kinesis-settings` opção do CreateEndpointcomando.

Na CLI, utilize os seguintes parâmetros de solicitação da opção `kinesis-settings`:

> **Note**
>
> Compatibilidade com a configuração do endpoints do `IncludeNullAndEmpty` está disponível nas versões 3.4.1 e superiores do AWS DMS . Mas o suporte para as outras configurações de endpoint a seguir para destinos do Kinesis Data Streams está disponível em. AWS DMS

- `MessageFormat`: o formato de saída dos registros criados no endpoint. O formato da mensagem é `JSON` (padrão) ou `JSON_UNFORMATTED` (uma única linha sem guia).
- `IncludeControlDetails`: mostra informações detalhadas de controle para definição de tabelas, definição de colunas e alterações de tabelas e colunas na saída de mensagem do Kinesis. O padrão é `false`.
- `IncludeNullAndEmpty`: inclui colunas NULL e vazias no destino. O padrão é `false`.
- `IncludePartitionValue`: mostra o valor da partição na saída da mensagem do Kinesis, a menos que o tipo de partição seja `schema-table-type`. O padrão é `false`.
- `IncludeTableAlterOperations`: inclui todas as operações da linguagem de definição de dados (DDL) que alteram a tabela nos dados de controle, como `rename-table`, `drop-table`, `add-column`, `drop-column` e `rename-column`. O padrão é `false`.
- `IncludeTransactionDetails`: fornece informações detalhadas sobre transações do banco de dados de origem. Essas informações incluem um timestamp de confirmação, uma posição no log

e valores para `transaction_id`, `previous_transaction_id` e `transaction_record_id` (o deslocamento de registro dentro de uma transação). O padrão é `false`.

- `PartitionIncludeSchemaTable`: prefixa os nomes de esquema e de tabela em valores de partições, quando o tipo de partição for `primary-key-type`. Isso aumenta a distribuição de dados entre estilhaços do Kinesis. Por exemplo, suponha que um esquema `SysBench` tenha milhares de tabelas, e cada tabela tenha apenas um intervalo limitado para uma chave primária. Nesse caso, a mesma chave primária é enviada de milhares de tabelas para o mesmo estilhaço, o que provoca o controle de utilização. O padrão é `false`.

O exemplo a seguir mostra a opção `kinesis-settings` em uso com um exemplo de comando `create-endpoint` emitido utilizando a AWS CLI.

```
aws dms create-endpoint --endpoint-identifier=$target_name --engine-name kinesis --
endpoint-type target
--region us-east-1 --kinesis-settings
 ServiceAccessRoleArn=arn:aws:iam::333333333333:role/dms-kinesis-role,
StreamArn=arn:aws:kinesis:us-east-1:333333333333:stream/dms-kinesis-target-
doc,MessageFormat=json-unformatted,
IncludeControlDetails=true,IncludeTransactionDetails=true,IncludePartitionValue=true,PartitionI
IncludeTableAlterOperations=true
```

Configurações da tarefa de carga máxima com vários threads

Para ajudar a aumentar a velocidade da transferência, AWS DMS oferece suporte a uma carga completa multisegmentada em uma instância de destino do Kinesis Data Streams. O DMS oferece suporte a esse multithreading com configurações de tarefa que incluem o seguinte:

- `MaxFullLoadSubTasks`: utilize esta opção para indicar o número máximo de tabelas de origem a serem carregadas em paralelo. O DMS carrega cada tabela na tabela de destino do Kinesis correspondente utilizando uma subtarefa dedicada. O padrão é 8; o valor máximo é 49.
- `ParallelLoadThreads`— Use essa opção para especificar o número de threads AWS DMS usados para carregar cada tabela em sua tabela de destino do Kinesis. O valor máximo para um destino do Kinesis Data Streams é 32. Você pode solicitar o aumento desse limite máximo.
- `ParallelLoadBufferSize`: utilize essa opção para especificar o número máximo de registros a serem armazenados em buffer utilizado pelos threads de carga paralela para carregar dados no destino do Kinesis. O valor padrão é 50. Valor máximo de 1.000. Use essa configuração com `ParallelLoadThreads`; `ParallelLoadBufferSize` é válido somente quando há mais de um thread.

- `ParallelLoadQueuesPerThread`: utilize esta opção para especificar o número de filas que cada thread simultâneo acessa para extrair registros de dados das filas e gerar uma carga em lote para o destino. O padrão é um. No entanto, para destinos do Kinesis de vários tamanhos de carga útil, o intervalo válido é de 5 a 512 filas por thread.

Configurações da tarefa de carga de CDC multithread

É possível melhorar o desempenho da captura de dados de alterações (CDC) para endpoints de destino de streaming de dados em tempo real, como o Kinesis, utilizando configurações de tarefa para modificar o comportamento da chamada da API `PutRecords`. Para fazer isso, especifique o número de threads simultâneos, filas por thread e o número de registros a serem armazenados em um buffer usando as configurações da tarefa `ParallelApply*`. Por exemplo, suponha que você queira executar um carregamento de CDC e aplicar 128 threads em paralelo. Você também quer acessar 64 filas por thread, com 50 registros armazenados por buffer.

Para promover o desempenho do CDC, AWS DMS oferece suporte a estas configurações de tarefas:

- `ParallelApplyThreads`— Especifica o número de threads simultâneos que são AWS DMS usados durante o carregamento do CDC para enviar registros de dados para um endpoint de destino do Kinesis. O valor padrão é zero (0) e o valor máximo é 32.
- `ParallelApplyBufferSize`: especifica o número máximo de registros a serem armazenados em cada fila de buffer para threads simultâneos enviarem para um endpoint de destino do Kinesis durante uma carga de CDC. O valor padrão é 100 e o valor máximo é 1.000. Use essa opção quando `ParallelApplyThreads` especificar mais de um thread.
- `ParallelApplyQueuesPerThread`: especifica o número de filas que cada thread acessa para extrair registros de dados das filas e gerar uma carga em lote para um endpoint do Kinesis durante a CDC. O valor padrão é 1 e o valor máximo é 512.

Ao usar configurações da tarefa `ParallelApply*`, o `partition-key-type` padrão é a `primary-key` da tabela, não o `schema-name.table-name`.

Utilizar uma imagem anterior para visualizar valores originais de linhas da CDC para um fluxo de dados do Kinesis como destino

Ao gravar atualizações da CDC em um destino de streaming de dados, como o Kinesis, é possível visualizar os valores originais de linhas do banco de dados de origem antes da alteração por uma

atualização. Para tornar isso possível, AWS DMS preenche uma imagem anterior dos eventos de atualização com base nos dados fornecidos pelo mecanismo do banco de dados de origem.

Diferentes mecanismos de banco de dados de origem fornecem diferentes quantidades de informações para uma imagem anterior:

- O Oracle fornece atualizações para colunas somente se elas forem alteradas.
- O PostgreSQL fornece somente dados para colunas que fazem parte da chave primária (alterada ou não). Para fornecer dados para todas as colunas (alteradas ou não), você precisa definir `REPLICA_IDENTITY` como `FULL` em vez de `DEFAULT`. Observe que você deve escolher cuidadosamente a configuração `REPLICA_IDENTITY` para cada tabela. Se você definir `REPLICA_IDENTITY` como `FULL`, todos os valores da coluna serão gravados continuamente no registro em log de gravação antecipada (WAL). Isso pode causar problemas de desempenho ou de recursos com tabelas que são atualizadas com frequência.
- O MySQL geralmente fornece dados para todas as colunas, exceto para os tipos de dados BLOB e CLOB (alterados ou não).

Para habilitar a criação de imagem anterior para adicionar valores originais do banco de dados de origem à saída do AWS DMS , use a configuração de tarefa `BeforeImageSettings` ou o parâmetro `add-before-image-columns`. Esse parâmetro aplica uma regra de transformação de coluna.

`BeforeImageSettings` adiciona um novo atributo JSON a cada operação de atualização com valores coletados do sistema de banco de dados de origem, conforme mostrado a seguir.

```
"BeforeImageSettings": {
    "EnableBeforeImage": boolean,
    "FieldName": string,
    "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

> **Note**
>
> Aplique somente `BeforeImageSettings` às AWS DMS tarefas que contêm um componente do CDC, como carga total mais tarefas do CDC (que migram dados existentes e replicam as alterações em andamento), ou às tarefas somente do CDC (que replicam

somente as alterações de dados). Não aplique `BeforeImageSettings` a tarefas que são somente de carga total.

Para opções `BeforeImageSettings`, aplica-se o seguinte:

- Defina a opção `EnableBeforeImage` como `true` para habilitar a criação de imagem anterior. O padrão é `false`.
- Use a opção `FieldName` para atribuir um nome ao novo atributo JSON. Quando `EnableBeforeImage` for `true`, `FieldName` será necessário e não poderá estar vazio.
- A opção `ColumnFilter` especifica uma coluna a ser adicionada usando imagem anterior. Para adicionar somente colunas que fazem parte das chaves primárias da tabela, use o valor padrão, `pk-only`. Para adicionar qualquer coluna que tenha um valor de imagem anterior, use `all`. Observe que a imagem anterior não contém colunas com tipos de dados LOB, como CLOB ou BLOB.

```
"BeforeImageSettings": {
    "EnableBeforeImage": true,
    "FieldName": "before-image",
    "ColumnFilter": "pk-only"
  }
```

> **Note**
>
> Os destinos do Amazon S3 não são compatíveis com `BeforeImageSettings`. Para destinos do S3, utilize somente a regra de transformação `add-before-image-columns` para executar a criação da imagem anterior durante a CDC.

Usar uma regra de transformação de imagem anterior

Como alternativa às configurações de tarefa, é possível usar o parâmetro `add-before-image-columns`, que aplica uma regra de transformação de coluna. Com esse parâmetro, é possível ativar a imagem anterior durante a CDC em destinos de streaming de dados, como o Kinesis.

Usando `add-before-image-columns` em uma regra de transformação, é possível aplicar um controle mais refinado dos resultados da imagem anterior. As regras de transformação permitem que

você use um localizador de objetos que oferece controle sobre as tabelas selecionadas para a regra. Além disso, é possível encadear regras de transformação, o que permite que regras diferentes sejam aplicadas a tabelas diferentes. Depois, você poderá manipular as colunas produzidas usando outras regras.

> **Note**
>
> Não use o parâmetro `add-before-image-columns` junto com a configuração da tarefa `BeforeImageSettings` na mesma tarefa. Em vez disso, use o parâmetro ou a configuração, mas não ambos, para uma única tarefa.

Um tipo de regra `transformation` com o parâmetro `add-before-image-columns` de uma coluna deve fornecer uma seção `before-image-def`. Por exemplo:

```
{
  "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,
  }
}
```

O valor de `column-prefix` precede um nome de coluna e o valor padrão de `column-prefix` é `BI_`. O valor de `column-suffix` é anexado ao nome da coluna e o padrão é vazio. Não defina `column-prefix` e `column-suffix` como strings vazias.

Escolha um valor para `column-filter`. Para adicionar somente colunas que fazem parte das chaves primárias da tabela, escolha `pk-only`. Escolha `non-lob` para adicionar somente colunas que não sejam do tipo LOB. Ou escolha `all` para adicionar qualquer coluna que tenha um valor de imagem anterior.

Exemplo de uma regra de transformação de imagem anterior

A regra de transformação no exemplo a seguir adiciona uma nova coluna chamada `BI_emp_no` no destino. Portanto, uma instrução como `UPDATE employees SET emp_no = 3 WHERE emp_no =`

1; preenche o campo BI_emp_no com 1. Ao gravar atualizações da CDC em destinos do Amazon S3, a coluna BI_emp_no possibilita identificar qual linha original foi atualizada.

```
{
  "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"
      }
    }
  ]
}
```

Para obter informações sobre como usar a ação da regra add-before-image-columns, consulte Regras de transformação e ações.

Pré-requisitos para usar um stream de dados do Kinesis como destino para AWS Database Migration Service

Função do IAM para usar um stream de dados do Kinesis como destino para AWS Database Migration Service

Antes de configurar um stream de dados do Kinesis como destino AWS DMS, certifique-se de criar uma função do IAM. Essa função deve permitir AWS DMS assumir e conceder acesso aos fluxos de dados do Kinesis para os quais estão sendo migrados. O conjunto mínimo de permissões de acesso é mostrado na seguinte política do IAM.

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

O perfil que você utiliza para a migração para um fluxo de dados do Kinesis deve ter as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:PutRecord",
                "kinesis:PutRecords"
```

```
            ],
            "Resource": "arn:aws:kinesis:region:accountID:stream/streamName"
        }
    ]
}
```

Acessando um stream de dados do Kinesis como destino para AWS Database Migration Service

Na AWS DMS versão 3.4.7 e superior, para se conectar a um endpoint do Kinesis, você deve fazer o seguinte:

- Configure o DMS para usar endpoints da VPC. Para obter mais informações sobre a configuração de endpoints da VPC, consulte Configurar endpoints da VPC como endpoints de origem e de destino do AWS.
- Configure o DMS para usar rotas públicas; ou seja, torne a instância de replicação pública. Para obter mais informações sobre instâncias de replicação públicas, consulte Instâncias de replicação públicas e privadas.

Limitações ao usar o Kinesis Data Streams como destino para AWS Database Migration Service

Aplicam-se as seguintes limitações ao utilizar o Kinesis Data Streams como destino:

- AWS DMS publica cada atualização em um único registro no banco de dados de origem como um registro de dados em um determinado stream de dados do Kinesis, independentemente das transações. No entanto, é possível incluir detalhes da transação para cada registro de dados utilizando parâmetros relevantes da API do `KinesisSettings`.
- O modo Full LOB não é compatível.
- O tamanho máximo do LOB compatível é 1 MB.
- Os Kinesis Data Streams não é compatível com a desduplicação. As aplicações que consumem dados de um fluxo precisam tratar os registros duplicados. Para obter mais informações, consulte Tratar registros duplicados no Guia do desenvolvedor do Amazon Kinesis Data Streams.
- AWS DMS suporta as duas formas a seguir para chaves de partição:
  - `SchemaName.TableName`: uma combinação de esquema e nome da tabela.
  - `${AttributeName}`: o valor de um dos campos no JSON ou a chave primária da tabela no banco de dados de origem.

- Para obter informações sobre como criptografar dados em repouso no Kinesis Data Streams, consulte Proteção de dados no Kinesis Data Streams, no Guia do desenvolvedor do AWS Key Management Service .
- O `BatchApply` não é compatível com um endpoint do Kinesis. A utilização da aplicação em lote (por exemplo, a configuração `BatchApplyEnabled` da tarefa de metadados de destino) para um destino do Kafka pode resultar em perda de dados.
- Os destinos do Kinesis só são compatíveis com um stream de dados do Kinesis na mesma AWS conta e na Região da AWS mesma instância de replicação.
- Ao migrar de uma fonte MySQL, os dados não incluem BeforeImage os tipos de dados CLOB e BLOB. Para ter mais informações, consulte Utilizar uma imagem anterior para visualizar valores originais de linhas da CDC para um fluxo de dados do Kinesis como destino.
- AWS DMS não suporta a migração de valores do tipo de `BigInt` dados com mais de 16 dígitos. Para contornar essa limitação, você pode usar a regra de transformação a seguir para converter a coluna `BigInt` em uma string. Para obter mais informações sobre regras transformação, consulte Regras de transformação e ações.

```
{
    "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
    }
}
```

Utilizar o mapeamento de objetos para migrar dados para um fluxo de dados do Kinesis

AWS DMS usa regras de mapeamento de tabelas para mapear dados da fonte para o stream de dados do Kinesis de destino. Para mapear dados para um fluxo de destino, utilize um tipo de regra

de mapeamento de tabela chamado mapeamento de objetos. Utilize o mapeamento de objetos para definir como os registros de dados na origem são mapeados para os registros de dados publicados para o fluxo de dados do Kinesis.

O fluxo de dados do Kinesis não tem estrutura predefinida além de uma chave de partição. Em uma regra de mapeamento de objeto, os valores possíveis de um `partition-key-type` para registros de dados são `schema-table`, `transaction-id`, `primary-key` `constant` e `attribute-name`.

Para criar uma regra de mapeamento de objetos, especifique `rule-type` como `object-mapping`. Essa regra especifica o tipo de mapeamento de objeto que você deseja usar.

A estrutura da regra é a seguinte:

```
{
    "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 atualmente suporta `map-record-to-record` e `map-record-to-document` como os únicos valores válidos para o `rule-action` parâmetro. Essas configurações afetam valores que não são excluídos como parte da lista de atributos `exclude-columns`. Os `map-record-to-document` valores `map-record-to-record` e especificam como AWS DMS manipula esses registros por padrão. Esses valores não afetam os mapeamentos de atributos de forma alguma.

Utilize `map-record-to-record` ao migrar de um banco de dados relacional para um fluxo de dados do Kinesis. Esse tipo de regra utiliza o valor `taskResourceId.schemaName.tableName` encontrado no banco de dados relacional como a chave de partição no fluxo de dados do Kinesis e cria um atributo para cada coluna no banco de dados de origem.

Ao utilizar `map-record-to-record`, observe o seguinte:

- Essa configuração afeta somente as colunas excluídas pela lista `exclude-columns`.
- Para cada coluna desse tipo, AWS DMS cria um atributo correspondente no tópico de destino.
- AWS DMS cria esse atributo correspondente independentemente de a coluna de origem ser usada em um mapeamento de atributos.

Utilize `map-record-to-document` para colocar colunas de origem em um único documento sem formatação no fluxo de destino apropriado utilizando o nome do atributo "_doc". O AWS DMS coloca os dados em um único mapa sem formatação na origem chamada "`_doc`". Esse posicionamento se aplica a qualquer coluna na tabela de origem que não aparece na lista de atributos `exclude-columns`.

Uma maneira de compreender o `map-record-to-record` é vê-lo em ação. Para este exemplo, suponha que você está começando com uma linha de tabela do banco de dados relacional com a seguinte estrutura de dados:

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876543210	29/02/1988

Para migrar essas informações de um esquema chamado `Test` para um fluxo de dados do Kinesis, crie regras para mapear os dados para o fluxo de destino. A regra a seguir ilustra o mapeamento.

```
{
    "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"
            }
        }
    ]
}
```

Veja a seguir uma ilustração do formato do registro resultante no fluxo de dados do Kinesis:

- StreamName: XXX
- PartitionKey: Test.Customers //schmaname.tableName
- Data: //A seguinte mensagem do JSON

```
  {
     "FirstName": "Randy",
     "LastName": "Marsh",
     "StoreId":  "5",
     "HomeAddress": "221B Baker Street",
     "HomePhone": "1234567890",
     "WorkAddress": "31 Spooner Street, Quahog",
     "WorkPhone": "9876543210",
     "DateOfBirth": "02/29/1988"
  }
```

No entanto, suponha que você utilize as mesmas regras, mas altere o parâmetro `rule-action` para `map-record-to-document` e exclua determinadas colunas. A regra a seguir ilustra o mapeamento.

```
{
 "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"
    ]
   }
  }
 ]
}
```

Nesse caso, as colunas não listadas no parâmetro `exclude-columns`, `FirstName`, `LastName`, `StoreId` e `DateOfBirth` são mapeadas para `_doc`. Veja a seguir o formato do registro resultante.

```
      {
           "data":{
               "_doc":{
                   "FirstName": "Randy",
                   "LastName": "Marsh",
                   "StoreId":  "5",
                   "DateOfBirth": "02/29/1988"
               }
           }
```

```
        }
```

Reestruturação de dados com mapeamento de atributo

É possível reestruturar os dados enquanto estiver migrando-os para um fluxo de dados do Kinesis utilizando um mapa de atributo. Por exemplo, você pode combinar vários campos na origem em um único campo no destino. O mapa de atributo a seguir ilustra como reestruturar os dados.

```
{
    "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}"
                            }
                        }
                    }
                ]
            }
        }
    ]
}
```

Para definir um valor constante para `partition-key`, especifique um valor de `partition-key`. Por exemplo, é possível fazer isso para forçar o armazenamento de todos os dados em um único fragmento. O mapeamento a seguir ilustra esse método.

```
{
    "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**
>
> O valor do `partition-key` para um registro de controle para uma tabela específica é `TaskId.SchemaName.TableName`. O valor do `partition-key` para um registro de controle específico para uma tarefa é o `TaskId` daquele registro. A especificação de um valor do `partition-key` no mapeamento do objeto não tem impacto sobre o `partition-key` no caso dos registros de controle.

Formato de mensagem do Kinesis Data Streams

A saída JSON é simplesmente uma lista de pares chave/valor. Um formato de mensagem JSON_UNFORMATTED é uma string JSON de linha única com novo delimitador de linha.

AWS DMS fornece os seguintes campos reservados para facilitar o consumo dos dados do Kinesis Data Streams:

RecordType

O tipo de registro pode ser dados ou controle. Os registros de dados representam as linhas reais na origem. Os registros de controle são relacionados a importantes eventos no stream, como a reinicialização de uma tarefa, por exemplo.

Operation

Para registros de dados, a operação pode ser `load`, `insert`, `update` ou `delete`.

Para registros de controle, a operação pode ser `create-table`, `rename-table`, `drop-table`, `change-columns`, `add-column`, `drop-column`, `rename-column` ou `column-type-change`.

SchemaName

O esquema de origem para o registro. Esse campo pode estar vazio para um registro de controle.

TableName

A tabela de origem para um registro. Esse campo pode estar vazio para um registro de controle.

Timestamp

A marca de data e hora de quando a mensagem do JSON foi criada. O campo é formatado com o formato ISO 8601.

Usando o Apache Kafka como alvo para AWS Database Migration Service

Você pode usar AWS DMS para migrar dados para um cluster Apache Kafka. O Apache Kafka é uma plataforma de streaming distribuída. É possível utilizar o Apache Kafka para a ingestão e o processamento de dados de streaming em tempo real.

AWS também oferece Amazon Managed Streaming for Apache Kafka (Amazon MSK) para uso como destino. AWS DMS O Amazon MSK é um serviço de streaming totalmente gerenciado do Apache Kafka que simplifica a implementação e o gerenciamento de instâncias do Apache Kafka. Ele funciona com versões de código aberto do Apache Kafka, e você acessa instâncias do Amazon MSK como AWS DMS destinos, exatamente como qualquer instância do Apache Kafka. Para obter mais informações, consulte O que é o Amazon MSK? no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

Um cluster do Kafka armazena fluxos de registros em categorias chamadas tópicos que são divididos em partições. As partições são sequências de registros de dados identificadas exclusivamente (mensagens) em um tópico. As partições podem ser distribuídas entre vários agentes em um cluster para permitir o processamento paralelo dos registros de um tópico. Para obter mais informações sobre tópicos e partições e sua distribuição no Apache Kafka, consulte Tópicos e logs e Distribuição.

O cluster do Kafka pode ser uma instância do Amazon MSK, um cluster em execução em uma instância do Amazon EC2 ou um cluster on-premises. Uma instância do Amazon MSK ou um cluster em uma instância do Amazon EC2 pode estar na mesma VPC ou em uma diferente. Se o cluster estiver on-premises, será possível utilizar o seu próprio servidor de nomes on-premises para a instância de replicação para resolver o nome do host do cluster. Para obter informações sobre como configurar um servidor de nomes na instância de replicação, consulte Utilização do seu próprio servidor de nomes on-premises. Para obter mais informações sobre a configuração de uma rede, consulte Configurar uma rede para uma instância de replicação.

Ao utilizar um cluster do Amazon MSK, verifique se o grupo de segurança permite acesso da instância de replicação. Para obter informações sobre como alterar o grupo de segurança de um cluster do Amazon MSK, consulte Alterar o grupo de segurança de um cluster do Amazon MSK.

AWS Database Migration Service publica registros em um tópico do Kafka usando JSON. Durante a conversão, o AWS DMS serializa cada registro do banco de dados de origem em um par atributo-valor no formato JSON.

Para migrar os dados de uma fonte de dados compatível para um cluster de destino do Kafka, utilize o mapeamento de objetos. Com o mapeamento de objetos, você determina como estruturar os registros de dados no tópico de destino. Também é possível definir uma chave de partição para cada tabela, que será utilizada pelo Apache Kafka para agrupar os dados em suas partições.

Atualmente, AWS DMS oferece suporte a um único tópico por tarefa. Para uma única tarefa com várias tabelas, todas as mensagens vão para um único tópico. Cada mensagem inclui uma seção de metadados que identifica o esquema e a tabela de destino. AWS DMS as versões 3.4.6 e superiores oferecem suporte à replicação multitópica usando mapeamento de objetos. Para ter mais informações, consulte Replicação de multitópico utilizando o mapeamento de objetos.

Configurações do endpoint do Apache Kafka

Você pode especificar os detalhes da conexão por meio das configurações do endpoint no AWS DMS console ou da `--kafka-settings` opção na CLI. Os requisitos para cada configuração são os seguintes:

- `Broker`: especifique a localização de um ou mais agentes no cluster do Kafka na forma de uma lista separada por vírgulas de cada *`broker-hostname`*`:`*`port`*. Um exemplo é `"ec2-12-345-678-901.compute-1.amazonaws.com:2345,ec2-10-987-654-321.compute-1.a` Essa configuração pode especificar os locais de qualquer um ou de todos os agentes no cluster. Todos os agentes de cluster se comunicam para lidar com o particionamento de registros de dados migrados para o tópico.
- `Topic`: (opcional) especifique o nome do tópico com um comprimento máximo de 255 letras e símbolos. É possível utilizar ponto (.), sublinhado (_) e sinal de subtração (-). Os nomes de tópicos com um ponto (.) ou sublinhado (_) podem colidir em estruturas de dados internas. Utilize qualquer um, mas não esses dois símbolos no nome do tópico. Se você não especificar um nome de tópico, AWS DMS use `"kafka-default-topic"` como tópico de migração.

> **Note**
>
> Para AWS DMS criar um tópico de migração especificado por você ou o tópico padrão, defina-o `auto.create.topics.enable = true` como parte da configuração do cluster

do Kafka. Para obter mais informações, consulte Limitações ao usar o Apache Kafka como alvo para AWS Database Migration Service.

- `MessageFormat`: o formato de saída dos registros criados no endpoint. O formato da mensagem é `JSON` (padrão) ou `JSON_UNFORMATTED` (uma única linha sem guia).
- `MessageMaxBytes`: o tamanho máximo em bytes dos registros criados no endpoint. O padrão é 1.000.000.

> **Note**
>
> Você só pode usar o AWS CLI/SDK para mudar `MessageMaxBytes` para um valor não padrão. Por exemplo, para modificar o endpoint existente do Kafka e alterar `MessageMaxBytes`, utilize o comando a seguir.

```
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`: fornece informações detalhadas sobre transações do banco de dados de origem. Essas informações incluem um timestamp de confirmação, uma posição no log e valores para `transaction_id`, `previous_transaction_id` e `transaction_record_id` (o deslocamento de registro dentro de uma transação). O padrão é `false`.
- `IncludePartitionValue`: mostra o valor da partição na saída da mensagem do Kafka, a menos que o tipo de partição seja `schema-table-type`. O padrão é `false`.
- `PartitionIncludeSchemaTable`: prefixa os nomes de esquema e de tabela em valores de partições, quando o tipo de partição for `primary-key-type`. Isso aumenta a distribuição de dados entre partições do Kafka. Por exemplo, suponha que um esquema `SysBench` tenha milhares de tabelas, e cada tabela tenha apenas um intervalo limitado para uma chave primária. Nesse caso, a mesma chave primária é enviada de milhares de tabelas para a mesma partição, o que provoca o controle de utilização. O padrão é `false`.
- `IncludeTableAlterOperations`: inclui todas as operações da linguagem de definição de dados (DDL) que alteram a tabela nos dados de controle, como `rename-table`, `drop-table`, `add-column`, `drop-column` e `rename-column`. O padrão é `false`.

- `IncludeControlDetails`: mostra informações detalhadas de controle para definição de tabela, definição de coluna e alterações de tabela e coluna na saída de mensagem do Kafka. O padrão é `false`.
- `IncludeNullAndEmpty`: inclui colunas NULL e vazias no destino. O padrão é `false`.
- `SecurityProtocol`: define uma conexão segura a um endpoint de destino do Kafka utilizando Transport Layer Security (TLS). As opções incluem `ssl-authentication`, `ssl-encryption` e `sasl-ssl`. A utilização de `sasl-ssl` requer `SaslUsername` e `SaslPassword`.
- `SslEndpointIdentificationAlgorithm`— Define a verificação do nome do host para o certificado. Essa configuração é suportada na AWS DMS versão 3.5.1 e posterior. As opções incluem o seguinte:
  - `NONE`: desative a verificação do nome do host do broker na conexão do cliente.
  - `HTTPS`: Habilite a verificação do nome do host do broker na conexão do cliente.

É possível utilizar configurações para ajudar a aumentar a velocidade da transferência. Para fazer isso, o AWS DMS é compatível com uma carga multithreaded completa para um cluster de destino do Apache Kafka. O AWS DMS é compatível com esse multithreading com configurações de tarefa que incluem o seguinte:

- `MaxFullLoadSubTasks`— Use essa opção para indicar o número máximo de tabelas de origem a serem carregadas paralelamente. AWS DMS carrega cada tabela em sua tabela de destino correspondente do Kafka usando uma subtarefa dedicada. O padrão é 8; o valor máximo é 49.
- `ParallelLoadThreads`— Use essa opção para especificar o número de segmentos AWS DMS usados para carregar cada tabela em sua tabela de destino do Kafka. O valor máximo para um destino do Apache Kafka é 32. Você pode solicitar o aumento desse limite máximo.
- `ParallelLoadBufferSize`: utilize esta opção para especificar o número máximo de registros a serem armazenados no buffer utilizado pelos threads de carregamento paralelo utilizam para carregar dados no destino do Kafka. O valor padrão é 50. Valor máximo de 1.000. Use essa configuração com `ParallelLoadThreads`; `ParallelLoadBufferSize` é válido somente quando há mais de um thread.
- `ParallelLoadQueuesPerThread`: utilize esta opção para especificar o número de filas que cada thread simultâneo acessa para extrair registros de dados das filas e gerar uma carga em lote para o destino. O padrão é um. O máximo é 512.

É possível melhorar o desempenho da captura de dados de alteração (CDC) para endpoints do Kafka ajustando as configurações da tarefa para threads paralelos e operações em massa. Para fazer isso, especifique o número de threads simultâneos, filas por thread e o número de registros a serem armazenados em um buffer usando as configurações da tarefa `ParallelApply*`. Por exemplo, suponha que você queira executar um carregamento de CDC e aplicar 128 threads em paralelo. Você também quer acessar 64 filas por thread, com 50 registros armazenados por buffer.

Para promover o desempenho do CDC, AWS DMS oferece suporte a estas configurações de tarefas:

- `ParallelApplyThreads`— Especifica o número de threads simultâneos que são AWS DMS usados durante um carregamento do CDC para enviar registros de dados para um endpoint de destino do Kafka. O valor padrão é zero (0) e o valor máximo é 32.
- `ParallelApplyBufferSize`: especifica o número máximo de registros a serem armazenados em cada fila de buffer para threads simultâneos enviarem para um endpoint de destino do Kafka durante uma carga de CDC. O valor padrão é 100 e o valor máximo é 1.000. Use essa opção quando `ParallelApplyThreads` especificar mais de um thread.
- `ParallelApplyQueuesPerThread`: especifica o número de filas que cada thread acessa para extrair registros de dados das filas e gerar uma carga em lote para um endpoint do Kafka durante a CDC. O padrão é um. O máximo é 512.

Ao usar configurações da tarefa `ParallelApply*`, o `partition-key-type` padrão é a `primary-key` da tabela, não o `schema-name.table-name`.

Conectar-se ao Kafka utilizando Transport Layer Security (TLS)

O cluster do Kafka aceita conexões seguras utilizando Transport Layer Security (TLS). Com o DMS, é possível utilizar qualquer uma das três opções de protocolo de segurança a seguir para proteger uma conexão de endpoint do Kafka.

Criptografia SSL (**`server-encryption`**)

Os clientes validam a identidade do servidor por meio do certificado do servidor. Uma conexão criptografada é feita entre o servidor e o cliente.

Autenticação SSL (**`mutual-authentication`**)

O servidor e o cliente validam a identidade entre si por meio de seus próprios certificados. Uma conexão criptografada é feita entre o servidor e o cliente.

SASL-SSL (**`mutual-authentication`**)

O método Simple Authentication and Security Layer (SASL) substitui o certificado do cliente por um nome de usuário e senha para validar a identidade do cliente. Especificamente, forneça um nome de usuário e uma senha que o servidor registrou para que o servidor possa validar a identidade de um cliente. Uma conexão criptografada é feita entre o servidor e o cliente.

> **Important**
>
> O Apache Kafka e o Amazon MSK aceitam certificados resolvidos. Essa é uma limitação conhecida do Kafka e do Amazon MSK a ser resolvida. Para obter mais informações, consulte Problemas do Apache Kafka, KAFKA-3700.
> Se estiver utilizando o Amazon MSK, considere utilizar listas de controle de acesso (ACLs) como uma solução alternativa para essa limitação conhecida. Para obter mais informações sobre como utilizar ACLs, consulte a seção ACLs do Apache Kafka do Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.
> Se estiver utilizando um cluster do Kafka autogerenciado, consulte Comentário datado de 18/out/21 para obter informações sobre como configurar o cluster.

Utilizar criptografia SSL com o Amazon MSK ou um cluster do Kafka autogerenciado

É possível utilizar a criptografia SSL para proteger uma conexão de endpoint ao Amazon MSK ou um cluster do Kafka autogerenciado. Ao utilizar o método de autenticação de criptografia SSL, os clientes validam a identidade de um servidor por meio do certificado do servidor. Uma conexão criptografada é feita entre o servidor e o cliente.

Como utilizar a criptografia SSL para conectar-se ao Amazon MSK

- Defina a configuração do endpoint do protocolo de segurança (`SecurityProtocol`) utilizando a opção `ssl-encryption` ao criar o endpoint do Kafka de destino.

  O exemplo de JSON a seguir define o protocolo de segurança como criptografia SSL.

```
"KafkaSettings": {
    "SecurityProtocol": "ssl-encryption",
}
```

Para utilizar a criptografia SSL para um cluster do Kafka autogerenciado

1. Se estiver utilizando uma Autoridade de Certificação (CA) privada no cluster do Kafka on-premises, faça upload do certificado da CA privada e obtenha um nome do recurso da Amazon (ARN).
2. Defina a configuração do endpoint do protocolo de segurança (`SecurityProtocol`) utilizando a opção `ssl-encryption` ao criar o endpoint do Kafka de destino. O exemplo de JSON a seguir define o protocolo de segurança como `ssl-encryption`.

```
"KafkaSettings": {
    "SecurityProtocol": "ssl-encryption",
}
```

3. Se estiver utilizando uma CA privada, defina `SslCaCertificateArn` no ARN obtido na primeira etapa acima.

Utilizar autenticação SSL

É possível utilizar a autenticação SSL para proteger uma conexão de endpoint ao Amazon MSK ou um cluster do Kafka autogerenciado.

Para ativar a autenticação e a criptografia do cliente utilizando a autenticação SSL para conectar-se ao Amazon MSK, faça o seguinte:

- Prepare uma chave privada e um certificado público para o Kafka.
- Faça upload dos certificados no gerenciador de certificados do DMS.
- Crie um endpoint de destino do Kafka com os ARNs dos certificados correspondentes especificados nas configurações do endpoint do Kafka.

Como preparar uma chave privada e um certificado público para o Amazon MSK.

1. Crie uma instância do EC2 e configure um cliente para utilizar a autenticação conforme descrito nas etapas de 1 a 9 na seção Autenticação de cliente do Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

Depois de concluir essas etapas, você tem o ARN de um certificado (o ARN do certificado público salvo no ACM) e uma chave privada contida em um arquivo `kafka.client.keystore.jks`.

2. Obtenha o certificado público e copie o certificado no arquivo `signed-certificate-from-acm.pem`, utilizando o comando a seguir:

```
aws acm-pca get-certificate --certificate-authority-arn Private_CA_ARN --
certificate-arn Certificate_ARN
```

O comando retorna informações semelhantes às do exemplo a seguir.

```
{"Certificate": "123", "CertificateChain": "456"}
```

Copie o equivalente de `"123"` no arquivo `signed-certificate-from-acm.pem`.

3. Obtenha a chave privada importando a chave `msk-rsa` de `kafka.client.keystore.jks to keystore.p12`, conforme mostrado no exemplo a seguir.

```
keytool -importkeystore \
-srckeystore kafka.client.keystore.jks \
-destkeystore keystore.p12 \
-deststoretype PKCS12 \
-srcalias msk-rsa-client \
-deststorepass test1234 \
-destkeypass test1234
```

4. Utilize o comando a seguir para exportar `keystore.p12` para o formato `.pem`.

```
Openssl pkcs12 -in keystore.p12 -out encrypted-private-client-key.pem –nocerts
```

A mensagem Inserir frase secreta do PEM é exibida e identifica a chave aplicada para criptografar o certificado.

5. Remova os atributos bag e os atributos-chave do arquivo `.pem` para garantir que a primeira linha comece com a sequência de caracteres a seguir.

```
                            ---BEGIN ENCRYPTED PRIVATE KEY---
```

Como fazer upload de um certificado público e de uma chave privada no gerenciador de certificados do DMS e testar a conexão ao Amazon MSK

1. Faça upload do gerenciador de certificados do DMS utilizando o comando a seguir.

```
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. Crie um endpoint de destino do Amazon MSK e teste a conexão para garantir que a autenticação TLS funcione.

```
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

É possível utilizar a autenticação SSL para proteger uma conexão a um cluster do Kafka autogerenciado. Em alguns casos, é possível utilizar uma Autoridade de Certificação (CA) privada no cluster do Kafka on-premises. Nesse caso, faça upload da cadeia de CAs, do certificado público e da chave privada para o gerenciador de certificados do DMS. Utilize o nome do recurso da Amazon (ARN) correspondente nas configurações do endpoint ao criar o endpoint de destino do Kafka on-premises.

Como preparar uma chave privada e um certificado assinado para um cluster do Kafka autogerenciado

1. Gere um par de chaves como mostrado no exemplo a seguir.

```
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. Gere uma solicitação de assinatura de certificado (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. Utilize a CA no truststore do cluster para assinar a CSR. Se não tiver uma CA, você poderá criar sua própria CA privada.

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days validate-days
```

4. Importe `ca-cert` para o truststore e o keystore do servidor. Se você não tiver um truststore, utilize o seguinte comando para criar o truststore e importar `ca-cert` nele.

```
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. Assine o certificado.

```
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. Importe o certificado assinado para o keystore.

```
keytool -keystore kafka.server.keystore.jks -import -file signed-certificate.pem -
alias on-premise-rsa -storepass your-keystore-password
-keypass your-key-password
```

7. Utilize o comando a seguir para importar a chave `on-premise-rsa` de `kafka.server.keystore.jks` para `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. Utilize o comando a seguir para exportar `keystore.p12` para o formato `.pem`.

```
Openssl pkcs12 -in keystore.p12 -out encrypted-private-server-key.pem –nocerts
```

9. Faça upload de `encrypted-private-server-key.pem`, `signed-certificate.pem` e `ca-cert` para o gerenciador de certificados do DMS.
10. Crie um endpoint utilizando os ARNs retornados.

```
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
```

Utilizar a autenticação SASL-SSL para se conectar ao Amazon MSK

O método Simple Authentication and Security Layer (SASL) utiliza um nome de usuário e uma senha para validar a identidade de um cliente e faz uma conexão criptografada entre o servidor e o cliente.

Para utilizar o SASL, primeiro crie um nome de usuário e uma senha seguros ao configurar o cluster do Amazon MSK. Para obter uma descrição de como configurar um nome de usuário e senha seguros para um cluster do Amazon MSK, consulte Configurar a autenticação SASL/SCRAM para

um cluster do Amazon MSK no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

Crie o endpoint de destino do Kafka, defina a configuração do endpoint do protocolo de segurança (`SecurityProtocol`) utilizando a opção `sasl-ssl`. Defina também as opções `SaslUsername` e `SaslPassword`. Verifique se essas opções são consistentes com o nome de usuário e senha seguros criados ao configurar o cluster do Amazon MSK pela primeira vez, conforme mostrado no exemplo de JSON a seguir.

```
"KafkaSettings": {
    "SecurityProtocol": "sasl-ssl",
    "SaslUsername":"Amazon MSK cluster secure user name",
    "SaslPassword":"Amazon MSK cluster secure password"
}
```

> **Note**
>
> - Atualmente, AWS DMS oferece suporte somente a SASL-SSL público apoiado pela CA. O DMS não é compatível com SASL-SSL para utilização com o Kafka autogerenciado compatível com a CA privada.
> - Para autenticação SASL-SSL, AWS DMS suporta o mecanismo SCRAM-SHA-512 por padrão. AWS DMS as versões 3.5.0 e superiores também suportam o mecanismo Plain. Para ser compatível com o mecanismo Plain, defina o parâmetro `SaslMechanism` do tipo de dados da API do `KafkaSettings` como `PLAIN`.

Utilizar uma imagem anterior para visualizar os valores originais de linhas da CDC para o Apache Kafka como destino

Ao gravar as atualizações de CDC em um destino de streaming de dados, como o Kafka, é possível visualizar os valores originais de linhas do banco de dados de origem antes da alteração por uma atualização. Para tornar isso possível, AWS DMS preenche uma imagem anterior dos eventos de atualização com base nos dados fornecidos pelo mecanismo do banco de dados de origem.

Diferentes mecanismos de banco de dados de origem fornecem diferentes quantidades de informações para uma imagem anterior:

- O Oracle fornece atualizações para colunas somente se elas forem alteradas.
- O PostgreSQL fornece somente dados para colunas que fazem parte da chave primária (alterada ou não). Se a replicação lógica estiver em uso e a REPLICA IDENTITY FULL estiver definida para a tabela de origem, será possível obter informações completas de antes e depois na linha gravada nos WALs e disponíveis aqui.
- O MySQL geralmente fornece dados para todas as colunas (alteradas ou não).

Para habilitar a criação de imagem anterior para adicionar valores originais do banco de dados de origem à saída do AWS DMS , use a configuração de tarefa `BeforeImageSettings` ou o parâmetro `add-before-image-columns`. Esse parâmetro aplica uma regra de transformação de coluna.

`BeforeImageSettings` adiciona um novo atributo JSON a cada operação de atualização com valores coletados do sistema de banco de dados de origem, conforme mostrado a seguir.

```
"BeforeImageSettings": {
    "EnableBeforeImage": boolean,
    "FieldName": string,
    "ColumnFilter": pk-only (default) / non-lob / all (but only one)
}
```

> **Note**
>
> Aplique `BeforeImageSettings` às tarefas de carga máxima e de CDC (que migram dados existentes e replicam alterações contínuas) ou às tarefas de somente CDC (que replicam somente as alterações de dados). Não aplique `BeforeImageSettings` a tarefas que são somente de carga total.

Para opções `BeforeImageSettings`, aplica-se o seguinte:

- Defina a opção `EnableBeforeImage` como `true` para habilitar a criação de imagem anterior. O padrão é `false`.
- Use a opção `FieldName` para atribuir um nome ao novo atributo JSON. Quando `EnableBeforeImage` for `true`, `FieldName` será necessário e não poderá estar vazio.

- A opção `ColumnFilter` especifica uma coluna a ser adicionada usando imagem anterior. Para adicionar somente colunas que fazem parte das chaves primárias da tabela, use o valor padrão, `pk-only`. Para adicionar somente colunas que não são do tipo LOB, utilize `non-lob`. Para adicionar qualquer coluna que tenha um valor de imagem anterior, use `all`.

```
"BeforeImageSettings": {
    "EnableBeforeImage": true,
    "FieldName": "before-image",
    "ColumnFilter": "pk-only"
  }
```

Usar uma regra de transformação de imagem anterior

Como alternativa às configurações de tarefa, é possível usar o parâmetro `add-before-image-columns`, que aplica uma regra de transformação de coluna. Com esse parâmetro, é possível habilitar a criação de imagem anterior durante a CDC em destinos de streaming de dados, como o Kafka.

Usando `add-before-image-columns` em uma regra de transformação, é possível aplicar um controle mais refinado dos resultados da imagem anterior. As regras de transformação permitem que você use um localizador de objetos que oferece controle sobre as tabelas selecionadas para a regra. Além disso, é possível encadear regras de transformação, o que permite que regras diferentes sejam aplicadas a tabelas diferentes. Depois, você poderá manipular as colunas produzidas usando outras regras.

> **Note**
>
> Não use o parâmetro `add-before-image-columns` junto com a configuração da tarefa `BeforeImageSettings` na mesma tarefa. Em vez disso, use o parâmetro ou a configuração, mas não ambos, para uma única tarefa.

Um tipo de regra `transformation` com o parâmetro `add-before-image-columns` de uma coluna deve fornecer uma seção `before-image-def`. Por exemplo:

```
    {
      "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,
      }
    }
```

O valor de `column-prefix` precede um nome de coluna e o valor padrão de `column-prefix` é `BI_`. O valor de `column-suffix` é anexado ao nome da coluna e o padrão é vazio. Não defina `column-prefix` e `column-suffix` como strings vazias.

Escolha um valor para `column-filter`. Para adicionar somente colunas que fazem parte das chaves primárias da tabela, escolha `pk-only`. Escolha `non-lob` para adicionar somente colunas que não sejam do tipo LOB. Ou escolha `all` para adicionar qualquer coluna que tenha um valor de imagem anterior.

Exemplo de uma regra de transformação de imagem anterior

A regra de transformação no exemplo a seguir adiciona uma nova coluna chamada `BI_emp_no` no destino. Portanto, uma instrução como `UPDATE employees SET emp_no = 3 WHERE emp_no = 1;` preenche o campo `BI_emp_no` com 1. Ao gravar atualizações da CDC em destinos do Amazon S3, a coluna `BI_emp_no` possibilita identificar qual linha original foi atualizada.

```
{
  "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"
      }
    }
  ]
}
```

Para obter informações sobre como usar a ação da regra `add-before-image-columns`, consulte Regras de transformação e ações.

Limitações ao usar o Apache Kafka como alvo para AWS Database Migration Service

Aplicam-se as seguintes limitações ao utilizar o Apache Kafka como destino:

- AWS DMS Os endpoints de destino do Kafka não oferecem suporte ao controle de acesso do IAM para o Amazon Managed Streaming for Apache Kafka (Amazon MSK).
- O modo Full LOB não é compatível.
- Especifique um arquivo de configuração do Kafka para seu cluster com propriedades que permitem AWS DMS criar novos tópicos automaticamente. Inclua a configuração, `auto.create.topics.enable = true`. Se estiver utilizando o Amazon MSK, será possível especificar a configuração padrão ao criar o cluster do Kafka e alterar a configuração `auto.create.topics.enable` para `true`. Para obter mais informações sobre as configurações padrão, consulte A configuração padrão do Amazon MSK no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka. Se você precisar modificar um cluster existente do Kafka criado usando o Amazon MSK, execute o AWS CLI comando `aws kafka create-configuration` para atualizar sua configuração do Kafka, como no exemplo a seguir:

  ```
  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"
}
```

Aqui, `//~/kafka_configuration` é o arquivo configuração criado com as configurações de propriedades necessárias.

Se você estiver usando sua própria instância do Kafka instalada no Amazon EC2, modifique a configuração do cluster Kafka com `auto.create.topics.enable = true` a configuração para AWS DMS permitir a criação automática de novos tópicos, usando as opções fornecidas com sua instância.

- AWS DMS publica cada atualização em um único registro no banco de dados de origem como um registro de dados (mensagem) em um determinado tópico do Kafka, independentemente das transações.
- AWS DMS suporta as duas formas a seguir para chaves de partição:
  - `SchemaName.TableName`: uma combinação de esquema e nome da tabela.
  - `${AttributeName}`: o valor de um dos campos no JSON ou a chave primária da tabela no banco de dados de origem.
- O `BatchApply` não é compatível com um endpoint do Kafka. A utilização da aplicação em lote (por exemplo, a configuração da tarefa de metadados de destino `BatchApplyEnabled`) para um destino do Kafka pode resultar em perda de dados.
- AWS DMS não suporta a migração de valores do tipo de `BigInt` dados com mais de 16 dígitos. Para contornar essa limitação, você pode usar a regra de transformação a seguir para converter a coluna `BigInt` em uma string. Para obter mais informações sobre regras transformação, consulte Regras de transformação e ações.

```
{
    "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
    }
}
```

Utilizar o mapeamento de objetos para migrar dados para um tópico do Kafka

AWS DMS usa regras de mapeamento de tabelas para mapear dados da fonte para o tópico de destino do Kafka. Para mapear dados para um tópico de destino, utilize um tipo de regra de mapeamento de tabelas chamado mapeamento de objetos. Utilize o mapeamento de objetos para definir como os registros de dados na origem são mapeados para os registros de dados publicados em um tópico do Kafka.

Os tópicos do Kafka não têm uma estrutura predefinida além de uma chave de partição.

> **Note**
>
> Não é necessário utilizar o mapeamento de objetos. É possível utilizar o mapeamento de tabela normal para várias transformações. No entanto, o tipo de chave de partição seguirá estes comportamentos padrão:
>
> - A chave primária é utilizada como uma chave de partição para a carga máxima.
> - Se nenhuma configuração da tarefa de aplicação paralela for utilizada, `schema.table` será utilizada como uma chave de partição para a CDC.
> - Se as configurações de tarefas de aplicação paralela forem utilizadas, a chave primária será utilizada como uma chave de partição para a CDC.

Para criar uma regra de mapeamento de objetos, especifique `rule-type` como `object-mapping`. Essa regra especifica o tipo de mapeamento de objeto que você deseja usar.

A estrutura da regra é a seguinte:

```
{
    "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 atualmente suporta `map-record-to-record` e `map-record-to-document` como os únicos valores válidos para o `rule-action` parâmetro. Essas configurações afetam valores que não são excluídos como parte da lista de atributos `exclude-columns`. Os `map-record-to-document` valores `map-record-to-record` e especificam como AWS DMS manipula esses registros por padrão. Esses valores não afetam os mapeamentos de atributos de forma alguma.

Utilize o `map-record-to-record` ao migrar de um banco de dados relacional para um tópico do Kafka. Esse tipo de regra utiliza o valor `taskResourceId.schemaName.tableName` encontrado no banco de dados relacional como a chave de partição no tópico do Kafka e cria um atributo para cada coluna no banco de dados de origem.

Ao utilizar `map-record-to-record`, observe o seguinte:

- Essa configuração afeta somente as colunas excluídas pela lista `exclude-columns`.
- Para cada coluna desse tipo, AWS DMS cria um atributo correspondente no tópico de destino.
- AWS DMS cria esse atributo correspondente independentemente de a coluna de origem ser usada em um mapeamento de atributos.

Uma maneira de compreender o `map-record-to-record` é vê-lo em ação. Para este exemplo, suponha que você está começando com uma linha de tabela do banco de dados relacional com a seguinte estrutura de dados:

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner	9876543210	29/02/1988

FirstName	LastName	StoreId	HomeAddre ss	HomePhon	WorkAddre ss	WorkPhone	DateofBir th
					Street, Quahog		

Para migrar essas informações de um esquema chamado `Test` para um tópico do Kafka, crie regras para mapear os dados para o tópico de destino. A regra a seguir ilustra o mapeamento.

```
{
    "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"
            }
        }
    ]
}
```

Com um tópico do Kafka e uma chave de partição determinados (neste caso, `taskResourceId.schemaName.tableName`), o seguinte ilustra o formato do registro resultante utilizando os nossos exemplos de dados no tópico de destino do Kafka:

```
  {
     "FirstName": "Randy",
```

```
    "LastName": "Marsh",
    "StoreId":  "5",
    "HomeAddress": "221B Baker Street",
    "HomePhone": "1234567890",
    "WorkAddress": "31 Spooner Street, Quahog",
    "WorkPhone": "9876543210",
    "DateOfBirth": "02/29/1988"
  }
```

Tópicos

- Reestruturação de dados com mapeamento de atributo
- Replicação de multitópico utilizando o mapeamento de objetos
- Formato de mensagem do Apache Kafka

Reestruturação de dados com mapeamento de atributo

É possível reestruturar os dados ao migrá-los para um tópico do Kafka utilizando um mapa de atributos. Por exemplo, você pode combinar vários campos na origem em um único campo no destino. O mapa de atributo a seguir ilustra como reestruturar os dados.

```
{
    "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}"
                            }
                        }
                    }
                ]
            }
        }
    ]
}
```

Para definir um valor constante para `partition-key`, especifique um valor de `partition-key`. Por exemplo, é possível fazer isso para forçar o armazenamento de todos os dados em uma única partição. O mapeamento a seguir ilustra esse método.

```
{
    "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**
>
> O valor do `partition-key` para um registro de controle para uma tabela específica é `TaskId.SchemaName.TableName`. O valor do `partition-key` para um registro de controle específico para uma tarefa é o `TaskId` daquele registro. A especificação de um valor do `partition-key` no mapeamento do objeto não tem impacto sobre o `partition-key` no caso dos registros de controle.

Replicação de multitópico utilizando o mapeamento de objetos

Por padrão, AWS DMS as tarefas migram todos os dados de origem para um dos tópicos do Kafka a seguir:

- Conforme especificado no campo Tópico do endpoint de AWS DMS destino.
- Conforme especificado por `kafka-default-topic`, se o campo Tópico do endpoint de destino não estiver preenchido e a configuração `auto.create.topics.enable` do Kafka estiver definida como `true`.

Com as versões 3.4.6 e posteriores do AWS DMS mecanismo, você pode usar o `kafka-target-topic` atributo para mapear cada tabela de origem migrada para um tópico separado. Por exemplo, as regras de mapeamento de objetos a seguir migram as tabelas de origem `Customer` e `Address`

para os tópicos `customer_topic` e `address_topic` do Kafka, respectivamente. Ao mesmo tempo, AWS DMS migra todas as outras tabelas de origem, incluindo a `Bills` tabela no `Test` esquema, para o tópico especificado no endpoint de destino.

```
{
    "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"
            }
        }
    ]
}
```

Ao utilizar a replicação de multitópico do Kafka, é possível agrupar e migrar tabelas de origem para tópicos separados do Kafka utilizando uma única tarefa de replicação.

Formato de mensagem do Apache Kafka

A saída JSON é simplesmente uma lista de pares chave/valor.

RecordType

O tipo de registro pode ser dados ou controle. Os registros de dados representam as linhas reais na origem. Os registros de controle são relacionados a importantes eventos no stream, como a reinicialização de uma tarefa, por exemplo.

Operation

Para registros de dados, a operação pode ser `load`, `insert`, `update` ou `delete`.

Para registros de controle, a operação pode ser `create-table`, `rename-table`, `drop-table`, `change-columns`, `add-column`, `drop-column`, `rename-column` ou `column-type-change`.

SchemaName

O esquema de origem para o registro. Esse campo pode estar vazio para um registro de controle.

TableName

A tabela de origem para um registro. Esse campo pode estar vazio para um registro de controle.

Timestamp

A marca de data e hora de quando a mensagem do JSON foi criada. O campo é formatado com o formato ISO 8601.

O exemplo de mensagem JSON a seguir ilustra uma mensagem de tipo de dados com todos os metadados adicionais.

```
{
   "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"
   }
}
```

O exemplo de mensagem JSON a seguir ilustra uma mensagem de tipo de controle.

```
{
   "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"
    }
}
```

Utilizar um cluster do Amazon OpenSearch Service como destino do AWS Database Migration Service

É possível utilizar o AWS DMS para migrar dados para o Amazon OpenSearch Service (OpenSearch Service). O OpenSearch Service é um serviço gerenciado que facilita a implantação, a operação e a escala de clusters do OpenSearch Service.

No OpenSearch Service, você trabalha com índices e documentos. Um índice é uma coleção de documentos e um documento é um objeto JSON que contém valores escalares, matrizes e outros objetos. O OpenSearch fornece uma linguagem de consulta baseada em JSON, para que você possa consultar dados em um índice e recuperar os documentos correspondentes.

Quando o AWS DMS cria índices para um endpoint de destino do OpenSearch Service, ele cria um índice para cada tabela do endpoint de origem. O custo da criação um índice do OpenSearch Service depende de vários fatores: Do número de índices criados, da quantidade total de dados nesses

índices e da pequena quantidade de metadados armazenados que o OpenSearch armazena de cada um deles.

Configure o cluster do OpenSearch Service com recursos de computação e de armazenamento apropriados para o escopo da migração. É recomendável considerar os seguintes fatores, dependendo da tarefa de replicação a ser utilizada:

- Para uma carga máxima de dados, considere a quantidade total de dados a ser migrada e a velocidade da transferência.
- Para replicar alterações contínuas, considere a frequência das atualizações e os requisitos de latência para todo o processo.

Configure também as configurações de índice no cluster do OpenSearch, prestando atenção à contagem de documentos.

Configurações da tarefa de carga completa multithreaded

Para ajudar a aumentar a velocidade da transferência, o AWS DMS é compatível com uma carga máxima multithreaded para um cluster de destino do OpenSearch Service. O AWS DMS é compatível com esse multithreading com configurações de tarefas que incluem o seguinte:

- `MaxFullLoadSubTasks`: utilize esta opção para indicar o número máximo de tabelas de origem a serem carregadas em paralelo. O DMS carrega cada tabela no índice de destino do OpenSearch Service correspondente utilizando uma subtarefa dedicada. O padrão é 8; o valor máximo é 49.
- `ParallelLoadThreads`: utilize esta opção para especificar o número de threads que AWS DMS utiliza para carregar cada tabela no índice de destino do OpenSearch Service. O valor máximo de um destino do OpenSearch Service é 32. É possível solicitar o aumento desse limite máximo.

  > **Note**
  >
  > Se você não alterar `ParallelLoadThreads` de seu padrão (0), o AWS DMS transferirá um único registro por vez. Essa abordagem coloca uma carga indevida no cluster do OpenSearch Service. Defina essa opção como 1 ou mais.

- `ParallelLoadBufferSize`: utilize esta opção para especificar o número máximo de registros a ser armazenado no buffer utilizado pelos threads de carga paralela para o destino do OpenSearch Service. O valor padrão é 50. O valor máximo é 1.000. Utilize essa configuração com

`ParallelLoadThreads`. `ParallelLoadBufferSize` é válido somente quando há mais de um thread.

Para obter mais informações sobre como o DMS carrega um cluster do OpenSearch Service utilizando multithreading, consulte a publicação do blog da AWS Escalar o Amazon OpenSearch Service para migrações do AWS Database Migration Service.

Configurações da tarefa de carga de CDC multithreaded

É possível melhorar o desempenho da captura de dados de alteração (CDC) para um cluster de destino do OpenSearch Service utilizando as configurações da tarefa para modificar o comportamento da chamada da API `PutRecords`. Para fazer isso, especifique o número de threads simultâneos, as filas por thread e o número de registros a serem armazenados em um buffer utilizando as configurações da tarefa `ParallelApply*`. Por exemplo, suponha que você queira executar uma carga de CDC e aplicar 32 threads em paralelo. Você também quer acessar 64 filas por thread, com 50 registros armazenados por buffer.

> **Note**
>
> Compatibilidade com a utilização de configurações da tarefa `ParallelApply*` durante a CDC para endpoints de destino do Amazon OpenSearch Service está disponível nas versões 3.4.0 e posteriores do AWS DMS.

Para promover o desempenho da CDC, o AWS DMS é compatível com estas configurações de tarefa:

- `ParallelApplyThreads`: especifica o número de threads simultâneos que AWS DMS utiliza durante uma carga de CDC para enviar registros de dados para um endpoint de destino do OpenSearch Service. O valor padrão é zero (0) e o valor máximo é 32.
- `ParallelApplyBufferSize`: especifica o número máximo de registros a serem armazenados em cada fila de buffer para que os threads simultâneos enviem para um endpoint de destino do OpenSearch Service durante uma carga de CDC. O valor padrão é 100 e o valor máximo é 1.000. Utilize essa opção quando `ParallelApplyThreads` especificar mais de um thread.
- `ParallelApplyQueuesPerThread`: especifica o número de filas que cada thread acessa para extrair registros de dados das filas e gerar uma carga em lote para um endpoint do OpenSearch Service durante a CDC.

Ao utilizar configurações da tarefa `ParallelApply*`, `partition-key-type` padrão é a `primary-key` da tabela, não `schema-name.table-name`.

Migrar de uma tabela de banco de dados relacional para um índice do OpenSearch Service

O AWS DMS é compatível com a migração de dados para tipos de dados escalares do OpenSearch Service. Ao migrar de um banco de dados relacional, como o Oracle ou o MySQL, para o OpenSearch Service, convém reestruturar a forma como você armazena esses dados.

O AWS DMS é compatível com os seguintes tipos de dados escalares do OpenSearch Service:

- Booleano
- Data
- Float
- Int
- String

O AWS DMS converte dados do tipo Date no tipo String. É possível especificar o mapeamento personalizado para interpretar essas datas.

O AWS DMS não é compatível com a migração de tipos de dados de LOB.

Pré-requisitos para utilizar o Amazon OpenSearch Service como destino do AWS Database Migration Service

Antes de começar a trabalhar com um banco de dados OpenSearch Service como destino do AWS DMS, crie um perfil do AWS Identity and Access Management (IAM). Esse perfil deve permitir que o AWS DMS acesse os índices do OpenSearch Service no endpoint de destino. O conjunto mínimo de permissões de acesso é mostrado na seguinte política do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "dms.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}
```

O perfil utilizado para a migração para o OpenSearch Service deve ter as seguintes permissões.

```
{
    "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/*"
        }
    ]
}
```

No exemplo anterior, substitua *region* pelo identificador da região da AWS, *account-id* pelo seu ID de conta da AWS, e *domain-name* pelo nome do domínio do Amazon OpenSearch Service. Um exemplo é `arn:aws:es:us-west-2:123456789012:domain/my-es-domain`

Configurações de endpoint ao utilizar o OpenSearch Service como um destino do AWS DMS

É possível utilizar as configurações de endpoint para configurar o destino do OpenSearch Service de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--elasticsearch-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o OpenSearch Service como destino.

Nome do atributo	Valores válidos	Valor padrão e descrição
`FullLoadErrorPercentage`	Um inteiro positivo maior que 0, mas não maior que 100.	10: para uma tarefa de carga máxima, esse atributo determina o limite de erros permitidos antes que ocorra uma falha na tarefa. Por exemplo, suponha que haja 1.500 linhas no endpoint de origem e o parâmetro foi definido como 10. A tarefa vai falhar se o AWS DMS encontrar mais de 150 erros (10% do total de linhas) ao gravar no endpoint de destino.
`ErrorRetryDuration`	Um inteiro positivo maior do que 0.	300 – Se ocorrer um erro no endpoint de destino, o AWS DMS tentará novamente ao longo deste número de segundos. Caso contrário, a tarefa apresentará falha.

Limitações ao utilizar o Amazon OpenSearch Service como destino do AWS Database Migration Service

Aplicam-se as seguintes limitações ao utilizar o Amazon OpenSearch Service como destino:

- O OpenSearch Service utiliza mapeamento dinâmico (previsão automática) para determinar os tipos de dados a serem utilizados para os dados da migração.
- O OpenSearch Service armazena cada documento com um ID exclusivo. Veja um exemplo a seguir:

  ```
  "_id": "D359F8B537F1888BC71FE20B3D79EAE6674BE7ACA9B645B0279C7015F6FF19FD"
  ```

  Cada ID de documento tem 64 bytes de comprimento; lembre-se de que você precisará desse espaço para armazenamento. Por exemplo, se você migrar 100.000 linhas de uma origem do AWS DMS, o índice do OpenSearch Service resultante precisará de armazenamento para 6.400.000 bytes adicionais.

- Com o OpenSearch Service, não é possível atualizar os atributos da chave primária. Essa restrição é importante quando se utiliza a replicação contínua com a captura de dados de alteração (CDC), pois ela pode gerar dados indesejados no destino. No modo CDC, as chaves primárias são mapeadas para os valores de SHA256, que têm 32 bytes de comprimento. Esses são convertidos em strings de 64 bytes legíveis por seres humanos e utilizados como IDs de documento do OpenSearch Service.
- Se encontrar itens que não podem ser migrados, o AWS DMS gravará as mensagens de erro no Amazon CloudWatch Logs. Esse comportamento é diferente do de outros endpoints de destino do AWS DMS, que gravam os erros em uma tabela de exceções.
- O AWS DMS não é compatível com a conexão a um cluster do Amazon ES que tenha o controle de acesso refinado ativado com usuário mestre e senha.
- O AWS DMS não é compatível com o OpenSearch Service com tecnologia sem servidor.
- O OpenSearch Service não é compatível com a gravação de dados em índices preexistentes.

Tipos de dados de destino do Amazon OpenSearch Service.

Quando o AWS DMS migra os dados de bancos de dados heterogêneos, o serviço mapeia os tipos de dados do banco de dados de origem para tipos de dados intermediários, chamados de tipos de dados do AWS DMS. Em seguida, o serviço mapeia os tipos de dados intermediários para os tipos de dados de destino. A tabela a seguir mostra cada tipo de dados do AWS DMS e o tipo para o qual ele mapeia no OpenSearch Service.

Tipo de dados do AWS DMS	Tipo de dados do OpenSearch Service
Booleano	booleano
Data	string
Time	date
Timestamp	date
INT4	integer
Real4	float
UINT4	integer

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Utilizar o Amazon DocumentDB como destino para o AWS Database Migration Service

Para obter informações sobre as versões do Amazon DocumentDB (compatíveis com o MongoDB) compatíveis com o AWS DMS, consulte Metas para AWS DMS. É possível utilizar o AWS DMS para migrar os dados para o Amazon DocumentDB (compatível com MongoDB) de qualquer mecanismo de dados de origem compatível com o AWS DMS. O mecanismo pode estar em um serviço gerenciado pela AWS, como o Amazon RDS, o Aurora ou o Amazon S3. Ou o mecanismo pode estar em um banco de dados autogerenciado, como o MongoDB em execução no Amazon EC2 ou on-premises.

É possível utilizar o AWS DMS para replicar dados de origem para bancos de dados, coleções ou documentos do Amazon DocumentDB.

> **Note**
>
> Se o endpoint de origem for o MongoDB ou o Amazon DocumentDB, execute a migração no Modo documento.

O MongoDB armazena dados em um formato JSON binário (BSON). O AWS DMS é compatível com todos os tipos de dados BSON compatíveis com o Amazon DocumentDB. Para obter uma lista desses tipos de dados, consulte APIs, operações e tipos de dados do MongoDB compatíveis no Guia do desenvolvedor do Amazon DocumentDB.

Se o endpoint de origem for um banco de dados relacional, o AWS DMS mapeará os objetos do banco de dados para o Amazon DocumentDB da seguinte maneira:

- Um banco de dados relacional ou esquema de banco de dados, é mapeado para um banco de dados Amazon DocumentDB.
- As tabelas dentro de um banco de dados relacional são mapeadas para coleções no Amazon DocumentDB.
- Os registros em uma tabela relacional são mapeados para documentos no Amazon DocumentDB. Cada documento é construído a partir de dados no registro de origem.

Se o endpoint de origem for o Amazon S3, os objetos resultantes do Amazon DocumentDB corresponderão às regras de mapeamento do AWS DMS para o Amazon S3. Por exemplo, considere o URI a seguir.

```
s3://mybucket/hr/employee
```

Neste caso, o AWS DMS mapeia os objetos em `mybucket` para o Amazon DocumentDB da seguinte maneira:

- A parte de nível superior do URI (`hr`) é mapeada para um banco de dados Amazon DocumentDB.
- A próxima parte do URI (`employee`) é mapeada para uma coleção do Amazon DocumentDB.
- Cada objeto em `employee` é mapeado para um documento no Amazon DocumentDB.

Para obter mais informações sobre as regras de mapeamento do Amazon S3, consulte Usando o Amazon S3 como fonte para AWS DMS.

Configurações do endpoint do Amazon DocumentDB

Nas versões 3.5.0 e superiores do AWS DMS, é possível melhorar o desempenho da captura de dados de alteração (CDC) para endpoints do Amazon DocumentDB ajustando as configurações da tarefa para threads paralelos e operações em massa. Para fazer isso, especifique o número de threads simultâneos, as filas por thread e o número de registros a serem armazenados em um buffer utilizando as configurações da tarefa `ParallelApply*`. Por exemplo, suponha que você queira executar uma carga de CDC e aplicar 128 threads em paralelo. Você também quer acessar 64 filas por thread, com 50 registros armazenados por buffer.

Para promover o desempenho da CDC, o AWS DMS é compatível com estas configurações de tarefa:

- `ParallelApplyThreads`: especifica o número de threads simultâneos que AWS DMS utiliza durante uma carga de CDC para enviar registros de dados para um endpoint de destino do Amazon DocumentDB. O valor padrão é zero (0) e o valor máximo é 32.
- `ParallelApplyBufferSize`: especifica o número máximo de registros a serem armazenados em cada fila de buffer para que os threads simultâneos enviem para um endpoint de destino do Amazon DocumentDB durante uma carga de CDC. O valor padrão é 100 e o valor máximo é 1.000. Utilize essa opção quando `ParallelApplyThreads` especificar mais de um thread.

- `ParallelApplyQueuesPerThread`: especifica o número de filas que cada thread acessa para utilizar registros de dados das filas e gerar uma carga em lote para um endpoint do Amazon DocumentDB durante a CDC. O padrão é 1. O máximo é 512.

Para obter mais detalhes sobre como trabalhar com o Amazon DocumentDB como destino do AWS DMS, consulte as seguintes seções:

Tópicos

> **Note**
>
> Para obter uma explicação detalhada do processo de migração, consulte Migração do MongoDB para o Amazon DocumentDB no Guia de migração passo a passo do AWS Database Migration Service.

Mapear dados de uma origem para um destino do Amazon DocumentDB

O AWS DMS lê registros do endpoint de origem e constrói documentos JSON com base nos dados que lê. Para cada documento JSON, o AWS DMS deve determinar um campo `_id` para atuar como um identificador exclusivo. Ele grava o documento JSON em uma coleção do Amazon DocumentDB, utilizando o campo `_id` como uma chave primária.

Dados de origem que são uma coluna individual

Se os dados de origem consistirem em uma única coluna, os dados deverão ser de um tipo de string. (Dependendo do mecanismo de origem, os tipos de dados reais podem ser VARCHAR, NVARCHAR, TEXT, LOB, CLOB ou semelhante.) O AWS DMS presume que os dados são um documento JSON válido e replica os dados para o Amazon DocumentDB no estado em que se encontra.

Se o documento JSON resultante contiver um campo chamado `_id`, o campo será utilizado como o `_id` exclusivo no Amazon DocumentDB.

Se o JSON não contiver um campo `_id`, o Amazon DocumentDB gerará um valor de `_id` automaticamente.

Dados de origem que são várias colunas

Se os dados de origem consistirem em várias colunas, o AWS DMS criará um documento JSON a partir de todas essas colunas. Para determinar o campo `_id` para o documento, o AWS DMS procederá da seguinte maneira:

- Se uma das colunas for chamada `_id`, os dados dessa coluna serão utilizados como o `_id` de destino.
- Se não houver uma coluna `_id`, mas os dados de origem tiverem uma chave primária ou um índice exclusivo, o AWS DMS usará essa chave ou esse valor de índice como o valor de `_id`. Os dados da chave primária ou do índice exclusivo também aparece como campos explícitos no documento JSON.
- Se não houver nenhuma coluna `_id` e nenhuma chave primária ou índice exclusivo, o Amazon DocumentDB gerará um valor de `_id` automaticamente.

Coagir um tipo de dados no endpoint de destino

O AWS DMS pode modificar as estruturas de dados ao gravar em um endpoint de destino do Amazon DocumentDB. É possível solicitar essas alterações renomeando colunas e tabelas no endpoint de origem ou fornecendo regras de transformação que são aplicadas quando uma tarefa está sendo executada.

Utilizar um documento JSON aninhado (json_ prefix)

Para coagir um tipo de dados, é possível prefixar o nome da coluna de origem com `json_` (ou seja, `json_`*`columnName`*) manualmente ou utilizando uma transformação. Nesse caso, a coluna é criada como um documento JSON aninhado dentro do documento de destino, e não como um campo de string.

Por exemplo, suponha que você deseja migrar o documento a seguir de um endpoint de origem do MongoDB.

```
{
```

```
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",
    "ContactDetails": "{"Home": {"Address": "Boston","Phone": "1111111"},"Work":
 { "Address": "Boston", "Phone": "2222222222"}}"
}
```

Se você não coagir nenhum dos tipos de dados de origem, o documento `ContactDetails` incorporado será migrado como uma string.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",
    "ContactDetails": "{\"Home\": {\"Address\": \"Boston\",\"Phone\": \"1111111\"},
\"Work\": { \"Address\": \"Boston\", \"Phone\": \"2222222222\"}}"
}
```

No entanto, é possível adicionar uma regra de transformação para coagir `ContactDetails` para um objeto JSON. Por exemplo, suponha que o nome original da coluna de origem seja `ContactDetails`. Para forçar o tipo de dados como JSON aninhado, a coluna no endpoint de origem precisa ser renomeada como “json_ContactDetails adicionando o prefixo “*json_*” na origem manualmente ou por meio de regras de transformação. Por exemplo, é possível utilizar a regra de transformação abaixo:

```
{
    "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
    }
```

```
    ]
}
```

O AWS DMS replica o campo ContactDetails como JSON aninhado, da seguinte forma.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",
    "ContactDetails": {
        "Home": {
            "Address": "Boston",
            "Phone": "1111111111"
        },
        "Work": {
            "Address": "Boston",
            "Phone": "2222222222"
        }
    }
}
```

Utilizar uma matriz JSON (array_ prefix)

Para coagir um tipo de dados, é possível prefixar o nome de uma coluna com `array_` (ou seja, `array_`*`columnName`*) manualmente ou utilizando uma transformação. Neste caso, o AWS DMS considera a coluna como uma matriz JSON e a cria como tal no documento de destino.

Suponha que você deseja migrar o documento a seguir de um endpoint de origem do MongoDB.

```
{
    "_id" : "1",
    "FirstName": "John",
    "LastName": "Doe",

    "ContactAddresses": ["Boston", "New York"],

    "ContactPhoneNumbers": ["1111111111", "2222222222"]
}
```

Se você não coagir nenhum dos tipos de dados de origem, o documento `ContactDetails` incorporado será migrado como uma string.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",

    "ContactAddresses": "[\"Boston\", \"New York\"]",

    "ContactPhoneNumbers": "[\"1111111111\", \"2222222222\"]"

}
```

No entanto, é possível adicionar regras de transformação para coagir `ContactAddress` e `ContactPhoneNumbers` para matrizes JSON, conforme mostrado na tabela a seguir.

Nome original da coluna de origem	Coluna de origem renomeada
`ContactAddress`	`array_ContactAddress`
`ContactPhoneNumbers`	`array_ContactPhoneNumbers`

O AWS DMS replica `ContactAddress` e `ContactPhoneNumbers` da seguinte maneira.

```
{
    "_id": "1",
    "FirstName": "John",
    "LastName": "Doe",
    "ContactAddresses": [
        "Boston",
        "New York"
    ],
    "ContactPhoneNumbers": [
        "1111111111",
        "2222222222"
    ]
}
```

Conectar-se ao Amazon DocumentDB utilizando TLS

Por padrão, um cluster recém-criado do Amazon DocumentDB aceita conexões seguras somente quando o Transport Layer Security (TLS) é utilizado. Quando o TLS está ativado, cada conexão ao Amazon DocumentDB requer uma chave pública.

É possível recuperar a chave pública do Amazon DocumentDB baixando o arquivo `rds-combined-ca-bundle.pem` de um bucket do Amazon S3 hospedado pela AWS. Para obter mais informações sobre como baixar esse arquivo, consulte Criptografar conexões utilizando TLS no Guia do desenvolvedor do Amazon DocumentDB

Depois de baixar esse arquivo .pem, é possível importar a chave pública que ele contém no AWS DMS conforme descrito a seguir.

AWS Management Console

Para importar o arquivo (.pem) da chave pública

1. Abra o console do AWS DMS em https://console.aws.amazon.com/dms.
2. No painel de navegação, escolha Certificates.
3. Selecione Import certificate (Importar certificado) e faça o seguinte:

   - Para Certificate identifier (Identificador do certificado), insira um nome exclusivo para o certificado, por exemplo `docdb-cert`.
   - Em Importar arquivo, navegue até o local onde você salvou o arquivo .pem.

   Quando estiver satisfeito com as configurações, selecione Add new CA certificate (Adicionar novo certificado de CA).

AWS CLI

Utilize o comando `aws dms import-certificate`, conforme mostrado no exemplo a seguir.

```
aws dms import-certificate \
    --certificate-identifier docdb-cert \
    --certificate-pem file://./rds-combined-ca-bundle.pem
```

Ao criar um endpoint de destino do AWS DMS, forneça o identificador do certificado (por exemplo, `docdb-cert`). Além disso, defina o parâmetro do modo SSL como `verify-full`.

Conexão aos clusters elásticos do Amazon DocumentDB como destino

Nas versões 3.4.7 e superiores do AWS DMS, é possível criar um endpoint de destino do Amazon DocumentDB como um cluster elástico. Se você criar o endpoint de destino como um cluster elástico, precisará anexar um novo certificado SSL ao endpoint do cluster elástico do Amazon DocumentDB porque o certificado SSL existente não funcionará.

Como anexar um novo certificado SSL ao endpoint do cluster elástico do Amazon DocumentDB

1. Em um navegador, abra https://www.amazontrust.com/repository/SFSRootCAG2.pem e salve o conteúdo em um arquivo `.pem` com um nome de arquivo exclusivo, por exemplo `SFSRootCAG2.pem`. Esse é o arquivo de certificado que você precisa importar nas etapas subsequentes.
2. Crie o endpoint do cluster elástico e defina as seguintes opções:
    a. Em Configuração do endpoint, escolha Adicionar novo certificado CA.
    b. Em Identificador de certificado, insira **`SFSRootCAG2.pem`**.
    c. Em Importar arquivo de certificado, escolha Escolher arquivo e navegue até o arquivo `SFSRootCAG2.pem` baixado anteriormente.
    d. Selecione e abra o arquivo `SFSRootCAG2.pem` baixado.
    e. Escolha Importar certificado.
    f. No menu suspenso Escolher um certificado, escolha SFSRootCAG2.pem.

O novo certificado SSL do arquivo `SFSRootCAG2.pem` baixado agora está anexado ao endpoint do cluster elástico do Amazon DocumentDB.

Replicação contínua com o Amazon DocumentDB como destino

Se a replicação contínua (captura de dados de alteração, CDC) estiver ativada para o Amazon DocumentDB como destino, as versões 3.5.0 e superior do AWS DMS proporcionarão uma melhoria no desempenho vinte vezes maior do que nas versões anteriores. Em versões anteriores, em que o AWS DMS trata até 250 registros por segundo, AWS DMS agora trata aproximadamente 5000 registros/segundo. O AWS DMS também garante que os documentos no Amazon DocumentDB permaneçam sincronizados com a origem. Quando um registro da origem é criado ou atualizado, o AWS DMS primeiro determina qual registro do Amazon DocumentDB será afetado fazendo o seguinte:

- Se o registro da origem tiver uma coluna chamada `_id`, o valor dessa coluna determinará o `_id` correspondente na coleção do Amazon DocumentDB.
- Se não houver uma coluna `_id`, mas os dados de origem tiverem uma chave primária ou um índice exclusivo, o AWS DMS usará essa chave ou esse valor de índice como o `_id` da coleção do Amazon DocumentDB.
- Se o registro da origem não tiver uma coluna `_id`, uma chave primária ou um índice exclusivo, o AWS DMS corresponderá todas as colunas de origem aos campos correspondentes na coleção do Amazon DocumentDB.

Quando um novo registro da origem é criado, o AWS DMS grava um documento correspondente no Amazon DocumentDB. Se um registro da origem existente for atualizado, o AWS DMS atualizará os campos correspondentes no documento de destino no Amazon DocumentDB. Todos os campos que existem no documento de destino, mas não no registro da origem permanecem inalterados.

Quando um registro da origem é excluído, o AWS DMS exclui o documento correspondente do Amazon DocumentDB.

Alterações estruturais (DDL) na origem

Com a replicação contínua, qualquer alteração nas estruturas de dados da origem (como tabelas, colunas e assim por diante) é propagada para seus equivalentes no Amazon DocumentDB. Em bancos de dados relacionais, essas alterações são iniciadas utilizando instruções da linguagem de definição de dados (DDL). É possível ver como o AWS DMS propaga essas alterações para o Amazon DocumentDB na tabela a seguir.

DDL na origem	Efeito no destino do Amazon DocumentDB
`CREATE TABLE`	Cria uma coleção vazia.
Instrução que renomeia uma tabela (`RENAME TABLE`, `ALTER TABLE...RENAME` e semelhante)	Renomeia a coleção.
`TRUNCATE TABLE`	Remove todos os documentos da coleção, mas somente se `HandleSourceTableTruncated` for `true`. Para obter mais informações, consulte Configurações de tarefa

DDL na origem	Efeito no destino do Amazon DocumentDB
	para processamento de DDL de processamento de alterações.
`DROP TABLE`	Exclui a coleção, mas somente se `HandleSourceTableDropped` for `true`. Para obter mais informações, consulte Configurações de tarefa para processamento de DDL de processamento de alterações.
Instrução que adiciona uma coluna a uma tabela (`ALTER TABLE...ADD` e semelhante)	A instrução DDL é ignorada e um aviso é emitido. Quando o primeiro `INSERT` é realizado na origem, o novo campo é adicionado ao documento de destino.
`ALTER TABLE...RENAME COLUMN`	A instrução DDL é ignorada e um aviso é emitido. Quando o primeiro `INSERT` é realizado na origem, o campo recém-nomeado é adicionado ao documento de destino.
`ALTER TABLE...DROP COLUMN`	A instrução DDL é ignorada e um aviso é emitido.
Instrução que altera o tipo de dados da coluna (`ALTER COLUMN...MODIFY` e semelhante)	A instrução DDL é ignorada e um aviso é emitido. Quando o primeiro `INSERT` é realizado na origem com o novo tipo de dados, o documento de destino é criado com um campo desse novo tipo de dados.

Limitações da utilização do Amazon DocumentDB como destino

As seguintes limitações se aplicam ao utilizar o Amazon DocumentDB como destino do AWS DMS:

- No Amazon DocumentDB, os nomes de coleção não podem conter o caractere cifrão ($). Além disso, os nomes do banco de dados não podem conter caracteres Unicode.
- O AWS DMS não é compatível com a mesclagem de várias tabelas de origem em uma única coleção do Amazon DocumentDB.

- Quando o AWS DMS processa as alterações de uma tabela de origem que não tem uma chave primária, qualquer colunas LOB na tabela é ignorada.
- Se a opção Alterar tabela estiver ativada, e o AWS DMS encontrar uma coluna de origem chamada "_id", essa coluna aparecerá como "__id" (dois sublinhados) na tabela de alteração.
- Se você escolher o Oracle como o endpoint de origem, a origem do Oracle deverá ter o registro em log suplementar total ativado. Caso contrário, se houver colunas na origem que não foram alteradas, os dados serão carregados no Amazon DocumentDB como valores nulos.
- A configuração de `TargetTablePrepMode:TRUNCATE_BEFORE_LOAD` da tarefa de replicação não é compatível para utilização com um endpoint de destino do DocumentDB.

Utilizar configurações de endpoint com o Amazon DocumentDB como destino

É possível utilizar as configurações de endpoint para configurar o destino do Amazon DocumentDB de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--doc-db-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações do endpoint que é possível utilizar com o Amazon DocumentDB como destino.

Nome do atributo	Valores válidos	Valor padrão e descrição

`replicateShardCollections`	booleano
	
	`true`
	
	`false`
		Quando `true`, essa configuração de endpoint tem os seguintes efeitos e impõe as seguintes limitações:
		• O AWS DMS tem permissão para replicar dados para coleções de fragmentos de destino. Essa configuração só será aplicável se o endpoint do DocumentDB de destino for um cluster elástico.
		• Defina `TargetTablePrepMode` como `DO_NOTHING`.
		• O AWS DMS define automaticamente `useUpdateLookup` como `false` durante a migração.

Tipos de dados de destino do Amazon DocumentDB

Na tabela a seguir, é possível encontrar os tipos de dados de destino do Amazon DocumentDB compatíveis ao utilizar o AWS DMS e o mapeamento padrão dos tipos de dados do AWS DMS. Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipo de dados do AWS DMS	Tipo de dados do Amazon DocumentDB
BOOLEAN	Booleano
BYTES	Dados binários
DATE	Data
TIME	String (UTF8)
DATETIME	Data
INT1	Inteiro de 32 bits
INT2	Inteiro de 32 bits
INT4	Inteiro de 32 bits
INT8	Inteiro de 64 bits
NUMERIC	String (UTF8)
REAL4	Double
REAL8	Double
STRING	Se os dados forem reconhecidos como JSON, o AWS DMS os migrará para o Amazon DocumentDB como um documento. Caso contrário, os dados serão mapeados como String (UTF8).
UINT1	Inteiro de 32 bits
UINT2	Inteiro de 32 bits

Tipo de dados do AWS DMS	Tipo de dados do Amazon DocumentDB
UINT4	Inteiro de 64 bits
UINT8	String (UTF8)
WSTRING	Se os dados forem reconhecidos como JSON, o AWS DMS os migrará para o Amazon DocumentDB como um documento. Caso contrário, os dados serão mapeados como String (UTF8).
BLOB	Binário
CLOB	Se os dados forem reconhecidos como JSON, o AWS DMS os migrará para o Amazon DocumentDB como um documento. Caso contrário, os dados serão mapeados como String (UTF8).
NCLOB	Se os dados forem reconhecidos como JSON, o AWS DMS os migrará para o Amazon DocumentDB como um documento. Caso contrário, os dados serão mapeados como String (UTF8).

Utilizar o Amazon Neptune como destino do AWS Database Migration Service

O Amazon Neptune é um serviço de banco de dados de grafos rápido, confiável e totalmente gerenciado que facilita a criação e a execução de aplicações que trabalham com conjuntos de dados altamente conectados. O recurso principal do Neptune é um mecanismo de banco de dados de grafo com projeto específico e alto desempenho. Esse mecanismo é otimizado para armazenar bilhões de relacionamentos e consultar grafos com latência de milissegundos. O Neptune é compatível com as linguagens de consulta de grafos populares do Apache TinkerPop Gremlin e do SPARQL do W3C. Para obter mais informações sobre o Amazon Neptune, consulte O que é o Amazon Neptune? no Guia do usuário do Amazon Neptune.

Sem um banco de dados de grafo, como o Neptune, é provável que você modele os dados altamente conectados em um banco de dados relacional. Como os dados têm conexões potencialmente dinâmicas, as aplicações que utilizam essas fontes de dados precisam modelar consultas de dados conectadas no SQL. Essa abordagem exigirá que você grave uma camada extra para converter consultas gráficas em SQL. Além disso, os bancos de dados relacionais vêm com rigidez de

esquema. Quaisquer alterações no esquema para modelar conexões alteradas requerem tempo de inatividade e manutenção adicional da conversão da consulta para oferecer suporte ao novo esquema. Além disso, o desempenho da consulta é outra grande restrição a ser considerada ao projetar seus aplicativos.

Os bancos de dados de grafos podem simplificar muito essas situações. Livre de um esquema, uma camada de consulta de grafos avançada (Gremlin ou SPARQL) e índices otimizados para consultas de grafos aumentam a flexibilidade e o desempenho. O banco de dados de grafo Amazon Neptune também tem recursos corporativos, como a criptografia em repouso, uma camada de autorização segura, backups padrão, suporte a multi-AZ, suporte a réplicas de leitura e outros.

Utilizando o AWS DMS, é possível migrar dados relacionais que modelam um grafo altamente conectado a um endpoint de destino do Neptune de um endpoint de origem do DMS para qualquer banco de dados SQL compatível.

Para obter mais detalhes, consulte as informações a seguir.

Tópicos

Visão geral da migração para o Amazon Neptune como destino

Antes de iniciar uma migração para um destino do Neptune, crie os seguintes recursos na sua conta da AWS:

- Um cluster do Neptune para o endpoint de destino.
- Um banco de dados relacional SQL compatível com AWS DMS para o endpoint de origem.
- Um bucket do Amazon S3 para o endpoint de destino. Crie esse bucket do S3 na mesma região da AWS que o cluster do Neptune. O AWS DMS utilizará esse bucket do S3 como armazenamento de arquivos intermediário para os dados de destino que carrega em massa para o banco de dados

Neptune. Para obter mais informações sobre como criar um bucket do S3, consulte Criar um bucket, no Guia do usuário do Amazon Simple Storage Service.

- Um endpoint de nuvem privada virtual (VPC) para o S3 na mesma VPC que o cluster do Neptune.
- Um perfil do AWS Identity and Access Management (IAM) que inclui uma política do IAM. Essa política deve especificar as permissões `GetObject`, `PutObject`, `DeleteObject` e `ListObject` para o bucket do S3 do endpoint de destino. Esse perfil será assumido pelo AWS DMS e pelo Neptune com acesso do IAM ao bucket do S3 de destino e ao banco de dados Neptune. Para obter mais informações, consulte Criar um perfil de serviço do IAM para acessar o Amazon Neptune como destino.

Depois de ter esses recursos, a configuração e a inicialização de uma migração para um destino do Neptune é semelhante a qualquer migração de carga máxima utilizando o console ou a API do DMS. No entanto, uma migração para um destino do Neptune exige algumas etapas exclusivas.

Como migrar um banco de dados relacional do AWS DMS para o Neptune

1. Crie uma instância de replicação conforme descrito em Criar uma instância de replicação.
2. Crie e teste um banco de dados relacional SQL compatível com AWS DMS para o endpoint de origem.
3. Crie e teste o endpoint de destino do banco de dados Neptune.

   Para conectar o endpoint de destino ao banco de dados Neptune, especifique o nome do servidor para o endpoint do cluster do Neptune ou para o endpoint da instância do gravador do Neptune. Além disso, especifique a pasta do bucket do S3 para o AWS DMS armazenar os arquivos intermediários de carga em massa no banco de dados do Neptune.

   Durante a migração, o AWS DMS armazenará todos os dados de destino migrados nesta pasta de bucket do S3 até o tamanho máximo de arquivo especificado. Quando esse armazenamento de arquivos atingir o tamanho máximo, o AWS DMS carregará em massa os dados do S3 armazenados no banco de dados de destino. Ele limpará a pasta para ativar o armazenamento de quaisquer dados de destino adicionais para carregamento posterior no banco de dados de destino. Para obter mais informações sobre como especificar essas configurações, consulte Especificar as configurações do endpoint do Amazon Neptune como destino.
4. Crie uma tarefa de replicação de carga máxima com os recursos criados nas etapas 1 a 3 e siga estas etapas:

a. Utilize o mapeamento da tabela de tarefas como de costume para identificar esquemas, tabelas e visualizações de origem específicos para migrar do banco de dados relacional utilizando as regras de seleção e de transformação apropriadas. Para obter mais informações, consulte Utilizar o mapeamento de tabela para especificar as configurações da tarefa.

b. Especifique os mapeamentos de destino, escolhendo uma das opções a seguir para especificar as regras de mapeamento de tabelas e visualizações de origem para o grafo do banco de dados de destino do Neptune:

   - JSON do Gremlin: para obter informações sobre como utilizar o JSON do Gremlin para carregar um banco de dados Neptune, consulte Formato de dados de carga do Gremlin no Guia do usuário do Amazon Neptune.
   - SPARQL RDB to Resource Description Framework Mapping Language (R2RML): para obter informações sobre como utilizar SPARQL R2RML, consulte a especificação do W3C R2RML: RDB para a linguagem de mapeamento RDF.

c. Execute um dos seguintes procedimentos:

   - Utilizando o console do AWS DMS, especifique opções de mapeamento de grafos com Regras de mapeamento de grafos na página Criar tarefa de migração de banco de dados.
   - Utilizando a API do AWS DMS, especifique essas opções utilizando o parâmetro de solicitação `TaskData` da chamada da API `CreateReplicationTask`.

   Para obter mais informações e exemplos que utilizam JSON de Gremlin e o SPARQL R2RML para especificar regras de mapeamento de grafos, consulte Especificar regras de mapeamento de grafos utilizando Gremlin e R2RML para o Amazon Neptune como destino.

5. Iniciar a replicação para sua tarefa de migração.

Especificar as configurações do endpoint do Amazon Neptune como destino

Para criar ou modificar um endpoint de destino, é possível utilizar o console `CreateEndpoint` ou as operações da API `ModifyEndpoint`.

Para um destino do Neptune no console do AWS DMS, defina as Configurações específicas do endpoint na página do console Criar endpoint ou Modificar endpoint. Para `CreateEndpoint` e `ModifyEndpoint`, especifique os parâmetros de solicitação para a opção `NeptuneSettings`. O exemplo a seguir mostra como fazer isso utilizando a CLI.

```
dms create-endpoint --endpoint-identifier my-neptune-target-endpoint
--endpoint-type target --engine-name neptune
--server-name my-neptune-db.cluster-cspckvklbvgf.us-east-1.neptune.amazonaws.com
--port 8192
--neptune-settings
     '{"ServiceAccessRoleArn":"arn:aws:iam::123456789012:role/myNeptuneRole",
       "S3BucketName":"my-bucket",
       "S3BucketFolder":"my-bucket-folder",
       "ErrorRetryDuration":57,
       "MaxFileSize":100,
       "MaxRetryCount": 10,
       "IAMAuthEnabled":false}‘
```

Aqui, a opção da CLI `--server-name` especifica o nome do servidor para o endpoint do gravador de cluster do Neptune. Ou é possível especificar o nome do servidor para um endpoint de instância do gravador do Neptune.

Os parâmetros de solicitação de opção `--neptune-settings` são:

- `ServiceAccessRoleArn`: (obrigatório) o nome do recurso da Amazon (ARN) do perfil de serviço criado para o endpoint de destino do Neptune. Para obter mais informações, consulte Criar um perfil de serviço do IAM para acessar o Amazon Neptune como destino.
- `S3BucketName`: (obrigatório) o nome do bucket do S3 em que o DMS pode armazenar temporariamente dados de grafo migrados em arquivos .csv antes de carregá-los em massa para o banco de dados de destino Neptune. O DMS mapeia os dados de origem SQL para dados de grafo antes de armazená-los nesses arquivos .csv.
- `S3BucketFolder`: (obrigatório) o caminho de uma pasta em que você deseja que o DMS armazene dados de grafo migrados no bucket do S3 especificado por `S3BucketName`.
- `ErrorRetryDuration`: (opcional) o número de milissegundos de espera do DMS para repetir uma carga de dados de grafo migrados para o banco de dados de destino Neptune antes de gerar um erro. O padrão é 250.
- `MaxFileSize`: (opcional) o tamanho máximo em KB de dados de grafo migrados armazenados em um arquivo .csv antes que o DMS execute a carga em massa dos dados para o banco de dados de destino Neptune. O padrão é 1.048.576 KB (1 GB). Se for bem-sucedido, o DMS limpará o bucket, pronto para armazenar o próximo lote de dados de grafo migrados.
- `MaxRetryCount`: (opcional) o número de vezes para o DMS repetir uma carga em massa de dados de grafo migrados para o banco de dados de destino do Neptune antes de gerar um erro. O padrão é 5.

- `IAMAuthEnabled`: (opcional) se desejar a autorização do IAM ativada para esse endpoint, defina esse parâmetro como `ServiceAccessRoleArn` e anexe o documento de política do IAM apropriado ao perfil de serviço especificado por `true`. O padrão é `false`.

Criar um perfil de serviço do IAM para acessar o Amazon Neptune como destino

Para acessar o Neptune como destino, crie um perfil de serviço utilizando o IAM. Dependendo da configuração do endpoint do Neptune, anexe a esse perfil alguns ou todos os documentos de confiança e de política do IAM. Ao criar o endpoint do Neptune, forneça o ARN desse perfil de serviço. Fazer isso permite que o AWS DMS e o Amazon Neptune tenham permissões para acessar o Neptune e o bucket do Amazon S3 associado.

Se você definir o parâmetro `IAMAuthEnabled` em `NeptuneSettings` como `true` na configuração do endpoint do Neptune, anexe uma política do IAM como a seguinte ao perfil de serviço. Se definir `IAMAuthEnabled` como `false`, é possível ignorar esta política.

```
// 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/*"
            }
        ]
    }
```

A política do IAM anterior permite acesso total ao cluster de destino do Neptune especificado por `Resource`.

Anexe uma política do IAM como a seguinte ao perfil de serviço. Essa política permite que o DMS armazene temporariamente dados de grafos migrados no bucket do S3 criados para carregamento em massa no banco de dados de destino do Neptune.

```
//Policy to access S3 bucket
```

```
{
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "ListObjectsInBucket0",
   "Effect": "Allow",
   "Action": "s3:ListBucket",
   "Resource": [
    "arn:aws:s3:::my-bucket"
   ]
  },
  {
   "Sid": "AllObjectActions",
   "Effect": "Allow",
   "Action": ["s3:GetObject",
    "s3:PutObject",
    "s3:DeleteObject"
   ],

   "Resource": [
    "arn:aws:s3:::my-bucket/"
   ]
  },
  {
   "Sid": "ListObjectsInBucket1",
   "Effect": "Allow",
   "Action": "s3:ListBucket",
   "Resource": [
    "arn:aws:s3:::my-bucket",
    "arn:aws:s3:::my-bucket/"
   ]
  }
 ]
}
```

A política do IAM anterior permite que a sua conta consulte o conteúdo do bucket do S3 (`arn:aws:s3:::my-bucket`) criado para o destino do Neptune. Ela também permite que sua conta opere totalmente no conteúdo de todos os arquivos e pastas do bucket (`arn:aws:s3:::my-bucket/`).

Edite a relação de confiança e anexe o seguinte perfil do IAM ao perfil de serviço para permitir que o AWS DMS e os serviços de banco de dados Amazon Neptune assumam o perfil.

```
{
```

```
  "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"
    }
  ]
}
```

Para obter informações sobre como especificar esse perfil de serviço para o endpoint de destino Neptune, consulte Especificar as configurações do endpoint do Amazon Neptune como destino.

Especificar regras de mapeamento de grafos utilizando Gremlin e R2RML para o Amazon Neptune como destino

As regras de mapeamento de grafos criadas especificam como os dados extraídos de uma origem de banco de dados relacional SQL são carregados em um destino de cluster de banco de dados Neptune. O formato dessas regras de mapeamento difere dependendo de se as regras carregam dados do grafos de propriedades utilizando os dados do Apache TinkerPop Gremlin ou Resource Description Framework (RDF) utilizando R2RML. A seguir, você encontrará informações sobre esses formatos e muito mais.

É possível especificar essas regras de mapeamento ao criar a tarefa de migração utilizando o console ou a API do DMS.

Utilizando o console, especifique essas regras de mapeamento utilizando Regras de mapeamento de grafos na página Criar tarefa de migração de banco de dados. Em Regras de mapeamento de grafos, é possível inserir e editar as regras de mapeamento diretamente utilizando o editor fornecido.

Ou é possível procurar um arquivo que contenha as regras de mapeamento no formato apropriado de mapeamento de grafos.

Utilizando a API, especifique essas opções utilizando o parâmetro de solicitação `TaskData` da chamada de API `CreateReplicationTask`. Defina `TaskData` como o caminho de um arquivo que contém as regras de mapeamento no formato de mapeamento de grafos apropriado.

Regras de mapeamento de grafos para geração de dados de grafo de propriedade utilizando o Gremlin

Utilizando o Gremlin para gerar os dados de grafos de propriedade, especifique um objeto JSON com uma regra de mapeamento para cada entidade de grafo a ser gerada dos dados de origem. O formato deste JSON é definido especificamente para carga em massa do Amazon Neptune. O modelo a seguir mostra como é cada regra nesse objeto:

```
{
    "rules": [
        {
            "rule_id": "(an identifier for this rule)",
            "rule_name": "(a name for this rule)",
            "table_name": "(the name of the table or view being loaded)",
            "vertex_definitions": [
                {
                    "vertex_id_template": "{col1}",
                    "vertex_label": "(the vertex to create)",
                    "vertex_definition_id": "(an identifier for this vertex)",
                    "vertex_properties": [
                        {
                            "property_name": "(name of the property)",
                            "property_value_template": "{col2} or text",
                            "property_value_type": "(data type of the property)"
                        }
                    ]
                }
            ]
        },
        {
            "rule_id": "(an identifier for this rule)",
            "rule_name": "(a name for this rule)",
            "table_name": "(the name of the table or view being loaded)",
            "edge_definitions": [
                {
```

```
                    "from_vertex": {
                        "vertex_id_template": "{col1}",
                        "vertex_definition_id": "(an identifier for the vertex
 referenced above)"
                    },
                    "to_vertex": {
                        "vertex_id_template": "{col3}",
                        "vertex_definition_id": "(an identifier for the vertex
 referenced above)"
                    },
                    "edge_id_template": {
                        "label": "(the edge label to add)",
                        "template": "{col1}_{col3}"
                    },
                    "edge_properties":[
                        {
                            "property_name": "(the property to add)",
                            "property_value_template": "{col4} or text",
                            "property_value_type": "(data type like String, int,
 double)"
                        }
                    ]
                }
            ]
        }
    ]
}
```

A presença de um rótulo de vértice implica que o vértice está sendo criado aqui. Sua ausência implica que o vértice está sendo criado por uma origem diferente, e essa definição está adicionando apenas propriedades de vértice. Especifique quantas definições de vértice e borda forem necessárias para especificar os mapeamentos para toda a origem do banco de dados relacional.

Veja a seguir uma regra de exemplo para uma tabela `employee`.

```
{
    "rules": [
        {
            "rule_id": "1",
            "rule_name": "vertex_mapping_rule_from_nodes",
            "table_name": "nodes",
```

```
            "vertex_definitions": [
                {
                    "vertex_id_template": "{emp_id}",
                    "vertex_label": "employee",
                    "vertex_definition_id": "1",
                    "vertex_properties": [
                        {
                            "property_name": "name",
                            "property_value_template": "{emp_name}",
                            "property_value_type": "String"
                        }
                    ]
                }
            ]
        },
        {
            "rule_id": "2",
            "rule_name": "edge_mapping_rule_from_emp",
            "table_name": "nodes",
            "edge_definitions": [
                {
                    "from_vertex": {
                        "vertex_id_template": "{emp_id}",
                        "vertex_definition_id": "1"
                    },
                    "to_vertex": {
                        "vertex_id_template": "{mgr_id}",
                        "vertex_definition_id": "1"
                    },
                    "edge_id_template": {
                        "label": "reportsTo",
                        "template": "{emp_id}_{mgr_id}"
                    },
                    "edge_properties":[
                        {
                            "property_name": "team",
                            "property_value_template": "{team}",
                            "property_value_type": "String"
                        }
                    ]
                }
            ]
        }
    ]
```

```
}
```

Aqui, as definições de vértice e borda mapeiam uma relação hierárquica de um nó `employee` com ID de funcionário (`EmpID`) e um nó `employee` com um ID de gerente (`managerId`).

Para obter mais informações sobre como criar regras de mapeamento de grafos utilizando JSON do Gremlin, consulte Formato de dados de carga do Gremlin no Guia do usuário do Amazon Neptune.

Regras de mapeamento de grafos para geração de dados RDF/SPARQL

Se você estiver carregando dados RDF a serem consultados utilizando SPARQL, grave as regras de mapeamento de grafos no R2RML. R2RML é uma linguagem W3C padrão de mapeamento de dados relacionais para RDF. Em um arquivo R2RML, um mapa triplo (por exemplo, `<#TriplesMap1>` a seguir) especifica uma regra para transformar cada linha de uma tabela lógica em RDF triplos. Um mapa de assunto (por exemplo, `rr:subjectMap` a seguir) especifica uma regra para gerar os assuntos dos RDF triplos gerados por um mapa triplo. Um mapa de objeto de predicado (por exemplo, qualquer `rr:predicateObjectMap` a seguir) é um perfil que cria um ou mais pares de objetos de predicado para cada linha de uma tabela lógica.

Segue um exemplo simples para uma tabela de `nodes`.

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix ex: <http://example.com/ns#>.

<#TriplesMap1>
    rr:logicalTable [ rr:tableName "nodes" ];
    rr:subjectMap [
        rr:template "http://data.example.com/employee/{id}";
        rr:class ex:Employee;
    ];
    rr:predicateObjectMap [
        rr:predicate ex:name;
        rr:objectMap [ rr:column "label" ];
    ]
```

No exemplo anterior, o mapeamento define nós de grafo mapeados a partir de uma tabela de funcionários.

Veja outro exemplo simples de tabela `Student`.

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix ex: <http://example.com/#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<#TriplesMap2>
    rr:logicalTable [ rr:tableName "Student" ];
    rr:subjectMap   [ rr:template "http://example.com/{ID}{Name}";
                      rr:class foaf:Person ];
    rr:predicateObjectMap [
        rr:predicate ex:id ;
        rr:objectMap  [ rr:column "ID";
                        rr:datatype xsd:integer ]
    ];
    rr:predicateObjectMap [
        rr:predicate foaf:name ;
        rr:objectMap  [ rr:column "Name" ]
    ].
```

No exemplo anterior, o mapeamento define nós de grafo que mapeiam relações de terceiros entre pessoas de uma tabela `Student`.

Para obter mais informações sobre como criar regras de mapeamento de grafos utilizando SPARQL R2RML, consulte a especificação W3C R2RML: RDB to RDF Mapping Language.

Tipos de dados para migração de Gremlin e R2RML para o Amazon Neptune como destino

O AWS DMS executa o mapeamento do tipo de dados do endpoint de origem do SQL para o Neptune de destino de duas formas. O modo utilizado depende do formato do mapeamento de grafos utilizado para carregar o banco de dados Neptune:

- Apache TinkerPop Gremlin, utilizando uma representação JSON dos dados de migração.
- SPARQL do W3C, utilizando uma representação R2RML dos dados de migração.

Para obter mais informações sobre esses dois formatos de mapeamento de grafos, consulte Especificar regras de mapeamento de grafos utilizando Gremlin e R2RML para o Amazon Neptune como destino.

Veja a seguir descrições dos mapeamentos de tipos de dados para cada formato.

Mapeamentos de tipos de dados de origem SQL para destino Gremlin

A tabela a seguir mostra os mapeamentos de tipos de dados de uma origem SQL para um destino Gremlin formatado.

O AWS DMS mapeia qualquer tipo de dados de origem SQL não listado para um Gremlin `String`.

Tipos de dados de origem SQL	Tipos de dados de destino Gremlin
`NUMERIC` (e variantes)	`Double`
`DECIMAL`	
`TINYINT`	`Byte`
`SMALLINT`	`Short`
`INT, INTEGER`	`Int`
`BIGINT`	`Long`
`FLOAT`	`Float`
`DOUBLE PRECISION`	
`REAL`	`Double`
`BIT`	`Boolean`
`BOOLEAN`	
`DATE`	`Date`
`TIME`	
`TIMESTAMP`	
`CHARACTER` (e variantes)	`String`

Para obter mais informações sobre os tipos de dados do Gremlin para carregamento do Neptune, consulte Tipos de dados do Gremlin no Guia do usuário do Neptune.

Mapeamentos de tipos de dados de origem SQL para destino R2RML (RDF)

A tabela a seguir mostra os mapeamentos de tipos de dados de uma origem SQL para um destino R2RML formatado.

Todos os tipos de dados RDF listados fazem distinção de maiúsculas e minúsculas, exceto RDF literal. O AWS DMS mapeia qualquer tipo de dados de origem SQL não listado em um RDF literal.

Um RDF literal é uma de muitas formas e tipos de dados lexicais e literais. Para obter mais informações, consulte RDF Literals na especificação W3C Resource Description Framework (RDF): conceitos e sintaxe abstrata.

Tipos de dados de origem SQL	Tipos de dados de destino R2RML (RDF)
`BINARY` (e variantes)	`xsd:hexBinary`
`NUMERIC` (e variantes)	`xsd:decimal`
`DECIMAL`	
`TINYINT`	`xsd:integer`
`SMALLINT`	
`INT`, `INTEGER`	
`BIGINT`	
`FLOAT`	`xsd:double`
`DOUBLE PRECISION`	
`REAL`	
`BIT`	`xsd:boolean`
`BOOLEAN`	
`DATE`	`xsd:date`

Tipos de dados de origem SQL	Tipos de dados de destino R2RML (RDF)
`TIME`	`xsd:time`
`TIMESTAMP`	`xsd:dateTime`
`CHARACTER` (e variantes)	RDF literal

Para obter mais informações sobre os tipos de dados RDF para carregamento do Neptune e de seus mapeamentos para tipos de dados de origem SQL, consulte Conversões de tipo de dados na especificação do W3C R2RML: RDB para linguagem de mapeamento RDF.

Limitações da utilização do Amazon Neptune como destino

Aplicam-se as seguintes limitações ao utilizar o Neptune como destino:

- Atualmente, o AWS DMS é compatível com tarefas de carga máxima somente para migração para um destino do Neptune. A migração de captura de dados de alterações (CDC) para um destino do Neptune não é compatível.
- Verifique se o banco de dados Neptune de destino é limpo manualmente de todos os dados antes de iniciar a tarefa de migração, como nos exemplos a seguir.

  Para descartar todos os dados (vértices e bordas) dentro do grafo, execute o seguinte comando do Gremlin.

  ```
  gremlin> g.V().drop().iterate()
  ```

  Para descartar vértices com o rótulo `'customer'`, execute o seguinte comando do Gremlin.

  ```
  gremlin> g.V().hasLabel('customer').drop()
  ```

  > **Note**
  >
  > A remoção de um grande conjunto de dados pode levar algum tempo. Talvez você queira iterar `drop()` com uma limitação, por exemplo, `limit(1000)`.

  Para descartar bordas com o rótulo `'rated'`, execute o seguinte comando do Gremlin.

```
gremlin> g.E().hasLabel('rated').drop()
```

> **Note**
>
> A remoção de um grande conjunto de dados pode levar algum tempo. Talvez você queira iterar `drop()` com uma limitação, por exemplo, `limit(1000)`.

- A operação da API do DMS `DescribeTableStatistics` pode retornar resultados imprecisos sobre determinada tabela devido à natureza das estruturas de dados do grafo do Neptune.

  Durante a migração, o AWS DMS verifica cada tabela de origem e utiliza o mapeamento de grafos para converter os dados de origem em um grafo do Neptune. Os dados convertidos são armazenados primeiro na pasta de bucket do S3 especificada para o endpoint de destino. Se a origem for verificada e esses dados intermediários do S3 forem gerados com êxito, `DescribeTableStatistics` pressupõe que os dados foram carregados com êxito no banco de dados de destino do Neptune. Entretanto, isso nem sempre é verdade. Para verificar se os dados foram carregados corretamente para determinada tabela, compare os valores de retorno `count()` nas extremidades da migração para essa tabela.

  No exemplo a seguir, o AWS DMS carregou uma tabela `customer` do banco de dados de origem, à qual é atribuído o rótulo `'customer'` no grafo do banco de dados de destino do Neptune. É possível certificar-se de que esse rótulo seja gravado no banco de dados de destino. Para isso, compare o número de linhas do `customer` disponíveis no banco de dados de origem com o número de linhas rotuladas do `'customer'` carregadas no banco de dados de destino do Neptune após a conclusão da tarefa.

  Para obter o número de linhas disponíveis do cliente no banco de dados de origem utilizando SQL, execute o seguinte procedimento.

  ```
  select count(*) from customer;
  ```

  Para obter o número de linhas rotuladas `'customer'` carregadas no grafo do banco de dados de destino utilizando Gremlin, execute o seguinte procedimento.

  ```
  gremlin> g.V().hasLabel('customer').count()
  ```

- Atualmente, se uma única tabela não for carregada, toda a tarefa falhará. Diferentemente de um banco de dados relacional de destino, os dados no Neptune são altamente conectados, o que torna impossível, em muitos casos, retomar uma tarefa. Se uma tarefa não puder ser retomada com êxito devido a esse tipo de falha de carga de dados, crie uma nova tarefa para carregar a tabela que apresentou falha no carregamento. Antes de executar essa nova tarefa, limpe manualmente a tabela parcialmente carregada do destino do Neptune.

  > **Note**
  >
  > É possível retomar uma tarefa com falha na migração para um destino do Neptune se a falha for recuperável (por exemplo, um erro de trânsito de rede).

- O AWS DMS é compatível com a maioria dos padrões de R2RML. No entanto, o AWS DMS não é compatível com determinados padrões R2RML, incluindo expressões inversas, junções e visualizações. Uma solução alternativa para uma exibição R2RML é criar uma exibição SQL personalizada correspondente no banco de dados de origem. Na tarefa de migração, utilize o mapeamento de tabela para escolher a exibição como entrada. Mapeie a exibição para uma tabela que seja então consumida pelo R2RML para gerar dados de grafo.
- Ao migrar dados de origem com tipos de dados SQL não é compatível com dos, os dados de destino resultantes podem ter perda de precisão. Para obter mais informações, consulte Tipos de dados para migração de Gremlin e R2RML para o Amazon Neptune como destino.
- O AWS DMS não é compatível com a migração de dados LOB para um destino do Neptune.

Utilizar o Redis como destino do AWS Database Migration Service

O Redis é um armazenamento de estrutura de dados de código aberto na memória utilizado como um banco de dados, cache e agente de mensagens. O gerenciamento de dados na memória pode resultar em operações de leitura ou gravação que demoram menos de um milissegundo e centenas de milhões de operações executadas a cada segundo. Como um armazenamento de dados na memória, o Redis capacita as aplicações que exigem tempos de resposta inferiores a um milissegundo.

Ao utilizar o AWS DMS, é possível migrar dados de qualquer banco de dados de origem compatível para um armazenamento de dados Redis de destino com de tempo de inatividade mínimo. Para obter informações adicionais sobre o Redis, consulte a Documentação do Redis.

Além do Redis on-premises, o AWS Database Migration Service é compatível com o seguinte:

- O Amazon ElastiCache para Redis como um armazenamento de dados de destino. O ElastiCache para Redis funciona com os clientes do Redis e utiliza o formato aberto de dados do Redis para armazenar os dados.
- Amazon MemoryDB para Redis como armazenamento de dados de destino. O MemoryDB é compatível com o Redis e permite criar aplicações utilizando todas as estruturas de dados, APIs e comandos do Redis em uso.

Para obter mais detalhes sobre como trabalhar com o Redis como destino do AWS DMS, consulte as seguintes seções:

Tópicos

Pré-requisitos para utilizar um cluster do Redis como destino do AWS DMS

O DMS é compatível com um destino do Redis on-premises em uma configuração independente ou como um cluster do Redis em que os dados são automaticamente fragmentados em vários nós. A fragmentação é o processo de separar os dados em blocos menores, chamados de fragmentos, que são espalhados por vários servidores ou nós. Na verdade, um fragmento é uma partição de dados que contém um subconjunto do conjunto total de dados e serve como uma fatia da workload geral.

Como o Redis é um armazenamento de dados NoSQL de chave-valor, a convenção de nomenclatura de chaves do Redis a ser utilizada quando a origem for um banco de dados relacional é schema-name.table-name.primary-key. No Redis, a chave e o valor não devem conter o caractere especial %. Caso contrário, o DMS ignorará o registro.

> **Note**
>
> Se estiver utilizando o ElastiCache para Redis como destino, o DMS será compatível somente com as configurações Modo de cluster ativado. Para obter mais informações sobre como utilizar o ElastiCache para Redis versão 6.x ou superior para criar um armazenamento de dados de destino ativado para o modo de cluster, consulte Introdução no Guia do usuário do Amazon ElastiCache para Redis.

Antes de começar a migração de um banco de dados, inicie o cluster do Redis com os seguintes critérios.

- O cluster tem um ou mais fragmentos.
- Se estiver utilizando um destino do ElastiCache para Redis, verifique se o cluster não utiliza o controle de acesso baseado em perfis do IAM. Em vez disso, utilize o Redis Auth para autenticar usuários.
- Ative multi-AZ (zonas de disponibilidade).
- Verifique se o cluster tem memória suficiente disponível para comportar os dados a serem migrados do banco de dados.
- Verifique se todos os dados do cluster do Redis de destino estão limpos antes de iniciar a tarefa de migração inicial.

Determine os requisitos de segurança da migração de dados antes de criar a configuração do cluster. O DMS é compatível com a migração para grupos de replicação de destino, independentemente de sua configuração de criptografia. Mas é possível ativar ou desativar a criptografia somente ao criar a configuração do cluster.

Limitações da utilização do Redis como destino do AWS Database Migration Service

Aplicam-se as seguintes limitações ao utilizar o Redis como destino:

- Como o Redis é um armazenamento de dados no-sql de chave-valor, a convenção de nomenclatura de chaves do Redis a ser utilizada quando a origem for um banco de dados relacional é `schema-name.table-name.primary-key`.
- No Redis, a chave-valor não pode conter o caractere especial `%`. Caso contrário, o DMS ignorará o registro.
- O DMS não migrará linhas que contenham caracteres especiais.
- O DMS não migrará campos que contenham caracteres especiais no nome do campo.
- O modo LOB completo não é compatível.
- Uma Autoridade de certificação (CA) privada não é compatível ao utilizar o ElastiCache para Redis como destino.

Migrar dados de um banco de dados relacional ou não relacional para um destino do Redis

É possível migrar dados de qualquer armazenamento de dados SQL ou NoSQL de origem diretamente para um destino do Redis. A configuração e o início de uma migração para um destino do Redis é semelhante a qualquer migração de carga máxima e de captura de dados de alteração e utiliza o console ou a API do DMS. Para executar uma migração de banco de dados para um destino do Redis, faça o seguinte.

- Crie uma instância de replicação para executar todos os processos da migração. Para obter mais informações, consulte Criar uma instância de replicação.
- Especifique um endpoint de origem. Para obter mais informações, consulte Criar endpoints de origem e de destino.
- Localize o nome DNS e o número da porta do cluster.
- Baixe um pacote de certificado que pode ser utilizado para verificar conexões SSL.
- Especifique um endpoint de destino, conforme descrito abaixo.
- Crie uma tarefa ou um conjunto de tarefas para definir as tabelas e os processos de replicação a serem utilizados. Para obter mais informações, consulte Criar uma tarefa.
- Migre dados do banco de dados de origem para o cluster de destino.

Você inicia uma migração do banco de dados de uma duas maneiras:

1. É possível escolher o console do AWS DMS e executar cada etapa lá.
2. É possível utilizar o AWS Command Line Interface (AWS CLI). Para obter mais informações sobre como utilizar a CLI com o AWS DMS, consulte AWS CLI para AWS DMS.

Como localizar o nome DNS e o número da porta do cluster

- Utilize o comando AWS CLI a seguir para fornecer `replication-group-id` com o nome do grupo de replicação:

```
aws elasticache describe-replication-groups --replication-group-id myreplgroup
```

Aqui, a saída mostra o nome DNS no atributo `Address` e o número da porta no atributo `Port` do nó primário no cluster.

```
 ...
"ReadEndpoint": {
"Port": 6379,
"Address": "myreplgroup-
111.1abc1d.1111.uuu1.cache.example.com"
}
...
```

Se estiver utilizando o MemoryDB para Redis como destino, utilize o comando AWS CLI a seguir para fornecer um endereço de endpoint ao cluster do Redis.

```
aws memorydb describe-clusters --clusterid clusterid
```

Baixe um pacote de certificado a ser utilizado para verificar as conexões SSL.

- Insira o comando `wget` na linha de comando. O Wget é uma ferramenta utilitária de linha de comando GNU gratuita utilizada para baixar arquivos na internet.

  ```
  wget https://s3.aws-api-domain/rds-downloads/rds-combined-ca-bundle.pem
  ```

  Aqui, *`aws-api-domain`* completa o domínio do Amazon S3 na região da AWS necessária para acessar o bucket do S3 especificado e o arquivo rds-combined-ca-bundle.pem que ele fornece.

Como criar um endpoint de destino utilizando o console do AWS DMS

Esse endpoint é para o destino do Redis que já está em execução.

- No console, escolha Endpoints no painel de navegação e escolha Criar endpoint. A tabela a seguir descreve as configurações.

Para esta opção	Faça o seguinte
Endpoint type	Escolha o tipo de endpoint de Destino.
Identificador de endpoint	Insira o nome do endpoint. Por exemplo, inclua o tipo de endpoint no nome, como **`my-redis-target`** .
Mecanismo de destino	Escolha Redis como o tipo de mecanismo de banco de dados ao qual você deseja que esse endpoint se conecte.
Nome do cluster	Insira o nome DNS do cluster do Redis.
Porta	Insira o número da porta do cluster do Redis.
Protocolo de segurança SSL	Escolha Texto sem formatação ou Criptografia SSL. Texto sem formatação: essa opção não fornece a criptografia Transport Layer Security (TLS) para o tráfego entre o endpoint e o banco de dados. Criptografia SSL: se escolher essa opção, insira o ARN do certificado da Autoridade de Certificação (CA) SSL para verificar o certificado do servidor e fazer uma conexão criptografada. No Redis on-premises, o DMS é compatível com Autoridades de Certificação (CAs) pública e privada. No ElastiCache para Redis, o DMS é compatível somente com uma CA pública.

Para esta opção	Faça o seguinte
Tipo de autenticação	Escolha o tipo de autenticação a ser executada ao conectar-se ao Redis. As opções incluem Nenhuma, Função de autenticação e Token de autenticação. Se você escolher Perfil de autenticação, forneça um Nome de usuário de autenticação e uma Senha de autenticação. Se você escolher o token de autenticação, forneça somente uma Senha de autenticação.
Instância de replicação	[Opcional] Somente se você quiser testar a conexão, escolha o nome da instância de replicação inserida anteriormente na página Criar instância de replicação.

Depois de você concluir o fornecimento de todas as informações do endpoint, o AWS DMS cria o endpoint de destino do Redis para utilização durante a migração do banco de dados.

Para obter informações sobre como criar uma tarefa de migração e iniciar a migração do banco de dados, consulte Criar uma tarefa.

Especificar as configurações de endpoint para o Redis como destino

Para criar ou modificar um endpoint de destino, é possível utilizar o console `CreateEndpoint` ou as operações da API `ModifyEndpoint`.

Para um destino do Redis no console do AWS DMS, defina as Configurações específicas do Endpoint na página do console Criar endpoint ou Modificar endpoint.

Ao utilizar as operações da API `CreateEndpoint` e `ModifyEndpoint`, especifique os parâmetros de solicitação para a opção `RedisSettings`. O exemplo a seguir mostra como fazer isso utilizando a AWS CLI.

```
aws dms create-endpoint --endpoint-identifier my-redis-target
--endpoint-type target --engine-name redis --redis-settings
'{"ServerName":"sample-test-
sample.zz012zz.cluster.eee1.cache.bbbxxx.com","Port":6379,"AuthType":"auth-token",
```

```
 "SslSecurityProtocol":"ssl-encryption", "AuthPassword":"notanactualpassword"}'

{
    "Endpoint": {
        "EndpointIdentifier": "my-redis-target",
        "EndpointType": "TARGET",
        "EngineName": "redis",
        "EngineDisplayName": "Redis",
        "TransferFiles": false,
        "ReceiveTransferredFiles": false,
        "Status": "active",
        "KmsKeyId": "arn:aws:kms:us-east-1:999999999999:key/x-b188188x",
        "EndpointArn": "arn:aws:dms:us-
east-1:555555555555:endpoint:ABCDEFGHIJKLMONOPQRSTUVWXYZ",
        "SslMode": "none",
        "RedisSettings": {
            "ServerName": "sample-test-sample.zz012zz.cluster.eee1.cache.bbbxxx.com",
            "Port": 6379,
            "SslSecurityProtocol": "ssl-encryption",
            "AuthType": "auth-token"
        }
    }
}
```

Os parâmetros `--redis-settings` são os seguintes:

- `ServerName`: (obrigatório) do tipo `string`, especifica o cluster do Redis para o qual os dados serão migrados e está na sua mesma VPC.
- `Port`: (obrigatório) do tipo `number`, o valor da porta utilizada para acessar o endpoint.
- `SslSecurityProtocol`: (opcional) os valores válidos incluem `plaintext` e `ssl-encryption`. O padrão é `ssl-encryption`.

  A opção `plaintext` não fornece a criptografia Transport Layer Security (TLS) para o tráfego entre o endpoint e o banco de dados.

  Utilize `ssl-encryption` para fazer uma conexão criptografada. `ssl-encryption` não exige o ARN de uma Autoridade de Certificação (CA) SSL para verificar o certificado de um servidor, mas um pode ser identificado opcionalmente utilizando a configuração `SslCaCertificateArn`. Se o ARN de autoridade de certificação não for fornecido, o DMS utilizará a CA raiz da Amazon.

Ao utilizar um destino do Redis on-premises, é possível utilizar `SslCaCertificateArn` para importar a Autoridade de certificação (CA) pública ou privada para o DMS e fornecer esse ARN para autenticação do servidor. Uma CA privada não é compatível com a utilização do ElastiCache para Redis como destino.

- `AuthType`: (obrigatório) indica o tipo de autenticação a ser executada ao conectar-se ao Redis. Os valores válidos são `none`, `auth-token` e `auth-role`.

  A opção `auth-token` exige que uma "*`AuthPassword`*" seja fornecida, enquanto a opção `auth-role` exige que "*`AuthUserName"`* e "*`AuthPassword`*" sejam fornecidos.

Utilizar o Babelfish como destino do AWS Database Migration Service

É possível migrar dados de um banco de dados de origem Microsoft SQL Server para um destino Babelfish utilizando o AWS Database Migration Service.

O Babelfish for Aurora PostgreSQL amplia seu banco de dados Amazon Aurora, edição compatível com PostgreSQL, com a capacidade de aceitar conexões de banco de dados provenientes de clientes Microsoft SQL Server. Fazer isso permite que aplicações originalmente criadas para o SQL Server funcionem diretamente com o Aurora PostgreSQL, com poucas alterações de código em comparação com uma migração tradicional e sem alterar drivers de bancos de dados.

Para obter informações sobre as versões do Babelfish compatíveis com o AWS DMS como destino, consulte Metas para AWS DMS. As versões anteriores do Babelfish no Aurora PostgreSQL exigem um upgrade antes de utilizar o endpoint do Babelfish.

> **Note**
>
> O endpoint de destino do Aurora PostgreSQL é a forma preferida de migrar dados para o Babelfish. Para obter mais informações, consulte Utilizar o Babelfish para Aurora PostgreSQL como destino.

Para obter informações sobre como utilizar o Babelfish como um endpoint de banco de dados, consulte Babelfish para Aurora PostgreSQL no Guia do usuário do Amazon Aurora para Aurora

Pré-requisitos para a utilização do Babelfish como destino do AWS DMS

Crie as tabelas antes de migrar os dados para garantir que o AWS DMS utilize os tipos de dados e metadados de tabela corretos. Se você não criar as tabelas no destino antes de executar a migração, o AWS DMS poderá criar as tabelas com tipos de dados e permissões incorretos. Por exemplo, o AWS DMS cria uma coluna de timestamp como binária(8) em vez disso, e não fornece a funcionalidade de timestamp/rowversion esperada.

Como preparar e criar as tabelas antes da migração

1. Execute as instruções DDL de criação de tabela que incluam quaisquer restrições exclusivas, chaves primárias ou restrições padrão.

   Não inclua restrições de chave estrangeira nem instruções DDL para objetos, como visualizações, procedimentos armazenados, perfis ou acionadores. É possível aplicá-las depois de migrar o banco de dados de origem.
2. Identifique quaisquer colunas de identidade, colunas computadas ou colunas contendo tipos de dados rowversion ou timestamp nas tabelas. Crie as regras de transformação necessárias para lidar com problemas conhecidos ao executar a tarefa de migração. Para obter mais informações, consulte, Regras de transformação e ações.
3. Identifique colunas com tipos de dados não compatíveis com o Babelfish. Altere as colunas afetadas na tabela de destino para utilizar os tipos de dados compatíveis ou crie uma regra de transformação que os remova durante a tarefa de migração. Para obter mais informações, consulte, Regras de transformação e ações.

   A tabela a seguir lista os tipos de dados de origem não compatíveis com o Babelfish e o tipo de dados de destino recomendado correspondente a ser utilizado.

Tipo de dados de origem	Tipo de dados do Babelfish recomendado
HEIRARCHYID	NVARCHAR(250)
GEOMETRY	VARCHAR(MAX)
GEOGRAPHY	VARCHAR(MAX)

Como definir o nível de unidades de capacidade do Aurora (ACUs) para o banco de dados de origem do Aurora PostgreSQL Sem Servidor V2

É possível melhorar o desempenho da tarefa de migração do AWS DMS antes de executá-la definindo o valor mínimo de ACU.

- Na janela Configurações da capacidade do Sem Servidor v2, defina as ACUs mínimas como **2** ou como um nível razoável para o cluster do banco de dados Aurora.

  Para obter mais informações sobre as unidades de capacidade do Aurora, consulte Escolher o intervalo de capacidade do Aurora Sem Servidor v2 para um cluster do Aurora no Guia do usuário do Amazon Aurora.

Depois de executar a tarefa de migração do AWS DMS, é possível redefinir o valor mínimo de ACUs como um nível razoável para o banco de dados de origem do Aurora PostgreSQL Sem Servidor V2.

Requisitos de segurança ao utilizar o Babelfish como destino do AWS Database Migration Service

Veja a seguir a descrição dos requisitos de segurança para utilizar o AWS DMS com um destino do Babelfish:

- O nome de usuário do administrador (o usuário administrador) utilizado para criar o banco de dados.
- Login e usuário do PSQL com permissões suficientes para SELECT, INSERT, UPDATE, DELETE e REFERENCES.

Permissões de usuário para utilizar o Babelfish como destino do AWS DMS

> **Important**
>
> Para fins de segurança, a conta de usuário utilizada para a migração de dados deve ser um usuário registrado em qualquer banco de dados Babelfish que você utilize como destino.

O endpoint de destino do Babelfish requer permissões mínimas de usuário para executar uma migração do AWS DMS.

Como criar um login e um usuário do Transact-SQL (T-SQL) com poucos privilégios

1. Crie um login e uma senha a serem utilizados ao se conectar ao servidor.

```
CREATE LOGIN dms_user WITH PASSWORD = 'password';
GO
```

2. Crie o banco de dados virtual para o cluster do Babelfish.

```
CREATE DATABASE my_database;
GO
```

3. Crie o usuário T-SQL para o banco de dados de destino.

```
USE my_database
GO
CREATE USER dms_user FOR LOGIN dms_user;
GO
```

4. Para cada tabela no banco de dados Babelfish, permissões de GRANT para as tabelas.

```
GRANT SELECT, DELETE, INSERT, REFERENCES, UPDATE ON [dbo].[Categories] TO dms_user;
```

Limitações da utilização do Babelfish como destino do AWS Database Migration Service

As seguintes limitações se aplicam à utilização de um banco de dados Babelfish como destino do AWS DMS:

- Somente o modo de preparação de tabela “Não executar nenhuma ação” é compatível.
- O tipo de dados ROWVERSION exige uma regra de mapeamento de tabela que remove o nome da coluna da tabela durante a tarefa de migração.
- O tipo de dados sql_variant não é compatível.
- O modo LOB completo é compatível. A utilização do SQL Server como um endpoint de origem exige a configuração `ForceFullLob=True` do Atributo de conexão do endpoint do SQL Server definida para que os LOBs sejam migrados para o endpoint de destino.
- As configurações da tarefa de replicação têm as seguintes limitações:

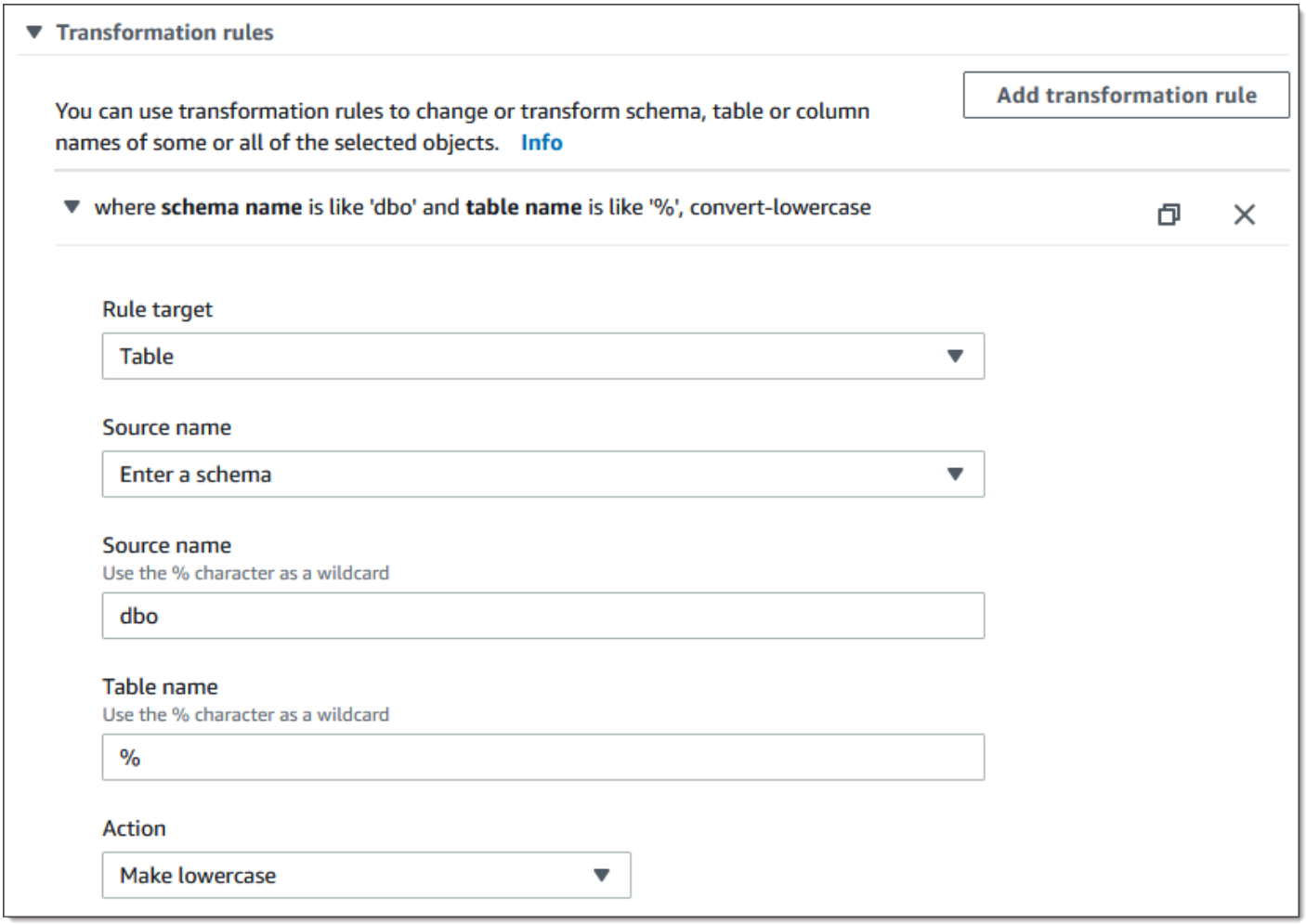
```
{
   "FullLoadSettings": {
      "TargetTablePrepMode": "DO_NOTHING",
      "CreatePkAfterFullLoad": false,
      }.

}
```

- Os tipos de dados TIME(7), DATETIME2(7) e DATETIMEOFFSET(7) no Babelfish limitam o valor de precisão da parte de segundos do tempo a seis dígitos. Considere utilizar um valor de precisão de seis para a tabela de destino ao utilizar esses tipos de dados. Nas versões 2.2.0 e superiores do Babelfish, ao utilizar TIME(7) e DATETIME2(7), o sétimo dígito de precisão é sempre zero.
- No modo DO_NOTHING, o DMS verifica se a tabela já existe. Se a tabela não existir no esquema de destino, o DMS criará a tabela com base na definição da tabela de origem e mapeará todos os tipos de dados definidos pelo usuário para o tipo de dados base.
- Uma tarefa de migração do AWS DMS para um destino Babelfish não é compatível com tabelas que tenham colunas que utilizam os tipos de dados ROWVERSION ou TIMESTAMP. É possível utilizar uma regra de mapeamento de tabela que remove o nome da coluna da tabela durante o processo de transferência. No exemplo de regra de transformação a seguir, uma tabela nomeada `Actor` na origem é transformada para remover todas as colunas que começam com os caracteres `Actor` da tabela `col` no destino.

```
{
  "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%"
 }
}]
}
```

- Para tabelas com colunas de identidade ou computadas, em que as tabelas de destino utilizam nomes com maiúsculas e minúsculas, como Categorias, crie uma ação de regra de transformação que converta os nomes das tabelas em minúsculas para a tarefa do DMS. O exemplo a seguir mostra como criar a ação da regra de transformação, Colocar em letras minúsculas, utilizando o console do AWS DMS. Para obter mais informações, consulte Regras de transformação e ações.



- Antes da versão 2.2.0 do Babelfish, o DMS limitava o número de colunas que você podia replicar para um endpoint de destino do Babelfish a vinte colunas. Com o Babelfish 2.2.0, o limite aumentou para cem colunas. Mas com as versões 2.4.0 e superiores do Babelfish, o número de

colunas que é possível replicar aumenta novamente. É possível executar o exemplo de código a seguir no banco de dados SQL Server para determinar quais tabelas são muito longas.

```
USE myDB;
GO
DECLARE @Babelfish_version_string_limit INT = 8000; -- Use 380 for Babelfish versions
 before 2.2.0
WITH bfendpoint
AS (
SELECT
 [TABLE_SCHEMA]
      ,[TABLE_NAME]
   , COUNT( [COLUMN_NAME] ) AS NumberColumns
   , ( SUM( LEN( [COLUMN_NAME] ) + 3)
  + SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
     + LEN( TABLE_SCHEMA ) + 3
  + 12 -- INSERT INTO string
  + 12)  AS InsertIntoCommandLength -- values string
      , CASE WHEN ( SUM( LEN( [COLUMN_NAME] ) + 3)
  + SUM( LEN( FORMAT(ORDINAL_POSITION, 'N0') ) + 3 )
     + LEN( TABLE_SCHEMA ) + 3
  + 12 -- INSERT INTO string
  + 12)  -- values string
   >= @Babelfish_version_string_limit
   THEN 1
   ELSE 0
  END AS IsTooLong
FROM [INFORMATION_SCHEMA].[COLUMNS]
GROUP BY [TABLE_SCHEMA], [TABLE_NAME]
)
SELECT *
FROM bfendpoint
WHERE IsTooLong = 1
ORDER BY TABLE_SCHEMA, InsertIntoCommandLength DESC, TABLE_NAME
;
```

Tipos de dados de destino do Babelfish

A tabela a seguir mostra os tipos de dados de destino do Babelfish que são compatíveis ao utilizar o AWS DMS e o mapeamento padrão dos tipos de dados do AWS DMS.

Para obter mais informações sobre os tipos de dados do AWS DMS, consulte Tipos de dados do AWS Database Migration Service.

Tipo de dados do AWS DMS	Tipo de dados do Babelfish
BOOLEAN	TINYINT
BYTES	VARBINARY(tamanho)
DATE	DATE
TIME	TIME
INT1	SMALLINT
INT2	SMALLINT
INT4	INT
INT8	BIGINT
NUMERIC	NUMERIC(p,s)
REAL4	REAL
REAL8	FLOAT
STRING	Se a coluna for de data ou hora, faça o seguinte: • No SQL Server 2008 e superior, utilize DATETIME2. • Para versões anteriores, se a escala for menor ou igual a 3, utilize DATETIME. Em todos os demais casos, utilize VARCHAR (37). Se a coluna não for uma data ou hora, utilize VARCHAR (tamanho).

Tipo de dados do AWS DMS	Tipo de dados do Babelfish
UINT1	TINYINT
UINT2	SMALLINT
UINT4	INT
UINT8	BIGINT
WSTRING	NVARCHAR(tamanho)
BLOB	VARBINARY(máximo) Para utilizar esse tipo de dados com o DMS, ative a utilização de BLOBs em uma tarefa específica. O DMS é compatível com os tipos de dados BLOB somente em tabelas que possuem uma chave primária.
CLOB	VARCHAR(máximo) Para utilizar esse tipo de dados com o DMS, ative a utilização de CLOBs em uma tarefa específica.
NCLOB	NVARCHAR(máximo) Para utilizar esse tipo de dados com o DMS, ative a utilização de NCLOBs em uma tarefa específica. Durante a CDC, o DMS é compatível com os tipos de dados NCLOB somente em tabelas que incluem uma chave primária.

Utilizar o Amazon Timestream como destino para o AWS Database Migration Service

Você pode usar o AWS Database Migration Service para migrar dados do banco de dados de origem para um endpoint de destino do Amazon Timestream, que aceita migrações de dados de carregamento completo e de CDC.

O Amazon Timestream é um serviço de banco de dados de séries temporais rápido, escalável e sem servidor criado para ingestão de dados de alto volume. Dados de séries temporais são uma sequência de pontos de dados coletados em um intervalo de tempo; eles são usados para medir eventos que mudam com o tempo. Ele é usado para coletar, armazenar e analisar métricas de aplicativos, DevOps aplicativos e aplicativos de análise de IoT. Assim que seus dados estiverem no Timestream, você poderá visualizar e identificar tendências e padrões nesses dados quase em tempo real. Para obter mais informações sobre o Amazon Timestream, consulte What is Amazon Timestream? no Guia do desenvolvedor do Amazon Timestream.

Tópicos

Pré-requisitos para usar o Amazon Timestream como destino para o AWS Database Migration Service

Antes de configurar o Amazon Timestream como destino para o AWS DMS, você deve criar um perfil do IAM. Esse perfil deve permitir que o AWS DMS tenha acesso aos dados que estão sendo migrados para o Amazon Timestream. O conjunto mínimo de permissões de acesso para o perfil que você usa para migrar para o Timestream é mostrado na política do IAM a seguir.

```
{
    "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"
        }
    ]
}
```

Se você pretende migrar todas as tabelas, use * para *TABLE_NAME* no exemplo acima.

Observe o seguinte sobre o uso do Timestream como destino:

- Se você pretende ingerir dados históricos com carimbos de data e hora superiores a um ano, recomendamos usar o AWS DMS para gravar os dados no Amazon S3 em um formato de valor separado por vírgula (csv). Em seguida, use o carregamento em lote do Timestream para ingerir os dados no Timestream. Para obter mais informações, consulte Using batch load in Timestream no Guia do desenvolvedor do Amazon Timestream.

- Para migrações de carga máxima de dados com menos de um ano, recomendamos definir o período de retenção do armazenamento de memória da tabela do Timestream com um valor superior ou igual ao carimbo de data e hora mais antigo. Depois que a migração for concluída, mude a retenção do armazenamento de memória da tabela para o valor desejado. Por exemplo, para migrar dados quando o carimbo de data e hora mais antigo é de dois meses, faça o seguinte:
  - Defina a retenção do armazenamento de memória da tabela de destino do Timestream para dois meses.
  - Inicie a migração de dados usando o AWS DMS.
  - Quando a migração de dados for concluída, altere o período de retenção da tabela do Timestream de destino para o valor desejado.

  Recomendamos estimar o custo do armazenamento de memória antes da migração, usando as informações nas seguintes páginas:
  - Amazon Timestream pricing
  - Calculadora de preços da AWS
- Para migrações de dados de CDC, recomendamos definir o período de retenção do armazenamento de memória da tabela de destino de forma que os dados ingeridos estejam dentro dos limites de retenção do armazenamento de memória. Para obter mais informações, consulte Best Practices: Writes no Guia do desenvolvedor do Amazon Timestream.

Configurações da tarefa de carga máxima com vários threads

Para ajudar a aumentar a velocidade da transferência de dados, o AWS DMS comporta a tarefa de migração de carga máxima com vários threads para um endpoint de destino do Timestream com estas configurações:

- `MaxFullLoadSubTasks`: utilize esta opção para indicar o número máximo de tabelas de origem a serem carregadas em paralelo. O DMS carrega cada tabela na tabela de destino correspondente do Amazon Timestream usando uma subtarefa dedicada. O padrão é 8; o valor máximo é 49.
- `ParallelLoadThreads`: utilize esta opção para especificar o número de threads que o AWS DMS utiliza para carregar cada tabela na respectiva tabela de destino do Amazon Timestream. O valor máximo de um destino do Timestream é 32. Você pode solicitar o aumento desse limite máximo.
- `ParallelLoadBufferSize`: utilize esta opção para especificar o número máximo de registros a serem armazenados no buffer utilizado pelos threads de carregamento paralelo para carregar

dados no destino do Amazon Timestream. O valor padrão é 50. Valor máximo de 1.000. Use essa configuração com `ParallelLoadThreads`; `ParallelLoadBufferSize` é válido somente quando há mais de um thread.

- `ParallelLoadQueuesPerThread`: utilize esta opção para especificar o número de filas que cada thread simultâneo acessa para extrair registros de dados das filas e gerar uma carga em lote para o destino. O padrão é um. No entanto, para destinos do Amazon Timestream de vários tamanhos de carga útil, o intervalo válido é de 5 a 512 filas por thread.

Configurações da tarefa de carga de CDC multithread

Para promover o desempenho da CDC, o AWS DMS oferece suporte a estas configurações de tarefa:

- `ParallelApplyThreads`: especifica o número de threads simultâneos que o AWS DMS utiliza durante uma carga de CDC para enviar registros de dados para um endpoint de destino do Timestream. O valor padrão é 0 e o valor máximo é 32.
- `ParallelApplyBufferSize`: especifica o número máximo de registros a serem armazenados em cada fila de buffer para os threads simultâneos enviarem a um endpoint de destino do Timestream durante uma carga de CDC. O valor padrão é 100 e o valor máximo é 1.000. Use essa opção quando `ParallelApplyThreads` especificar mais de um thread.
- `ParallelApplyQueuesPerThread`: especifica o número de filas que cada thread acessa para extrair registros de dados das filas e gerar um carregamento em lote para um endpoint do Timestream durante a CDC. O valor padrão é 1 e o valor máximo é 512.

Configurações de endpoint ao usar o Timestream como um destino para o AWS DMS

É possível utilizar as configurações de endpoint para configurar o banco de dados de destino do Timestream de forma semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--timestream-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoint que é possível utilizar com o Timestream como destino.

Nome	Descrição
`MemoryDuration`	Defina esse atributo para especificar o limite de retenção para armazenamento dos dados migrados na memória do Timestream. O tempo é medido em unidades de horas. O armazenamento de memória do Timestream é otimizado para um alto throughput de ingestão e acesso rápido. Valor padrão: 24 (horas) Valores válidos: de 1 a 8.736 (de 1 hora a 12 meses medidos em horas) Exemplo: `--timestream-settings '{"MemoryDuration": 20}'`
`DatabaseName`	Defina esse atributo para especificar o nome do banco de dados do Timestream de destino. Tipo: string Exemplo: `--timestream-settings '{"DatabaseName": " db_name"}'`
`TableName`	Defina esse atributo para especificar o nome da tabela do Timestream de destino. Tipo: string Exemplo: `--timestream-settings '{"TableName": " table_name "}'`
`MagneticDuration`	Defina esse atributo para especificar a duração magnética aplicada às tabelas do Timestream em dias. Esse é o limite de retenção para os dados ingeridos. O Timestream exclui qualquer carimbo de data e hora que exceda o limite de retenção. Para obter mais informaçõ

Nome	Descrição
	es, consulte Storage no Guia do desenvolvedor do Amazon Timestream. Exemplo: `--timestream-settings '{"MagneticDuration": "3"}'`
`CdcInsertsAndUpdates`	Defina esse atributo como `true` para especificar que o AWS DMS aplique somente inserções e atualizações, e não exclusões. O Timestream não permite a exclusão de registros; portanto, se esse valor for `false`, o AWS DMS anulará o registro correspondente no banco de dados do Timestream em vez de excluí-lo. Para obter mais informações, consulte Limitações a seguir. Valor padrão: `false` Exemplo: `--timestream-settings '{"CdcInsertsAndUpdates": "true"}'`
`EnableMagneticStoreWrites`	Defina esse atributo como `true` para habilitar gravações em armazenamento magnético. Quando esse valor é `false`, o AWS DMS não grava registros com um carimbo de data e hora anterior ao período de retenção do armazenamento de memória da tabela de destino, porque o Timestream não permite gravações no armazenamento magnético por padrão. Para obter mais informações, consulte Best Practices: Writes no Guia do desenvolvedor do Amazon Timestream. Valor padrão: `false` Exemplo: `--timestream-settings '{"EnableMagneticStoreWrites": "true"}'`

Criar e modificar um endpoint de destino do Amazon Timestream

Depois de criar um perfil do IAM e estabelecer o conjunto mínimo de permissões de acesso, você poderá criar um endpoint de destino do Amazon Timestream usando o console do AWS DMS ou o comando `create-endpoint` naAWS CLI com a sintaxe `--timestream-settings '{"EndpointSetting": "value", ...}'` do JSON.

Os exemplos a seguir mostram como criar e modificar um endpoint de destino do Timestream usando a AWS CLI.

Criar comando do endpoint de destino do Timestream

```
aws dms create-endpoint –endpoint-identifier timestream-target-demo
--endpoint-type target –engine-name timestream
--service-access-role-arn arn:aws:iam::123456789012:role/my-role
--timestream-settings
{
    "MemoryDuration": 20,
    "DatabaseName":"db_name",
    "MagneticDuration": 3,
    "CdcInsertsAndUpdates": true,
    "EnableMagneticStoreWrites": true,
}
```

Modificar comando do endpoint de destino do Timestream

```
aws dms modify-endpoint –endpoint-identifier timestream-target-demo
--endpoint-type target –engine-name timestream
--service-access-role-arn arn:aws:iam::123456789012:role/my-role
--timestream-settings
{
    "MemoryDuration": 20,
    "MagneticDuration": 3,
}
```

Utilizar o mapeamento de objetos para migrar dados para um tópico do Timestream

O AWS DMS utiliza regras de mapeamento de tabelas para correlacionar dados da origem com o tópico de destino do Timestream. Para mapear dados para um tópico de destino, utilize um tipo de regra de mapeamento de tabelas chamado mapeamento de objetos. Utilize o mapeamento de

objetos para definir como os registros de dados na origem são correlacionados com os registros de dados publicados em um tópico do Timestream.

Os tópicos do Timestream não têm uma estrutura predefinida, exceto uma chave de partição.

> **Note**
>
> Não é necessário utilizar o mapeamento de objetos. É possível utilizar o mapeamento de tabela normal para várias transformações. No entanto, o tipo de chave de partição seguirá estes comportamentos padrão:
>
> - A chave primária é utilizada como uma chave de partição para a carga máxima.
> - Se nenhuma configuração de tarefas de aplicação paralela for utilizada, `schema.table` será usada como uma chave de partição para a CDC.
> - Se as configurações de tarefas de aplicação paralela forem utilizadas, a chave primária será utilizada como uma chave de partição para a CDC.

Para criar uma regra de mapeamento de objetos, especifique `rule-type` como `object-mapping`. Essa regra especifica o tipo de mapeamento de objeto que você deseja usar. A estrutura da regra é a seguinte:

```
{
    "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
            }
        }
    ]
}
```

Atualmente, o AWS DMS oferece suporte a `map-record-to-record` e `map-record-to-document` como os únicos valores válidos para o parâmetro `rule-action`. Os valores `map-record-to-record` e `map-record-to-document` especificam o que o AWS DMS faz por padrão com registros que não são excluídos como parte da lista de atributos `exclude-columns`. Esses valores não afetam os mapeamentos de atributos de forma alguma.

Utilize `map-record-to-record` ao migrar de um banco de dados relacional para um tópico do Timestream. Esse tipo de regra utiliza o valor `taskResourceId.schemaName.tableName` encontrado no banco de dados relacional como a chave de partição no tópico do Timestream e cria um atributo para cada coluna no banco de dados de origem. Ao usar `map-record-to-record`, para qualquer coluna na tabela de origem que não aparece na lista de atributos `exclude-columns`, o AWS DMS cria um atributo correspondente no tópico de destino. Este atributo correspondente é criado, independentemente de a coluna de origem ser usada ou não em um mapeamento de atributos.

Uma maneira de compreender o `map-record-to-record` é vê-lo em ação. Para este exemplo, suponha que você está começando com uma linha de tabela do banco de dados relacional com a seguinte estrutura de dados:

FirstName	LastName	StoreId	HomeAddress	HomePhone	WorkAddress	WorkPhone	DateofBirth
Randy	Marsh	5	221B Baker Street	1234567890	31 Spooner Street, Quahog	9876543210	29/02/1988

Para migrar essas informações de um esquema chamado `Test` para um tópico do Timestream, crie regras para correlacionar os dados com o tópico de destino. A regra a seguir ilustra o mapeamento.

```
{
    "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"
            }
        }
    ]
}
```

Com um tópico do Timestream e uma chave de partição determinados (neste caso, `taskResourceId.schemaName.tableName`), o seguinte ilustra o formato do registro resultante utilizando os nossos exemplos de dados no tópico de destino do Timestream:

```
{
   "FirstName": "Randy",
   "LastName": "Marsh",
   "StoreId":  "5",
   "HomeAddress": "221B Baker Street",
   "HomePhone": "1234567890",
   "WorkAddress": "31 Spooner Street, Quahog",
   "WorkPhone": "9876543210",
   "DateOfBirth": "02/29/1988"
}
```

Limitações do uso do Amazon Timestream como destino para o AWS Database Migration Service

As seguintes limitações se aplicam ao utilizar o Amazon Timestream como destino:

- Dimensões e carimbos de data e hora: o Timestream usa as dimensões e os carimbos de data e hora nos dados de origem como uma chave primária composta e também não permite que você altere esses valores. Isso significa que, se você alterar o carimbo de data e hora ou as dimensões de um registro no banco de dados de origem, o banco de dados do Timestream tentará criar um registro. Portanto, é possível que, se você alterar a dimensão ou o carimbo de data e hora de um registro de forma que correspondam aos de outro registro existente, o AWS DMS atualizará os valores do outro registro em vez de criar um registro ou atualizar o registro correspondente anterior.
- Comandos DDL: a versão atual do AWS DMS só aceita os comandos DDL `CREATE TABLE` e `DROP TABLE`.
- Limitações de registro: o Timestream tem limitações para registros, como tamanho do registro e tamanho da medida. Para obter mais informações, consulte Cotas no Guia do desenvolvedor do Amazon Timestream.
- Exclusão de registros e valores nulos: o Timestream não comporta exclusão de registros. Para comportar a migração de registros excluídos da origem, o AWS DMS limpa os campos correspondentes nos registros no banco de dados de destino do Timestream. O AWS DMS altera

os valores nos campos do registro de destino correspondente para 0 nos campos numéricos, nulo nos campos de texto e falso nos campos boolianos.

- O Timestream como destino não aceita origens que não sejam bancos de dados relacionais (RDBMS).
- O AWS DMS só aceita o Timestream como destino nas seguintes regiões:
  - Leste dos EUA (Norte da Virgínia)
  - Leste dos EUA (Ohio)
  - Oeste dos EUA (Oregon)
  - Europa (Irlanda)
  - Europa (Frankfurt)
  - Ásia-Pacífico (Sydney)
  - Ásia-Pacífico (Tóquio)
- O Timestream como destino não aceita que a configuração de `TargetTablePrepMode` como `TRUNCATE_BEFORE_LOAD`. É recomendável não usar essa configuração.

Usar o Amazon RDS para Db2 e o IBM Db2 LUW como destino para o AWS DMS

Você pode migrar dados para um banco de dados do Amazon RDS para Db2 ou um Db2 on-premises com base em um banco de dados Db2 LUW usando o AWS Database Migration Service (AWS DMS).

Para obter informações sobre as versões do Db2 LUW compatíveis com o AWS DMS como destino, consulte Metas para AWS DMS.

É possível utilizar SSL para criptografar conexões entre o endpoint do Db2 LUW e a instância de replicação. Para obter mais informações sobre a utilização de SSL com um endpoint do Db2 LUW, consulte Usando SSL com AWS Database Migration Service.

Limitações de uso do Db2 LUW como destino para o AWS DMS

As seguintes limitações se aplicam ao utilizar um banco de dados Db2 LUW como destino para o AWS DMS: Para conhecer as limitações de uso do Db2 LUW como origem, consulte Limitações ao usar o Db2 LUW como fonte para AWS DMS.

- O AWS DMS só é compatível com o Db2 LUW como destino quando a origem é Db2 LUW ou Db2 para z/OS.
- Quando usado como destino, o Db2 LUW não comporta replicações com o modo LOB completo.
- Ele também não comporta o tipo de dados XML na fase de carregamento completo. Essa é uma limitação do utilitário dbload da IBM. Para obter mais informações, consulte IBM na documentação IBM Informix Servers.
- O AWS DMS trunca campos BLOB com valores correspondentes ao caractere de aspas duplas (“). Essa é uma limitação do utilitário dbload da IBM.

Configurações de endpoints ao utilizar o Db2 LUW como destino para o AWS DMS

É possível utilizar as configurações de endpoint para configurar o destino do Db2 LUW de maneira semelhante à utilização de atributos de conexão adicional. Você especifica as configurações ao criar o endpoint de destino utilizando o console do AWS DMS ou o comando `create-endpoint` na AWS CLI, com a sintaxe `--ibm-db2-settings '{"EndpointSetting": "value", ...}'` do JSON.

A tabela a seguir mostra as configurações de endpoints que é possível utilizar com o Db2 LUX como destino.

Nome	Descrição
`KeepCsvFiles`	Se verdadeiro, o AWS DMS salva todos os arquivos .csv no destino Db2 LUW que foram usados para replicar dados. O DMS usa esses arquivos para análise e solução de problemas.
`LoadTimeout`	Tempo transcorrido (em milissegundos) antes de o AWS DMS encerrar as operações realizadas pelo DMS no destino Db2. O valor padrão é 1.200 (20 minutos).
`MaxFileSize`	Especifica o tamanho máximo (em KB) dos arquivos .csv usados para transferir dados para o Db2 LUW.
`WriteBufferSize`	O tamanho (em KB) do buffer de gravação de arquivo na memória usado ao gerar arquivos .csv no disco local na

Nome	Descrição
	instância de replicação do DMS. O valor padrão é 1.024 (1 MB).

Configurar endpoints da VPC como endpoints de origem e de destino do AWS

O AWS DMS é compatível com endpoints de nuvem privada virtual (VPC) da Amazon como origens e destinos. AWS O DMS pode se conectar a qualquer banco de dados de origem ou de destino da AWS com endpoints da Amazon VPC, desde que rotas explicitamente definidas para esses bancos de dados de origem e de destino estejam definidas em sua VPC do AWS DMS.

Por ser compatível com endpoints da Amazon VPC, o AWS DMS facilita a manutenção da segurança de rede de ponta a ponta para todas as tarefas de replicação sem configuração adicional de rede. A utilização de endpoints da VPC para todos os endpoints de origem e de destino garante que todo o tráfego permaneça dentro da VPC e sob o seu controle. As atualizações para o AWS DMS versões 3.4.7 e superior exigem que você configure o AWS DMS para utilizar endpoints da VPC ou utilizar rotas públicas para todos os endpoints de origem e de destino que interagem com a Amazon Web Services a seguir:

- Amazon S3
- Amazon Kinesis
- AWS Secrets Manager
- Amazon DynamoDB
- Amazon Redshift
- Amazon OpenSearch Service

Talvez você precise de endpoints da VPC para compatibilidade com o AWS DMS a partir da versão 3.4.7, conforme descrito a seguir.

Quem é afetado ao migrar para o AWS DMS versões 3.4.7 e superior?

Você será afetado se estiver utilizando um ou mais dos endpoints do AWS DMS listados anteriormente, e esses endpoints não forem roteáveis publicamente ou não tiverem endpoints da VPC já associados a eles.

Quem não é afetado ao migrar para o AWS DMS versões 3.4.7 e superior?

Você não será afetado se:

- Você não estiver utilizando um ou mais dos endpoints do AWS DMS listados anteriormente.
- Você estiver utilizando qualquer um dos endpoints listados anteriormente e eles forem roteáveis publicamente.
- Você está utilizando qualquer um dos endpoints listados anteriormente e eles tiverem endpoints da VPC associados a eles.

Preparar uma migração para o AWS DMS versões 3.4.7 e superior

Para evitar falhas nas tarefas do AWS DMS ao utilizar qualquer um dos endpoints descritos anteriormente, utilize uma das etapas a seguir antes de atualizar o AWS DMS para a versão 3.4.7 ou superior:

- Torne os endpoints do AWS DMS afetados publicamente roteáveis. Por exemplo, adicione uma rota do gateway da Internet (IGW) a qualquer VPC já utilizada pela instância de replicação do AWS DMS para tornar todos os endpoints de origem e de destino roteáveis publicamente.
- Crie endpoints da VPC para acessar todos os endpoints de origem e de destino utilizados pelo AWS DMS, conforme descrito a seguir.

Para todos os endpoints da VPC existentes que você utiliza para os endpoints de origem e de destino do AWS DMS, verifique se eles utilizam uma política de confiança que esteja em conformidade com o documento da política XML `dms-vpc-role`. Para obter mais informações sobre o documento da política XML, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

Caso contrário, configure as instâncias de replicação como endpoints da VPC adicionando um endpoint da VPC à VPC que os contém. Se você configurou as instâncias de replicação sem endpoints públicos, a adição de um endpoint da VPC publicamente acessível à VPC que contém as instâncias de replicação as torna acessíveis publicamente. Não é necessário fazer mais nada para associar especificamente as instâncias de replicação ao endpoint da VPC.

> **Note**
>
> Serviços diferentes podem ter configurações exclusivas do endpoint da VPC. Por exemplo, ao utilizar o AWS Secrets Manager, normalmente não é necessário ajustar a tabela de roteamento. Sempre verifique os requisitos específicos de cada serviço.

Crie um endpoint da VPC na VPC que contém a instância de replicação

1. Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. Na barra de menu do console da VPC, escolha a mesma Região da AWS que a instância de replicação do AWS DMS.
3. No painel de navegação da VPC, escolha Endpoints.
4. Em Endpoints, escolha Criar endpoint.
5. Opcionalmente, é possível especificar um tag de nome. Por exemplo, **`my-endpoint-DynamoDB-01`**.
6. Em Serviços para o S3 ou o DynamoDB somente, escolha um Nome de serviço com o Tipo definido como Gateway.
7. Em VPC, escolha a mesma VPC da instância de replicação do AWS DMS para criar o endpoint.
8. Em Tabelas de rotas, escolha todos os valores de ID da tabela de rotas disponíveis.
9. Para especificar o controle de acesso, em Política, escolha Acesso total. Se quiser utilizar uma ferramenta de criação de políticas para especificar seu próprio controle de acesso, escolha Personalizado. Em qualquer caso, utilize uma política de confiança que esteja em conformidade com o documento de política JSON, `dms-vpc-role`. Para obter mais informações sobre esse documento de política, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.
10. Em Endpoints, verifique se o status do endpoint da VPC recém-criada é Disponível.

Para obter mais informações sobre como configurar endpoints da VPC para uma instância de replicação do AWS DMS, consulte Configurações de rede para migração de banco de dados. Para obter mais informações sobre a criação de endpoints da VPC da interface para acessar serviços da AWS em geral, consulte Acessar um serviço da AWS utilizando um endpoint da VPC de interface no Guia do PrivateLink do AWS. Para obter informações sobre a disponibilidade regional do AWS DMS para endpoints da VPC, consulte Tabela de regiões da AWS.

Instruções DDL compatíveis com o AWS DMS

Você pode executar declarações de Data Definition Language (DDL - Linguagem de definição de dados) no banco de dados de origem durante o processo de migração de dados. As instruções serão replicadas no banco de dados de destino pelo servidor de replicação.

As instruções DDL compatíveis incluem as seguintes:

- Create table
- Drop table
- Rename table
- Truncate table
- Add column
- Drop column
- Rename column
- Change column data type

O DMS não captura todas as instruções DDL compatíveis com alguns tipos de mecanismos de origem. E o DMS processa as instruções DDL de forma diferente ao aplicá-las a mecanismos de destino específicos. Para obter informações sobre quais instruções DDL são compatíveis com uma origem específica e como elas são aplicadas a um destino, consulte o tópico da documentação específica desse endpoint de origem e de destino

É possível utilizar as configurações de tarefas para definir a forma como o DMS processa o comportamento do DDL durante a captura de dados de alteração (CDC). Para ter mais informações, consulte Configurações de tarefa para processamento de DDL de processamento de alterações.

Trabalhar com tarefas do AWS DMS

Uma tarefa do AWS Database Migration Service (AWS DMS) é onde todo o trabalho acontece. Especifique quais tabelas (ou exibições) e esquemas usar para a migração e qualquer processamento especial, como requisitos de registro em log, dados da tabela de controle e gerenciamento de erros.

Uma tarefa pode consistir em três fases principais:

- Migração de dados existentes (carga máxima)
- A aplicação de alterações armazenadas em cache
- Replicação contínua (captura de dados de alteração)

Para obter mais informações e uma visão geral de como as tarefas de migração do AWS DMS migram os dados, consulte Visão de alto nível de AWS DMS

Ao criar uma tarefa de migração, é preciso saber algumas coisas:

- Antes de criar uma tarefa, crie um endpoint de origem, um endpoint de destino e uma instância de replicação.
- Você pode especificar muitas configurações de tarefa para ajustar a tarefa de migração. É possível configurá-las utilizando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API do AWS. Essas configurações incluem a especificação de como os erros de migração são gerenciados, como fazer registro em log de erros e como controlar informações da tabela. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.
- Após criar uma tarefa, você poderá executá-la imediatamente. As tabelas de destino com as definições de metadados necessárias são criadas e carregadas automaticamente, e você pode especificar a replicação contínua.
- Por padrão, o AWS DMS inicia a tarefa assim que ela é criada. Contudo, em algumas situações, você pode querer adiar o início da tarefa. Por exemplo, ao usar a AWS CLI, pode haver um processo que cria uma tarefa e um processo diferente que inicia a tarefa com base em algum evento de trigger. Conforme necessário, é possível adiar o início da tarefa.
- É possível monitorar, interromper ou reiniciar tarefas utilizando o console, a AWS CLI ou a API do AWS DMS. Para obter informações sobre como interromper uma tarefa utilizando a API do AWS DMS, consulte StopReplicationTask na Referência da API do AWS DMS.

A seguir, veja as ações que podem ser feitas durante o trabalho com uma tarefa do AWS DMS.

Tarefa	Documentação relevante
Criar uma tarefa Quando você cria uma tarefa, especifica a origem, o destino e a instância de replicação, juntament e com todas as configurações de migração.	Criar uma tarefa
Criar uma tarefa de replicação contínua Você pode configurar uma tarefa para fornecer replicação contínua entre a origem e o destino.	Criar tarefas para replicação contínua utilizando o AWS DMS
Aplicar as configurações de tarefas Cada tarefa tem configurações que podem ser definidas de acordo com as necessidades da migração do banco de dados. Crie essas configura ções em um arquivo JSON ou, no caso de algumas configurações, especifique-as usando o console do AWS DMS. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.	Especificando configurações de tarefas para tarefas do AWS Database Migration Service
Utilizar mapeamento de tabela O mapeamento de tabelas especific a configurações adicionais de tarefas para tabelas que utilizam vários tipos	Regras de seleção Regras de seleção e ações Regras de transformação

Tarefa	Documentação relevante
de regras. Essas regras permitem especificar a fonte de dados, o esquema de origem, as tabelas e visualizações, os dados, quaisquer transformações de tabela e dados que ocorrerão durante a tarefa e as configurações de como essas tabelas e colunas são migradas da origem para o destino.	Regras de transformação e ações Regras de configuração de tabelas Regras e operações de configurações de tabelas e coleções
Executar avaliações de pré-migração de tarefas É possível ativar e executar avaliações de pré-migração de tarefas que mostram problemas com um banco de dados de origem e de destino compatível que podem causar problemas durante a migração. Isso pode incluir problemas como tipos de dados incompatíveis, índices e chaves primárias incompatíveis e outras configurações de tarefas conflitantes. Essas avaliações de pré-migração são executadas antes de você executar a tarefa para identificar possíveis problemas antes que eles ocorram durante a migração.	Ativar e trabalhar com avaliações de pré-migração de uma tarefa

Tarefa	Documentação relevante
Validação de dados A validação de dados é uma configuração de tarefa que você pode usar para que o AWS DMS compare os dados no armazenamento de dados de destino com os dados do armazenamento de dados de origem.	Validação de dados do AWS DMS.
Modificar uma tarefa Quando uma tarefa é interrompida, é possível modificar as configurações dela.	Modificar uma tarefa
Mover uma tarefa Quando uma tarefa é interrompida, é possível transferir a tarefa para outra instância de replicação.	Mover uma tarefa
Recarregar tabelas durante uma tarefa Você poderá recarregar uma tabela durante uma tarefa se ocorrer um erro durante sua execução.	Recarregar tabelas durante uma tarefa

Tarefa	Documentação relevante
Aplicar filtros É possível usar filtros de origem para limitar o número e o tipo de registros transferidos da origem ao destino. Por exemplo, você pode especific ar que somente os funcionários localizados na sede serão movidos para o banco de dados de destino. Os filtros são aplicados a uma coluna de dados.	Usar filtros de origem
Monitorar uma tarefa Há várias maneiras de obter informações sobre o desempenho de uma tarefa e das tabelas usadas por ela.	Monitoramento de tarefas do AWS DMS
Gerenciar logs de tarefas Você pode visualizar e excluir logs de tarefas usando a API do AWS DMS ou a AWS CLI.	Visualização e gerenciamento dos logs de tarefas do AWS

Tópicos

- Criar uma tarefa
- Criar tarefas para replicação contínua utilizando o AWS DMS
- Modificar uma tarefa
- Mover uma tarefa
- Recarregar tabelas durante uma tarefa
- Utilizar o mapeamento de tabela para especificar as configurações da tarefa
- Usar filtros de origem
- Ativar e trabalhar com avaliações de pré-migração de uma tarefa

- Especificar dados complementares para configurações de tarefa

Criar uma tarefa

Para criar uma tarefa de AWS DMS migração, faça o seguinte:

- Antes de criar uma tarefa de migração, crie um endpoint de origem, um endpoint de destino e uma instância de replicação.
- Selecione um método de migração:
  - Migração de dados para o banco de dados de destino: este processo cria arquivos ou tabelas no banco de dados de destino e define automaticamente os metadados necessários no destino. Ele também preenche as tabelas com dados de origem. Os dados das tabelas são carregados em paralelo para melhorar a eficiência. Esse processo é a opção Migrar dados existentes no AWS Management Console e é chamado `Full Load` na API.
  - Captura de alterações durante a migração: este processo captura alterações no banco de dados de origem que ocorrem enquanto os dados estão sendo migrados da origem para o destino. Quando a migração dos dados solicitados originalmente for concluída, o processo de captura de dados de alteração (CDC) aplicará as alterações capturadas ao banco de dados de destino. As alterações são capturadas e aplicadas como unidades de transações confirmadas únicas, e é possível atualizar várias tabelas de destino diferentes como uma única confirmação de origem. Essa abordagem garante integridade transacional no banco de dados de destino. Esse processo é a opção Migração de dados existentes e replicação de alterações em andamento no console e se chama `full-load-and-cdc` na API.
  - Replicação apenas de alterações de dados no banco de dados de origem: este processo lê o arquivo de log de recuperação do sistema de gerenciamento do banco de dados de origem (DBMS) e agrupa as entradas de cada transação. Em alguns casos, não é AWS DMS possível aplicar alterações na meta dentro de um tempo razoável (por exemplo, se a meta não estiver acessível). Nesses casos, AWS DMS armazena em buffer as alterações no servidor de replicação pelo tempo que for necessário. Ele não relê os logs do DBMS de origem, que pode demorar muito tempo. Esse processo é a opção Replicate data changes only (Replicar somente alterações de dados) no console do AWS DMS .
- Determine como a tarefa deve lidar com grandes objetos binários (LOBs) na origem. Para ter mais informações, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS.

- Especifique as configurações da tarefa de migração. Elas incluem configurar o registro em log, especificar quais dados são gravados na tabela de controle de migração, como os erros são processados e outras configurações. Para obter mais informações sobre as configurações de tarefas, consulte Especificando configurações de tarefas para tarefas do AWS Database Migration Service.
- Configure o mapeamento da tabela para definir regras para selecionar e filtrar os dados que você está migrando. Para obter mais informações sobre o mapeamento de tabela, consulte Utilizar o mapeamento de tabela para especificar as configurações da tarefa. Antes de especificar o mapeamento, analise a seção da documentação sobre o mapeamento de tipo de dados do banco de dados de origem e de destino.
- Ative e execute avaliações de tarefas de pré-migração antes de executar a tarefa. Para obter informações sobre avaliações de pré-migração, consulte Ativar e trabalhar com avaliações de pré-migração de uma tarefa.
- Especifique todos os dados suplementares necessários para a tarefa migrar seus dados. Para ter mais informações, consulte Especificar dados complementares para configurações de tarefa.

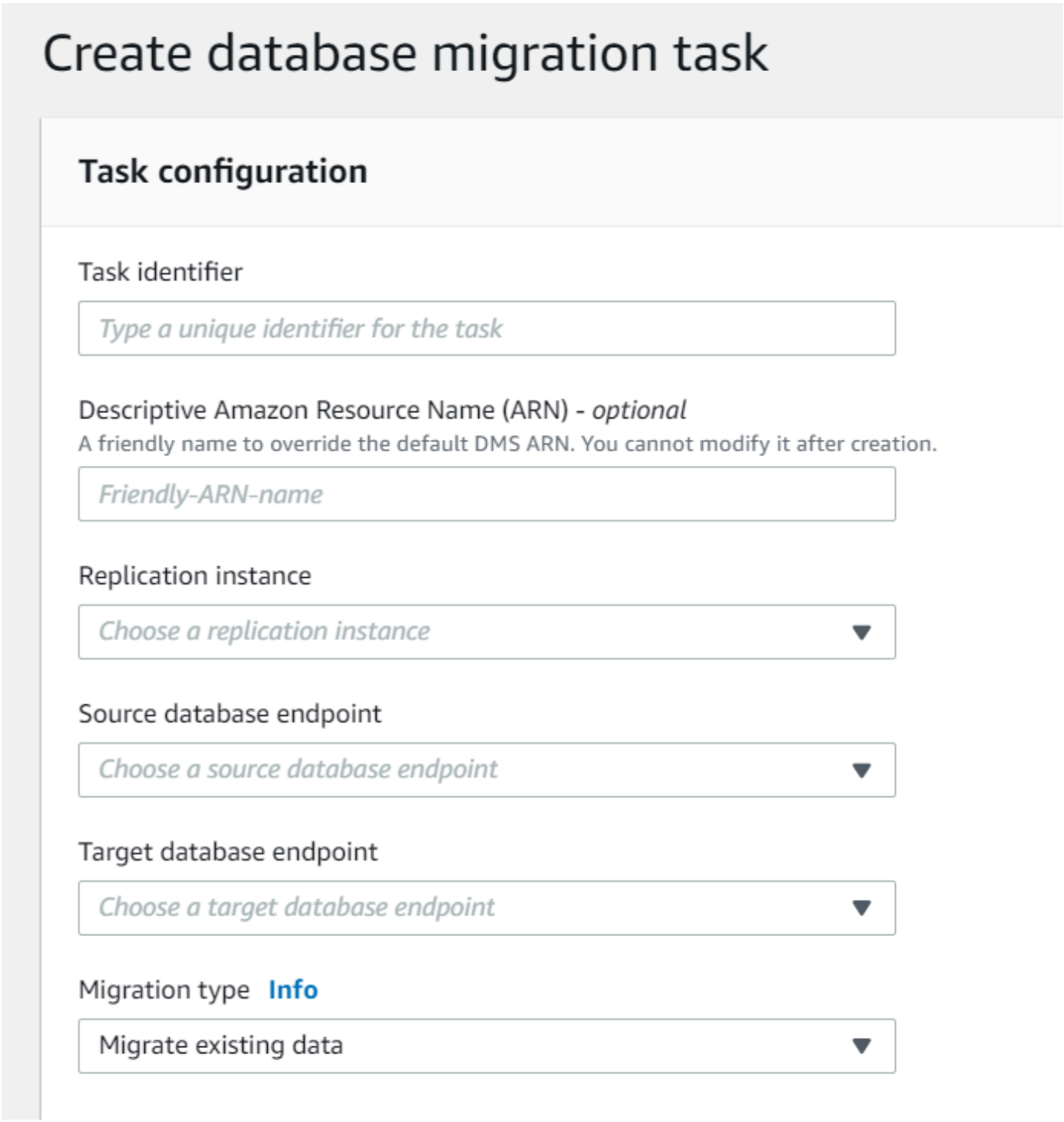
É possível optar por iniciar uma tarefa assim que você concluir a especificação de informações para essa tarefa na página Criar tarefa. Como alternativa, também é possível iniciar a tarefa na página Painel posteriormente.

O procedimento a seguir pressupõe que você já tenha especificado endpoints e informações da instância de replicação. Para obter mais informações sobre a configuração de endpoints, consulte Criar endpoints de origem e de destino.

Como criar uma tarefa de migração

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Se você estiver conectado como usuário AWS Identity and Access Management (IAM), verifique se você tem as permissões apropriadas para acessar AWS DMS. Para obter mais informações sobre as permissões necessárias, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, escolha Tarefas de migração de banco de dados e Criar tarefa.
3. Na página Criar tarefa de migração de banco de dados, na seção Configuração da tarefa, especifique as opções da tarefa. A tabela a seguir descreve as configurações.



Para esta opção	Faça o seguinte
Identificador de tarefa	Digite um nome para a tarefa.

Para esta opção	Faça o seguinte
Nome do recurso da Amazon (ARN) descritivo: opcional	Um nome amigável para substituir o AWS DMS ARN padrão. Não é possível alterar esse nome depois de criar a tarefa.
Replication instance	Exibe a instância de replicação que será utilizada.
Endpoint do banco de dados de origem	Exibe o endpoint de origem que será utilizado.
Endpoint do banco de dados de destino	Exibe o endpoint de destino que será utilizado.
Migration type	Escolha o método de migração que deseja usar. É possível escolher que apenas os dados existentes sejam migrados para o banco de dados de destino ou que as alterações em andamento sejam enviadas para o banco de dados de destino além dos dados migrados.

4. Na seção Configurações da tarefa, especifique os valores para a edição da tarefa, o modo de preparação da tabela de destino, a tarefa de interrupção, as configurações de LOB, a validação e o registro em log.

Para esta opção	Faça o seguinte
Modo de edição	Escolha se deseja utilizar o Assistente ou o editor JSON para especificar as configurações da tarefa. Se você escolher Assistente, as seguintes opções serão exibidas.
Modo de início da CDC para transações de origem	Essa configuração só estará visível se você escolher Replicar alterações de dados somente para o Tipo de migração na seção anterior. Desativar o modo de início personalizado da CDC: se você escolher esta opção, poderá iniciar a tarefa

Para esta opção	Faça o seguinte
	automaticamente utilizando a opção Automaticamente ao criar a seguir ou manualmente utilizando o console. Ativar o modo de início personalizado da CDC: se você escolher esta opção, poderá especificar um horário de início em UTC personalizado para iniciar o processamento das alterações.

Para esta opção	Faça o seguinte
Target table preparation mode	Essa configuração só estará visível se você escolher Migrar dados existentes ou Migrar dados existentes e replicar as alterações em andamento para o Tipo de migração na seção anterior. Não fazer nada — No modo Não fazer nada, AWS DMS pressupõe que as tabelas de destino tenham sido pré-criadas no destino. Se as tabelas não estiverem vazias, poderão ocorrer conflitos durante a migração dos dados e pode resultar em um erro na tarefa do DMS. Se a tabela de destino não existir, o DMS a criará. A estrutura da tabela permanece como está e quaisquer dados existentes são deixados na tabela. O modo Não executar nenhuma ação será uma escolha adequada para tarefas somente CDC quando as tabelas de destino forem pré-aterr adas da origem e a replicação contínua for aplicada para manter a origem e o destino em sincronia. Para pré-criar tabelas, é possível utilizar o AWS Schema Conversion Tool (AWS SCT). Para obter mais informações, consulte Instalando AWS SCT. Descartar tabelas no destino: no modo Descartar tabelas no destino, o AWS DMS descarta as tabelas de destino e as recria antes de iniciar a migração. Essa abordagem garante que as tabelas de destino estejam vazias quando a migração começar. AWS DMS cria somente os objetos necessários para migrar os dados com eficiência: tabelas, chaves primárias e, em alguns casos, índices exclusivos. AWS DMS não cria índices secundários, restrições de chave não primária ou padrões de dados de coluna. Se você estiver executando uma tarefa de carga máxima mais CDC ou somente de CDC, é recomendável pausar a migração neste momento. Depois, crie índices

Para esta opção	Faça o seguinte
	secundários que sejam compatíveis com a filtragem para instruções de atualização e exclusão. Talvez seja necessário executar algumas configurações no banco de dados de destino ao utilizar o modo Descartar tabelas no destino. Por exemplo, para um alvo Oracle, não é AWS DMS possível criar um esquema (usuário do banco de dados) por motivos de segurança. Nesse caso, você pré-cria o usuário do esquema para AWS DMS poder criar as tabelas quando a migração começar. Para a maioria dos outros tipos de destino, AWS DMS cria o esquema e todas as tabelas associadas com os parâmetros de configuração adequados. Truncar — No modo Truncar, AWS DMS trunca todas as tabelas de destino antes do início da migração. Se a tabela de destino não existir, o DMS a criará. A estrutura da tabela permanece como está, mas as tabelas são truncadas no destino. O modo Truncar é adequado para migrações de carga máxima ou carga máxima mais CDC em que o esquema de destino foi pré-criado antes do início da migração. Para recriar tabelas, é possível utilizar o AWS SCT. Para obter mais informações, consulte Instalando AWS SCT. **Note** Se o destino for o MongoDB, o modo Truncar não truncará as tabelas no destino. Em vez disso, ele descartará a coleção e perderá todos os índices. Evite o modo Truncar quando o destino for o MongoDB.

Para esta opção	Faça o seguinte
Stop task after full load completes	Essa configuração só estará visível somente se você escolher Migrar dados existentes e replicar as alterações em andamento para o Tipo de migração na seção anterior. Não interromper: não interrompe a tarefa, mas aplica as alterações armazenadas em cache imediatamente e prossegue. Interromper antes de aplicar alterações armazenadas em cache: interrompe a tarefa antes da aplicação das alterações armazenadas em cache. Utilizando essa abordagem, é possível adicionar índices secundários que podem agilizar a aplicação das alterações. Interromper depois de aplicar alterações armazenad as em cache: interrompe a tarefa após as alteraçõe s armazenadas em cache terem sido aplicadas. Utilizando essa abordagem, é possível adicionar chaves estrangeiras, se estiver utilizando a aplicação transacional.
Include LOB columns in replication	Não incluir colunas LOB: as colunas LOB são excluídas da migração. Modo LOB completo — Migre LOBs completos , independentemente do tamanho. AWS DMS migra LOBs por partes em partes controladas pelo parâmetro LOB Chunk size. Esse modo é mais lento que o modo LOB limitado. Modo LOB limitado: trunca LOBs de acordo com o valor do parâmetro Tamanho máximo de LOB. Esse modo é mais rápido que o modo LOB completo.

Para esta opção	Faça o seguinte
Tamanho máximo do LOB (kb)	Em Modo LOB limitado, as colunas de LOB que ultrapassam o Tamanho máximo do LOB são truncadas no valor Tamanho máximo do LOB especificado.
Ativar a validação	Ativa a validação de dados para verificar se eles são migrados corretamente da origem para o destino. Para ter mais informações, consulte Validação de dados do AWS DMS.
Ativar CloudWatch registros	Permite o registro pela Amazon CloudWatch.

5. Na seção Avaliação de pré-migração, selecione se uma avaliação de pré-migração deve ser executada. Uma avaliação de pré-migração avisa sobre possíveis problemas de migração antes de iniciar a tarefa de migração do banco de dados. Para ter mais informações, consulte Ativar e trabalhar com avaliações de pré-migração.
6. Na seção Configuração de startup da tarefa de migração, especifique se a tarefa deve ser iniciada automaticamente após a criação.
7. Na seção Tags, especifique as tags necessárias para organizar a tarefa. É possível utilizar tags para gerenciar os perfis e as políticas do IAM e rastrear os custos do DMS. Para ter mais informações, consulte Marcar recursos.
8. Assim que tiver terminado as configurações de tarefa, selecione Create task (Criar tarefa).

Especificando configurações de tarefas para tarefas do AWS Database Migration Service

Cada tarefa tem configurações que podem ser definidas de acordo com as necessidades da migração do banco de dados. Você cria essas configurações em um arquivo JSON ou, com algumas configurações, pode especificar as configurações usando o AWS DMS console. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

Há vários tipos principais de configurações de tarefa, conforme listado a seguir.

Tópicos

- Exemplo de configurações de tarefas
- Configurações de tarefa de metadados de destino
- Configurações de tarefa de carregamento completo
- Configurações de tarefa do Time Travel
- Configurações de registro de tarefa
- Configurações de tarefa de tabela de controle
- Configurações de tarefas de buffer de stream
- Configurações de ajuste de processamento de alterações
- Configurações da tarefa de validação de dados
- Configurações de tarefa para processamento de DDL de processamento de alterações
- Configurações da tarefa de substituição de caracteres
- Configurações de tarefa de imagem anterior
- Configurações de tarefa de tratamento de erros
- Salvar configurações de tarefa

Configurações de tarefa	Documentação relevante
Criar um relatório de avaliação de tarefa É possível criar um relatório de avaliação de tarefa que mostre os tipos de dados incompatíveis, que podem causar problemas durante a migração. É possível executar esse relatório na tarefa antes de executá-la, para descobrir possíveis problemas.	Ativar e trabalhar com avaliações de pré-migração de uma tarefa
Criar uma tarefa Quando você cria uma tarefa, especifica a origem, o destino e a instância de replicação, juntament	Criar uma tarefa

Configurações de tarefa	Documentação relevante
e com todas as configurações de migração.	
Criar uma tarefa de replicação contínua Você pode configurar uma tarefa para fornecer replicação contínua entre a origem e o destino.	Criar tarefas para replicação contínua utilizando o AWS DMS
Aplicar as configurações de tarefas Cada tarefa tem configurações que podem ser definidas de acordo com as necessidades da migração do banco de dados. Você cria essas configurações em um arquivo JSON ou, com algumas configurações, pode especificar as configurações usando o AWS DMS console.	Especificando configurações de tarefas para tarefas do AWS Database Migration Service
Validação de dados Use a validação de dados para AWS DMS comparar os dados do seu armazenamento de dados de destino com os dados do seu armazenam ento de dados de origem.	Validação de dados do AWS DMS
Modificar uma tarefa Quando uma tarefa é interrompida, é possível modificar as configurações dela.	Modificar uma tarefa

Configurações de tarefa	Documentação relevante
Recarregar tabelas durante uma tarefa Você poderá recarregar uma tabela durante uma tarefa se ocorrer um erro durante sua execução.	Recarregar tabelas durante uma tarefa
Utilizar mapeamento de tabela O mapeamento de tabela utiliza vários tipos de regras para especific ar as configurações da fonte de dados, do esquema de origem, dos dados e de quaisquer transformações que devem ocorrer durante a tarefa.	Regras de seleção Regras de seleção e ações Regras de transformação Regras de transformação e ações
Aplicar filtros É possível usar filtros de origem para limitar o número e o tipo de registros transferidos da origem ao destino. Por exemplo, você pode especific ar que somente os funcionários localizados na sede serão movidos para o banco de dados de destino. Os filtros são aplicados a uma coluna de dados.	Usar filtros de origem
Monitorar uma tarefa Há várias maneiras de obter informações sobre o desempenho de uma tarefa e das tabelas usadas por ela.	Monitoramento de tarefas do AWS DMS

Configurações de tarefa	Documentação relevante
Gerenciar logs de tarefas Você pode visualizar e excluir registros de tarefas usando a AWS DMS API ou AWS CLI.	Visualização e gerenciamento dos logs de tarefas do AWS

Exemplo de configurações de tarefas

Você pode usar o AWS Management Console ou o AWS CLI para criar uma tarefa de replicação. Se você usar o AWS CLI, defina as configurações da tarefa criando um arquivo JSON e, em seguida, especificando o URI file://do arquivo JSON como _ReplicationTaskSettingsparâmetro da CreateReplication operação Tarefa.

O exemplo a seguir mostra como usar o AWS CLI para chamar a `CreateReplicationTask` operação:

```
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
```

O exemplo anterior utiliza um arquivo de mapeamento de tabela chamado`tablemappings.json`. Para ver exemplos de mapeamento de tabela, consulte Utilizar o mapeamento de tabela para especificar as configurações da tarefa.

Um arquivo JSON de configurações de tarefas pode ter a seguinte aparência:

```
{
  "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
  }
}
```

Configurações de tarefa de metadados de destino

As configurações de metadados de destino incluem: Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

- `TargetSchema`: o nome do esquema da tabela de destino. Se essa opção de metadados estiver vazia, o esquema da tabela de origem será utilizado. O AWS DMS adicionará automaticamente o prefixo do proprietário do banco de dados de destino a todas as tabelas se não houver esquema de origem definido. Essa opção deve ficar vazia para endpoints de destino do tipo MySQL. A renomeação de um esquema no mapeamento de dados tem precedência sobre essa configuração.
- Configurações de LOB: configurações que determinam como objetos grandes (LOBs) são gerenciados. Se você definir `SupportLobs=true`, defina um dos seguintes como `true`:
  - `FullLobMode`: se você definir esta opção como `true`, insira um valor para a opção `LobChunkSize`. Insira o tamanho, em kilobytes, dos blocos de LOB a serem utilizados durante a replicação dos dados para o destino. A opção `FullLobMode` funciona melhor para tamanhos de LOB muito grandes, mas costuma deixar o carregamento mais lento. O valor recomendado para `LobChunkSize` é 64 kilobytes. Aumentar o valor de `LobChunkSize` acima de 64 kilobytes pode causar falhas na tarefa.
  - `InlineLobMaxSize`— Esse valor determina quais LOBs são AWS DMS transferidos em linha durante uma carga completa. A transferência de LOBs pequenos é mais eficiente do que procurá-los em uma tabela de origem. Durante uma carga completa, AWS DMS verifica todos os LOBs e executa uma transferência em linha para os LOBs menores que. `InlineLobMaxSize` AWS DMS transfere todos os LOBs maiores do que o `InlineLobMaxSize` in`FullLobMode`. O valor padrão de `InlineLobMaxSize` é 0, e o intervalo é de 1 a 102400 kilobytes (100 MB). Defina um valor para `InlineLobMaxSize` somente se você souber que a maioria dos LOBs é menor que o valor especificado em `InlineLobMaxSize`.
  - `LimitedSizeLobMode`: se você definir esta opção como `true`, insira um valor para a opção `LobMaxSize`. Insira o tamanho máximo, em kilobytes, de um LOB individual. O valor máximo recomendado para `LobMaxSize` é 102400 kilobytes (100 MB).

Para obter mais informações sobre os critérios para utilização dessas configurações de LOB de tarefa, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS. Também é possível controlar o gerenciamento de LOBs para tabelas individuais. Para ter mais informações, consulte Regras e operações de configurações de tabelas e coleções.

- `LoadMaxFileSize`: uma opção para endpoints de destino baseados em CSV, como o MySQL, o PostgreSQL e o Amazon Redshift, que são compatíveis com a utilização de valores separados por vírgulas (.csv) para carregar dados. `LoadMaxFileSize`define o tamanho máximo no disco de dados armazenados e descarregados, como arquivos .csv. Essa opção substitui o atributo de conexão de endpoint de destino, `maxFileSize`. É possível fornecer valores de 0, que indica que essa opção não substitui o atributo de conexão, a 100.000 KB.
- `BatchApplyEnabled`: determina se cada transação é aplicada individualmente ou se as alterações são confirmadas em lotes. O valor padrão é `false`.

  Quando `BatchApplyEnabled` está definido como `true`, o DMS exige uma chave primária (PK) ou uma chave exclusiva (UK) na(s) tabela(s) de origem. Sem uma PK ou UK nas tabelas de origem, somente as inserções em lote são aplicadas, mas não atualizações e exclusões em lote.

  Quando `BatchApplyEnabled` está definido como `true`, o AWS DMS gerará uma mensagem de erro se uma tabela de destino tiver uma restrição exclusiva e uma chave primária. Tabelas de destino com uma restrição exclusiva e uma chave primária não são compatíveis quando `BatchApplyEnabled` está definida como `true`.

  Quando `BatchApplyEnabled` é definida como verdadeira e AWS DMS encontra um erro de dados em uma tabela com a política padrão de tratamento de erros, a AWS DMS tarefa muda do modo em lote para o modo do resto das tabelas. one-by-one Para alterar esse comportamento, é possível definir a ação `"SUSPEND_TABLE"` nas seguintes políticas na propriedade de grupo `"ErrorBehavior"` grupo do arquivo JSON de configurações da tarefa:

  - `DataErrorPolicy`
  - `ApplyErrorDeletePolicy`
  - `ApplyErrorInsertPolicy`
  - `ApplyErrorUpdatePolicy`

  Para obter mais informações sobre essa propriedade de grupo `"ErrorBehavior"`, consulte o exemplo de arquivo JSON de configurações de tarefas em Especificando configurações de tarefas para tarefas do AWS Database Migration Service. Depois de definir essas políticas

como"SUSPEND_TABLE", a AWS DMS tarefa suspende os erros de dados em todas as tabelas que os geram e continua no modo em lote para todas as tabelas.

É possível utilizar o parâmetro `BatchApplyEnabled` com o parâmetro `BatchApplyPreserveTransaction`. Se `BatchApplyEnabled` estiver definido como `true`, o parâmetro `BatchApplyPreserveTransaction` determinará a integridade transacional.

Se `BatchApplyPreserveTransaction` estiver definido como `true`, a integridade transacional será preservada e será garantido que um lote contenha todas as alterações dentro de uma transação da origem.

Se `BatchApplyPreserveTransaction` estiver definido como `false`, poderá haver lapsos temporários na integridade transacional para melhorar o desempenho.

O parâmetro `BatchApplyPreserveTransaction` se aplica somente aos endpoints de destino Oracle e só será relevante quando o parâmetro `BatchApplyEnabled` estiver definido como `true`.

Quando colunas de LOB estiverem incluídas na replicação, é possível utilizar `BatchApplyEnabled` somente no modo LOB limitado.

Para obter mais informações sobre como utilizar essas configurações para uma carga de captura de dados de alteração (CDC), consulte Configurações de ajuste de processamento de alterações.

- `MaxFullLoadSubTasks`: indica o número máximo de tabelas a serem carregadas em paralelo. O padrão é 8; o valor máximo é 49.
- `ParallelLoadThreads`— Especifica o número de threads AWS DMS usados para carregar cada tabela no banco de dados de destino. Esse parâmetro tem valores máximos para destinos não RDBMS. O valor máximo para um destino do DynamoDB é 200. O valor máximo para uma meta do Amazon Kinesis Data Streams, Apache Kafka ou OpenSearch Amazon Service é 32. É possível pedir um aumento desse limite máximo. `ParallelLoadThreads`aplica-se às tarefas de carga máxima. Para obter informações sobre as configurações para o carregamento paralelo de tabelas individuais, consulte Regras e operações de configurações de tabelas e coleções.

  Essa configuração se aplica aos seguintes tipos de mecanismo de endpoint:

  - DynamoDB
  - Amazon Kinesis Data Streams
  - Amazon MSK

- OpenSearch Serviço Amazon
- Amazon Redshift

AWS DMS suporta `ParallelLoadThreads` o MySQL como um atributo de conexão extra. `ParallelLoadThreads`não se aplica ao MySQL como uma configuração de tarefa.

- `ParallelLoadBufferSize` especifica o número máximo de registros a serem armazenados em buffer que os threads de carga paralela utilizam para carregar dados no destino. O valor padrão é 50. Valor máximo de 1.000. No momento, essa configuração só é válida quando DynamoDB, Kinesis, Apache Kafka ou é o destino. OpenSearch Utilize esse parâmetro com `ParallelLoadThreads`. `ParallelLoadBufferSize` é válido somente quando há mais de um thread. Para obter informações sobre as configurações para o carregamento paralelo de tabelas individuais, consulte Regras e operações de configurações de tabelas e coleções.
- `ParallelLoadQueuesPerThread`: especifica o número de filas que cada thread simultâneo acessa para extrair registros de dados das filas e gerar uma carga em lote para o destino. O padrão é um. Essa configuração é válida somente quando o Kinesis ou o Apache Kafka é o destino.
- `ParallelApplyThreads`— Especifica o número de threads simultâneos que são AWS DMS usados durante um carregamento do CDC para enviar registros de dados para um endpoint de destino do Amazon DocumentDB, Kinesis, Amazon MSK ou Amazon Redshift. OpenSearch O padrão é zero (0).

  Essa configuração só se aplica à CDC. Esta configuração não se aplica à carga máxima.

  Essa configuração se aplica aos seguintes tipos de mecanismo de endpoint:

  - Amazon DocumentDB (compatível com MongoDB)
  - Amazon Kinesis Data Streams
  - Amazon Managed Streaming for Apache Kafka
  - OpenSearch Serviço Amazon
  - Amazon Redshift
- `ParallelApplyBufferSize`— Especifica o número máximo de registros a serem armazenados em cada fila de buffer para que threads simultâneos sejam enviados para um endpoint de destino do Amazon DocumentDB, Kinesis, Amazon MSK ou OpenSearch Amazon Redshift durante um carregamento do CDC. O valor padrão é 100. O valor máximo é 1000. Use essa opção quando `ParallelApplyThreads` especificar mais de um thread.

- `ParallelApplyQueuesPerThread`— Especifica o número de filas que cada thread acessa para retirar registros de dados das filas e gerar uma carga em lote para um Amazon DocumentDB, Kinesis, Amazon MSK ou endpoint durante o CDC. OpenSearch O valor padrão é 1.

Configurações de tarefa de carregamento completo

As configurações de carga máxima incluem o seguinte: Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

- Para indicar como tratar a carga no destino em um startup de carga máxima, especifique um dos seguintes valores para a opção `TargetTablePrepMode`:
  - `DO_NOTHING`: os dados e metadados da tabela de destino existente não são afetados.
  - `DROP_AND_CREATE`: a tabela existente é descartada, e uma nova tabela é criada em seu lugar.
  - `TRUNCATE_BEFORE_LOAD`: os dados são truncados sem afetar os metadados da tabela.
- Para atrasar a criação de uma chave primária ou de um índice único até a conclusão da carga máxima, defina a opção `CreatePkAfterFullLoad` como `true`.
- Para tarefas com carga máxima e CEC ativado, defina as seguintes opções para `Stop task after full load completes`:
  - `StopTaskCachedChangesApplied`: defina esta opção como `true` para interromper uma tarefa após a conclusão de uma carga máxima ser concluída e as alterações em cache serem aplicadas.
  - `StopTaskCachedChangesNotApplied`: defina esta opção como `true` para interromper uma tarefa antes que as alterações em cache sejam aplicadas.
- Para indicar o número máximo de tabelas a serem carregadas em paralelo, defina a opção `MaxFullLoadSubTasks`. O padrão é 8; o valor máximo é 49.
- Defina a opção `ParallelLoadThreads` para indicar quantos threads simultâneos o DMS utiliza durante um processo de carga máxima para enviar registros de dados para um endpoint de destino. O valor padrão é zero (0).

> **Important**
>
> `MaxFullLoadSubTasks` controla o número de tabelas ou segmentos de tabela a serem carregados em paralelo. `ParallelLoadThreads` controla o número de threads utilizados por uma tarefa de migração para executar as cargas em paralelo.

Essas configurações são multiplicativas. Dessa forma, o número total de threads utilizados durante uma tarefa de carga máxima é aproximadamente o resultado do valor de `ParallelLoadThreads` multiplicado pelo valor de `MaxFullLoadSubTasks` (`ParallelLoadThreads*MaxFullLoadSubtasks`).
Se você criar tarefas com um grande número de subtarefas de carga máxima e um grande número de threads de carga paralela, a tarefa poderá consumir muita memória e falhar.

- Você pode definir o número de segundos que AWS DMS aguarda o fechamento das transações antes de iniciar uma operação de carga total. Para fazer isso, se as transações estiverem abertas quando a tarefa começar, defina a opção `TransactionConsistencyTimeout`. O valor padrão é 600 (10 minutos). AWS DMS inicia a carga total após o valor do tempo limite ser atingido, mesmo se houver transações abertas. Uma full-load-only tarefa não espera por 10 minutos, mas começa imediatamente.
- Para indicar o número máximo de eventos que podem ser transferidos em conjunto, defina a opção `CommitRate`. O valor padrão é 10000 e o valor máximo é 50000.

Configurações de tarefa do Time Travel

Para registrar e depurar tarefas de replicação, você pode usar o AWS DMS Time Travel. Nessa abordagem, você utiliza o Amazon S3 para armazenar logs e criptografá-los utilizando as chaves de criptografia. Somente com acesso ao bucket do S3 do Time Travel, é possível recuperar os logs do S3 utilizando filtros de data e hora e visualizar, baixar e ofuscar os logs conforme necessário. Ao fazer isso, é possível “viajar no tempo” com segurança para investigar as atividades do banco de dados. O Time Travel funciona de forma independente do CloudWatch registro. Para obter mais informações sobre CloudWatch registro em log, consulteConfigurações de registro de tarefa.

Você pode usar o Time Travel em todas as AWS regiões com AWS DMS endpoints de origem Oracle, Microsoft SQL Server e PostgreSQL compatíveis e endpoints de destino PostgreSQL e MySQL AWS DMS compatíveis. É possível ativar o Time Travel somente para tarefas de carga máxima e captura de dados alterados (CDC) e para tarefas somente CDC. Para ativar o Time Travel ou modificar qualquer configuração existente do Time Travel, interrompa a tarefa de replicação.

As configurações do Time Travel incluem as seguintes propriedades `TTSettings`:

- `EnableTT`: se esta opção estiver definida como`true`, o Time Travel será ativado para a tarefa. O valor padrão é `false`.

  Tipo: booliano

Obrigatório: não

- `EncryptionMode`: o tipo de criptografia do lado do servidor que está sendo utilizada no bucket do S3 para armazenar os dados e logs. É possível especificar `"SSE_S3"` (o padrão) ou `"SSE_KMS"`.

  É possível alterar `EncryptionMode` de `"SSE_KMS"` para`"SSE_S3"`, mas não o contrário.

  Tipo: string

  Obrigatório: Não

- `ServerSideEncryptionKmsKeyId`— Se você especificar `"SSE_KMS"` para`EncryptionMode`, forneça o ID da sua AWS KMS chave gerenciada personalizada. Certifique-se de que a chave que você usa tenha uma política anexada que ative as permissões de usuário AWS Identity and Access Management (IAM) e permita o uso da chave.

  Somente a sua própria chave do KMS simétrica gerenciada e personalizada é compatível com a opção `"SSE_KMS"`.

  Tipo: string

  Obrigatório: somente se `EncryptionMode` for definido como `"SSE_KMS"`

- `ServiceAccessRoleArn`: o nome do recurso da Amazon (ARN) utilizado pelo serviço para acessar o perfil do IAM. Defina o nome do usuário como `dms-tt-s3-access-role`. Essa é uma configuração obrigatória que permite AWS DMS gravar e ler objetos de um bucket do S3.

  Tipo: string

  Obrigatório: se o Time Travel estiver ativado

  Veja a seguir um exemplo de política para esse perfil.

```
{
 "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"
            ]
        }
    ]
}
```

Veja a seguir um exemplo de política de confiança para esse perfil.

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

- `BucketName`: o nome do bucket do S3 para armazenar logs de Time Travel. Crie esse bucket do S3 antes de ativar os logs do Time Travel.

  Tipo: string

  Obrigatório: se o Time Travel estiver ativado

- `BucketFolder`: um parâmetro opcional para definir um nome de pasta no bucket do S3. Se você especificar esse parâmetro, o DMS criará os logs do Time Travel no caminho "/*BucketName*/*BucketFolder*/*taskARN*/*YYYY*/*MM*/*DD*/*hh*". Se você não especificar esse parâmetro, AWS DMS cria o caminho padrão como"/*BucketName*/dms-time-travel-logs/*taskARN*/*YYYY*/*MM*/*DD*/*hh*.

  Tipo: string

Obrigatório: Não

- `EnableDeletingFromS3OnTaskDelete`— Quando essa opção está definida como`true`, AWS DMS exclui os registros de viagem no tempo do S3 se a tarefa for excluída. O valor padrão é `false`.

  Tipo: string

  Obrigatório: Não

- `EnableRawData`: quando esta opção está definida como `true`, os dados brutos da linguagem de manipulação de dados (DML) dos logs do Time Travel aparecem sob a coluna `raw_data` dos logs do Time Travel. Para obter detalhes, consulte Utilizar os logs do Time Travel. O valor padrão é `false`. Quando esta opção é definida como `false`, somente o tipo de DML é capturado.

  Tipo: string

  Obrigatório: Não

- `RawDataFormat`— Nas AWS DMS versões 3.5.0 e superiores, quando `EnableRawData` está definido como. `true` Essa propriedade especifica um formato para os dados brutos da DML em um log do Time Travel e pode ser apresentada como:
  - `"TEXT"`: nomes e valores de colunas analisados e legíveis para eventos da DML capturados durante a CDC como campos `Raw`.
  - `"HEX"`: o hexadecimal original para nomes de colunas e valores capturados para eventos da DML durante a CDC.

  Essa propriedade se aplica às origens de bancos de dados Oracle e Microsoft SQL Server.

  Tipo: string

  Obrigatório: Não

- `OperationsToLog`: especifica o tipo de operação da DML a ser registrado em log nos logs do Time Travel. É possível especificar um dos seguintes:
  - `"INSERT"`
  - `"UPDATE"`
  - `"DELETE"`
  - `"COMMIT"`
  - `"ROLLBACK"`

- `"ALL"`

  O padrão é `"ALL"`.

  Tipo: string

  Obrigatório: Não

- `MaxRecordSize`: especifica o tamanho máximo dos registros de log do Time Travel que são registrados em log para cada linha. Utilize essa propriedade para controlar o crescimento dos logs do Time Travel em tabelas especialmente ocupadas. O padrão é 64 KB.

  Tipo: número inteiro

  Obrigatório: não

Para obter mais informações sobre como ativar e utilizar os logs do Time Travel, consulte os tópicos a seguir.

Tópicos

- Ativar os logs do Time Travel para uma tarefa
- Utilizar os logs do Time Travel
- Com que frequência AWS DMS carrega registros de viagem no tempo para o S3

Ativar os logs do Time Travel para uma tarefa

Você pode ativar a Viagem no Tempo para uma AWS DMS tarefa usando as configurações de tarefa descritas anteriormente. Verifique se a tarefa de replicação é interrompida antes de ativar o Time Travel.

Para ativar a Viagem no Tempo usando o AWS CLI

1. Crie um arquivo JSON de configuração de tarefas do DMS e adicione uma seção `TTSettings` como a seguinte. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

   ```
   .
   .
   .
      },
   ```

```
"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. Em uma ação de tarefa apropriada, especifique esse arquivo JSON utilizando a opção `--replication-task-settings`. Por exemplo, o fragmento de código da CLI a seguir especifica esse arquivo de configurações de Time Travel como parte de `create-replication-task`.

```
aws dms create-replication-task
--target-endpoint-arn arn:aws:dms:us-
east-1:112233445566:endpoint:ELS5O7YTYV452CAZR2EYBNQGILFHQIFVPWFRQAY \
--source-endpoint-arn arn:aws:dms:us-
east-1:112233445566:endpoint:HNX2BWIIN5ZYFF7F6UFFZVWTDFFSMTNOV2FTXZA \
--replication-instance-arn arn:aws:dms:us-
east-1:112233445566:rep:ERLHG2UA52EEJJKFYNYWRPCG6T7EPUAB5AWBUJQ \
--migration-type full-load-and-cdc --table-mappings 'file:///FilePath/
mappings.json' \
--replication-task-settings 'file:///FilePath/task-settings-tt-enabled.json' \
--replication-task-identifier test-task
                            .
                            .
                            .
```

Aqui, o nome desse arquivo de configurações do Time Travel é `task-settings-tt-enabled.json`.

De forma semelhante, é possível especificar esse arquivo como parte da ação `modify-replication-task`.

Observe o tratamento especial dos logs do Time Travel para as seguintes ações de tarefas:

- `start-replication-task`: ao executar uma tarefa de replicação, se um bucket do S3 utilizado pelo Time Travel não estiver acessível, a tarefa será marcada como `FAILED`.
- `stop-replication-task`— Quando a tarefa é interrompida, envia AWS DMS imediatamente todos os registros de viagem no tempo que estão atualmente disponíveis para a instância de replicação para o bucket do S3 usado para viagem no tempo.

Enquanto uma tarefa de replicação é executada, é possível alterar o valor de `EncryptionMode` de `"SSE_KMS"` para `"SSE_S3"`, mas não o contrário.

Se o tamanho dos logs do Time Travel de uma tarefa em andamento exceder 1 GB, o DMS enviará os logs para o S3 em cinco minutos após atingir esse tamanho. Depois que uma tarefa estiver em execução, se o bucket do S3 ou a chave do KMS ficarem inacessíveis, o DMS deixará de enviar logs para esse bucket. Se você achar que seus registros não estão sendo enviados para o bucket do S3, verifique o S3 e AWS KMS as permissões. Para obter mais detalhes sobre a frequência com que o DMS envia esses logs para o S3, consulte Com que frequência AWS DMS carrega registros de viagem no tempo para o S3.

Para ativar o Time Travel para uma tarefa existente no console, utilize a opção do editor JSON em Configurações de tarefa para adicionar uma seção `TTSettings`.

Utilizar os logs do Time Travel

Os Arquivos de log do Time Travel são arquivos de valores separados por vírgulas (CSV) com os campos a seguir.

```
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
```

Depois que os logs do Time Travel estiverem disponíveis no S3, é possível acessá-los e consultá-los diretamente com ferramentas, como o Amazon Athena. Ou é possível baixar os logs da mesma forma utilizada com qualquer arquivo do S3.

O exemplo a seguir mostra um log do Time Travel em que as transações de uma tabela chamada `mytable` são registradas em log. As terminações de linha do log a seguir foram adicionadas para facilitar a leitura.

```
"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'"
```

Com que frequência AWS DMS carrega registros de viagem no tempo para o S3

Para minimizar o uso do armazenamento de sua instância de replicação, AWS DMS descarregue periodicamente os registros de viagem no tempo dela.

Os logs do Time Travel são enviados para o bucket do Amazon S3 nos seguintes casos:

- Se o tamanho atual dos registros exceder 1 GB, AWS DMS faça o upload dos registros para o S3 em cinco minutos. Assim, AWS DMS pode fazer até 12 chamadas por hora para o S3 e AWS KMS para cada tarefa em execução.

- AWS DMS carrega os registros para o S3 a cada hora, independentemente do tamanho dos registros.
- Quando uma tarefa é interrompida, carrega AWS DMS imediatamente os registros de viagem no tempo para o S3.

Configurações de registro de tarefa

O Logging usa CloudWatch a Amazon para registrar informações durante o processo de migração. Usando as configurações de tarefa de registro em log, é possível especificar quais atividades de componentes serão registradas em log e qual quantidade de informações será gravada no log. As configurações da tarefa de registro em log são gravadas em um arquivo JSON. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

Você pode ativar o CloudWatch login de várias maneiras. Você pode selecionar a `EnableLogging` opção AWS Management Console ao criar uma tarefa de migração. Ou você pode definir a `EnableLogging` opção para `true` ao criar uma tarefa usando a AWS DMS API. Também é possível especificar `"EnableLogging": true` no JSON da seção de registro em log de configurações de tarefas.

Quando você define como`true`, `EnableLogging` AWS DMS atribui o nome do CloudWatch grupo e o nome do stream da seguinte forma. Não é possível definir esses valores diretamente.

- CloudWatchLogGroup: `dms-tasks-<REPLICATION_INSTANCE_IDENTIFIER>`
- CloudWatchLogStream: `dms-task-<REPLICATION_TASK_EXTERNAL_RESOURCE_ID>`

`<REPLICATION_INSTANCE_IDENTIFIER>` é o identificador da instância de replicação. `<REPLICATION_TASK_EXTERNAL_RESOURCE_ID>` é o valor da seção `<resourcename>` do ARN da tarefa. Para obter informações sobre como AWS DMS gerar ARNs de recursos, consulteConstruindo um nome de recurso da Amazon (ARN) para AWS DMS.

CloudWatch se integra com AWS Identity and Access Management (IAM), e você pode especificar quais CloudWatch ações um usuário em sua AWS conta pode realizar. Para obter mais informações sobre como trabalhar com o IAM em CloudWatch, consulte Gerenciamento de identidade e acesso para a Amazon CloudWatch e Registro de chamadas de CloudWatch API da Amazon no Guia CloudWatch do usuário da Amazon.

Para excluir os logs de tarefas, é possível definir `DeleteTaskLogs` como verdadeiro no JSON da seção de registro em log das configurações de tarefas.

É possível especificar o registro em log dos seguintes tipos de eventos:

- `FILE_FACTORY`: a fábrica de arquivos gerencia os arquivos utilizados para aplicação e carga em lote e gerencia os endpoints do Amazon S3.
- `METADATA_MANAGER`: o gerenciador de metadados gerencia os metadados de origem e de destino, o particionamento e o estado da tabela durante a replicação.
- `SORTER`: o `SORTER` recebe eventos recebidos do processo `SOURCE_CAPTURE`. Os eventos são agrupados em transações e passados para o componente de serviço `TARGET_APPLY`. Se o processo `SOURCE_CAPTURE` produzir eventos mais rapidamente do que o componente `TARGET_APPLY` pode consumir, o componente `SORTER` armazenará em cache os eventos acumulados no disco ou em um arquivo de troca. Os eventos armazenados em cache são uma causa comum de falta de armazenamento em instâncias de replicação.

  O componente de serviço `SORTER` gerencia os eventos armazenados em cache, reúne estatísticas da CDC e relata a latência da tarefa.
- `SOURCE_CAPTURE`: os dados da replicação contínua (CDC) são capturados no banco de dados ou serviço de origem e passados para o componente de serviço SORTER.
- `SOURCE_UNLOAD`: os dados são descarregados no banco de dados ou serviço de origem durante a carga máxima.
- `TABLES_MANAGER`: o gerenciador de tabelas rastreia as tabelas capturadas, gerencia a ordem da migração da tabela e coleta as estatísticas das tabelas.
- `TARGET_APPLY`: os dados e as instruções da linguagem de definição de dados (DDL) são aplicados no banco de dados de destino.
- `TARGET_LOAD`: os dados são carregados no banco de dados de destino.
- `TASK_MANAGER`: o gerenciador de tarefas gerencia as tarefas em execução e divide as tarefas em subtarefas para processamento paralelo de dados.
- `TRANSFORMATION`: eventos de transformação de mapeamento de tabela. Para ter mais informações, consulte Utilizar o mapeamento de tabela para especificar as configurações da tarefa.
- `VALIDATOR/ VALIDATOR_EXT`: o componente de serviço `VALIDATOR` verifica se os dados foram migrados com precisão da origem para o destino. Para ter mais informações, consulte Validação de dados.

Os seguintes componentes de registro em log geram uma grande quantidade de logs ao utilizar o nível de gravidade do log LOGGER_SEVERITY_DETAILED_DEBUG:

- COMMON
- ADDONS
- DATA_STRUCTURE
- COMMUNICATION
- FILE_TRANSFER
- FILE_FACTORY

Níveis de log diferentes DEFAULT são raramente necessários para esses componentes durante a solução de problemas. Não recomendamos alterar o nível de registro desses componentes, DEFAULT a menos que seja especificamente solicitado pelo AWS Support.

Depois de especificar um dos itens anteriores, é possível especificar a quantidade de informações registradas em log, conforme mostrado na lista a seguir.

Os níveis de severidade estão na ordem do menor para o maior nível de informações. Os níveis mais altos sempre incluem informações dos níveis mais baixos.

- LOGGER_SEVERITY_ERROR: as mensagens de erro são gravadas no log.
- LOGGER_SEVERITY_WARNING: avisos e mensagens de erro são gravados no log.
- LOGGER_SEVERITY_INFO: mensagens informativas, avisos e mensagens de erro são gravados no log.
- LOGGER_SEVERITY_DEFAULT: mensagens informativas, avisos e mensagens de erro são gravados no log.
- LOGGER_SEVERITY_DEBUG: mensagens de depuração, mensagens informativas, mensagens de erro e avisos são gravadas no log.
- LOGGER_SEVERITY_DETAILED_DEBUG: todas as informações são gravadas no log.

O exemplo de JSON a seguir mostra as configurações de tarefa para registrar em log todas as ações e os níveis de gravidade.

```
…
  "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
  },
…
```

Configurações de tarefa de tabela de controle

As tabelas de controle fornecem informações sobre uma AWS DMS tarefa. Elas também fornecem estatísticas úteis que é possível utilizar para planejar e gerenciar a tarefa de migração atual e as tarefas futuras. Você pode aplicar essas configurações de tarefa em um arquivo JSON ou escolhendo Configurações avançadas na página Criar tarefa no AWS DMS console. A tabela Aplicar exceções (`dmslogs.awsdms_apply_exceptions`) é sempre criada nos destinos de bancos de dados. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

AWS DMS cria tabelas de controle somente durante tarefas de carga total + CDC ou somente CDC, e não durante tarefas somente de carga total.

Para tarefas de carga máxima e CDC (migrar dados existentes e replicar alterações em andamento) e de somente CDC (somente replicar alterações de dados), também é possível criar tabelas adicionais, incluindo as seguintes:

- Status da replicação (dmslogs.awsdms_status): esta tabela fornece detalhes sobre a tarefa atual. Isso inclui o status da tarefa, a quantidade de memória consumida pela tarefa e o número de alterações que ainda não foram aplicadas ao destino. Essa tabela também fornece a posição no banco de dados de origem em que AWS DMS está sendo lida atualmente. Ela também indica se a tarefa está na fase de carga máxima ou de captura de dados de alteração (CDC).
- Tabelas suspensas (dmslogs.awsdms_suspended_tables): esta tabela fornece uma lista de tabelas suspensas e o motivo por que foram suspensas.
- Histórico de replicação (dmslogs.awsdms_history) esta tabela fornece informações sobre o histórico de replicação. Essas informações incluem o número e o volume de registros processados durante a tarefa, a latência no final de uma tarefa de CDC e outras estatísticas.

A tabela Aplicar exceções (`dmslogs.awsdms_apply_exceptions`) contém os seguintes parâmetros:

Coluna	Tipo	Descrição
TASK_NAME	nvchar	O ID do recurso da AWS DMS tarefa. O ID do recurso pode ser encontrado no ARN da tarefa.

Coluna	Tipo	Descrição
TABLE_OWNER	nvchar	O proprietário da tabela.
TABLE_NAME	nvchar	O nome da tabela.
ERROR_TIME	timestamp	A hora em que a exceção (erro) ocorreu.
STATEMENT	nvchar	A declaração que estava sendo executada quando o erro ocorreu.
ERRO	nvchar	O nome e a descrição do erro.

A tabela Replication Status (Status de replicação) `dmslogs.awsdms_status` contém o status atual da tarefa e do banco de dados de destino. Ela possui as seguintes configurações.

Coluna	Tipo	Descrição
SERVER_NAME	nvchar	O nome da máquina onde a tarefa de replicação está sendo executada.
TASK_NAME	nvchar	O ID do recurso da AWS DMS tarefa. O ID do recurso pode ser encontrado no ARN da tarefa.
TASK_STATUS	varchar	Um dos seguintes valores: • FULL LOAD • CHANGE PROCESSING (CDC) • NÃO ESTÁ FUNCIONANDO O status da tarefa é definido como FULL LOAD desde que haja pelo menos uma tabela em carga máxima. Após todas as tabelas

Coluna	Tipo	Descrição
		serem carregadas, o status da tarefa será alterado para CHANGE PROCESSING se CDC estiver ativado. A tarefa é definida como NÃO SENDO EXECUTADA antes de você iniciar a tarefa ou após a conclusão da tarefa.
STATUS_TIME	timestamp	O timestamp do status da tarefa.
PENDING_CHANGES	int	O número de registros de alteração que foram confirmados no banco de dados de origem e armazenados em cache na memória e no disco da instância de replicação.
DISK_SWAP_SIZE	int	A quantidade de espaço em disco usada por transações antigas ou descarregadas.
TASK_MEMORY	int	Memória atual utilizada em MB.
SOURCE_CURRENT_POSITION	varchar	A posição no banco de dados de origem que AWS DMS está sendo lida atualmente.
SOURCE_CURRENT_TIMESTAMP	timestamp	O carimbo de data/hora no banco de dados de origem que AWS DMS está sendo lido no momento.
SOURCE_TAIL_POSITION	varchar	A posição da transação de início mais antiga que ainda não está confirmada. Esse valor é a posição mais recente para a qual é possível reverter sem perder alterações.

Coluna	Tipo	Descrição
SOURCE_TAIL _TIMESTAMP	timestamp	O timestamp da transação de início mais antiga que ainda não está confirmada. Esse valor é o timestamp mais recente para o qual é possível reverter sem perder alterações.
SOURCE_TI MESTAMP _APPLIED	timestamp	O timestamp da última confirmaç ão de transação. Em um processo de aplicação em lote, esse valor é o timestamp da confirmação da última transação no lote.

A tabela suspensa (`dmslogs.awsdms_suspended_tables`) contém os seguintes parâmetros.

Coluna	Tipo	Descrição
SERVER_NAME	nvchar	O nome da máquina onde a tarefa de replicação está sendo executada.
TASK_NAME	nvchar	O nome da AWS DMS tarefa
TABLE_OWNER	nvchar	O proprietário da tabela.
TABLE_NAME	nvchar	O nome da tabela.
SUSPEND_REASON	nvchar	Motivo da suspensão.
SUSPEND_T IMESTAMP	timestamp	A hora em que ocorreu a suspensão.

A tabela Replication History (Histórico de replicação) (`dmslogs.awsdms_history`) contém os seguintes parâmetros:

Coluna	Tipo	Descrição
SERVER_NAME	nvchar	O nome da máquina onde a tarefa de replicação está sendo executada.
TASK_NAME	nvchar	O ID do recurso da AWS DMS tarefa. O ID do recurso pode ser encontrado no ARN da tarefa.
TIMESLOT_TYPE	varchar	Um dos seguintes valores: • FULL LOAD • CHANGE PROCESSING (CDC) Se a tarefa estiver executando tanto carga máxima quanto CDC, dois registros de histórico serão gravados no slot de tempo.
TIMESLOT	timestamp	O timestamp de término no slot de tempo.
TIMESLOT_ DURATION	int	A duração do slot de tempo, em minutos.
TIMESLOT_ LATENCY	int	A latência de destino no final do slot de tempo, em segundos. Este valor só se aplica a intervalos de tempo CDC.
RECORDS	int	O número de registros processad os durante o slot de tempo.
TIMESLOT_VOLUME	int	O volume de dados processados em MB.

A tabela Falha de validação (`awsdms_validation_failures_v1`) contém todas as falhas de validação de dados de uma tarefa. Para obter mais informações sobre a validação de dados, consulte Solução de problemas da validação de dados.

Veja a seguir as configurações adicionais de tabela de controle:

- `HistoryTimeslotInMinutes`: utilize esta opção para indicar o tamanho de cada slot de tempo na tabela Histórico de replicação. O padrão é 5 minutos.
- `ControlSchema`— Use essa opção para indicar o nome do esquema do banco de dados para as tabelas de controle do AWS DMS destino. Se você não inserir nenhuma informação nessa opção, as tabelas serão copiadas para o local padrão no banco de dados conforme listado a seguir:
  - PostgreSQL, público
  - Oracle, o esquema de destino
  - Microsoft SQL Server, dbo no banco de dados de destino
  - MySQL, awsdms_control
  - MariaDB, awsdms_control
  - Amazon Redshift, público
  - DynamoDB, criado como tabelas individuais no banco de dados
  - IBM Db2 LUW, awsdms_control

Configurações de tarefas de buffer de stream

Você pode definir as configurações do buffer de transmissão usando o AWS CLI, incluindo o seguinte. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

- `StreamBufferCount`: utilize esta opção para especificar o número de buffers de fluxo de dados da tarefa de migração. O número de buffer de fluxo padrão é 3. Aumentar o valor dessa configuração pode aumentar a velocidade de extração de dados. Contudo, esse aumento de desempenho depende altamente do ambiente de migração, incluindo o sistema de origem e a classe da instância do servidor de replicação. O padrão é suficiente na maioria de situações.
- `StreamBufferSizeInMB`: utilize esta opção para indicar o tamanho máximo de cada buffer de fluxo de dados. O tamanho padrão é 8 MB. Talvez seja preciso aumentar o valor para essa opção ao trabalhar com LOBs muito grandes. Também pode ser preciso aumentar o valor se você receber uma mensagem nos arquivos de log que o tamanho do buffer de fluxo é insuficiente. Ao calcular o tamanho dessa opção, é possível utilizar

a seguinte equação: `[Max LOB size (or LOB chunk size)]*[number of LOB columns]*[number of stream buffers]*[number of tables loading in parallel per task(MaxFullLoadSubTasks)]*3`

- `CtrlStreamBufferSizeInMB`: utilize esta opção para definir o tamanho do buffer de fluxo de controle. O valor é em megabytes e pode ser de 1 a 8. O valor padrão é 5. Pode ser necessário aumentá-lo ao trabalhar com um número muito grande de tabelas, como dezenas de milhares de tabelas.

Configurações de ajuste de processamento de alterações

As configurações a seguir determinam como AWS DMS manipula as alterações nas tabelas de destino durante a captura de dados de alterações (CDC). Várias dessas configurações dependem do valor do parâmetro de metadados de destino `BatchApplyEnabled`. Para obter mais informações sobre o parâmetro `BatchApplyEnabled`, consulte Configurações de tarefa de metadados de destino. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

Veja a seguir as configurações de ajuste do processamento de alterações:

As seguintes configurações aplicam-se somente quando o parâmetro de metadados de destino `BatchApplyEnabled` está definido como `true`.

- `BatchApplyPreserveTransaction`: se estiver definido como `true`, a integridade transacional será preservada e será garantido que um lote contenha todas as alterações dentro de uma transação da origem. O valor padrão é `true`. Essa configuração se aplica apenas a endpoints de destino da Oracle.

  Se definido como `false`, pode haver lapsos temporários na integridade para melhorar o desempenho. Não há garantia de que todas as alterações em uma transação da origem sejam aplicadas ao destino em um único lote.

  Por padrão, AWS DMS os processos mudam em um modo transacional, que preserva a integridade transacional. Se houver condições para lapsos temporários em integridade transacional, você poderá usar a opção batch optimized apply. Essa opção agrupa transações de maneira eficaz e as aplica em lotes para fins de eficiência. A utilização da opção de aplicação otimizada em lote quase sempre viola as restrições de integridade referencial. Portanto, é recomendável desativar essas restrições durante o processo de migração e ativá-las novamente como parte do processo de substituição.

- `BatchApplyTimeoutMin`— Define o tempo mínimo em segundos de AWS DMS espera entre cada aplicação de alterações em lote. O valor padrão é 1.
- `BatchApplyTimeoutMax`— Define o tempo máximo em segundos que AWS DMS espera entre cada aplicação de alterações em lote antes de atingir o tempo limite. O valor padrão é 30.
- `BatchApplyMemoryLimit`: define a quantidade máxima de memória em MB a ser utilizada para pré-processamento no Modo de aplicação otimizada para lotes. O valor padrão é 500.
- `BatchSplitSize`: define o número máximo de alterações aplicadas em um único lote. O valor padrão é 0, o que significa que não há limite aplicado.

As seguintes configurações aplicam-se somente quando o parâmetro de metadados de destino `BatchApplyEnabled` está definido como `false`.

- `MinTransactionSize`: define o número mínimo de alterações a serem incluídas em cada transação. O valor padrão é 1000.
- `CommitTimeout`— Define o tempo máximo em segundos AWS DMS para coletar transações em lotes antes de declarar um tempo limite. O valor padrão é 1.

Para a replicação bidirecional, aplicam-se as seguintes configurações somente quando o parâmetro de metadados de destino `BatchApplyEnabled` está definido como `false`.

- `LoopbackPreventionSettings`: essas configurações fornecem prevenção de loopback para cada tarefa de replicação contínua em qualquer par de tarefas envolvidas na replicação bidirecional. A Prevenção de loopback impede que alterações idênticas sejam aplicadas nas duas direções da replicação bidirecional, o que pode corromper os dados. Para obter mais informações sobre a replicação bidirecional, consulte Executar replicação bidirecional.

AWS DMS tenta manter os dados da transação na memória até que a transação seja totalmente confirmada com a origem, o destino ou ambos. Contudo, as transações maiores que a memória alocada ou não confirmadas dentro do limite de tempo especificado são gravadas no disco.

As seguintes configurações aplicam-se ao ajuste de processamento de alterações, independentemente do modo de processamento de alterações.

- `MemoryLimitTotal`: define o tamanho máximo em MB que todas as transações podem ocupar na memória antes de serem gravadas no disco. O valor padrão é 1024.

- `MemoryKeepTime`: define o tempo máximo em segundos que cada transação pode permanecer na memória antes de ser gravada no disco. A duração é calculada a partir do momento em que AWS DMS começou a capturar a transação. O valor padrão é 60.
- `StatementCacheSize`: define o número máximo de declarações preparadas a serem armazenadas no servidor para execução posterior ao aplicar alterações ao destino. O valor padrão é 50. O valor máximo é 200.

Veja a seguir um exemplo de como as configurações de tarefas que tratam o ajuste de processamento de alterações aparecem em um arquivo JSON de configuração de tarefas:

```
"ChangeProcessingTuning": {
        "BatchApplyPreserveTransaction": true,
        "BatchApplyTimeoutMin": 1,
        "BatchApplyTimeoutMax": 30,
        "BatchApplyMemoryLimit": 500,
        "BatchSplitSize": 0,
        "MinTransactionSize": 1000,
        "CommitTimeout": 1,
        "MemoryLimitTotal": 1024,
        "MemoryKeepTime": 60,
        "StatementCacheSize": 50
}
```

Para controlar a frequência de gravações em um destino do Amazon S3 durante uma tarefa de replicação de dados, é possível configurar os atributos de conexão adicionais `cdcMaxBatchInterval` e `cdcMinFileSize`. Isso pode resultar em melhor desempenho ao analisar os dados sem operações adicionais de sobrecarga. Para ter mais informações, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS.

Configurações da tarefa de validação de dados

É possível verificar se os dados foram migrados com precisão da origem para o destino. Se você habilitar a validação de uma tarefa, AWS DMS começará a comparar os dados de origem e de destino imediatamente após a execução de uma carga completa para uma tabela. Para obter mais informações sobre a validação de dados de tarefa, seus requisitos, o escopo do suporte a seu banco de dados e as métricas relatadas por ela, consulte Validação de dados do AWS DMS. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

As configurações de validação dos dados e seus valores incluem o seguinte:

- `EnableValidation`: ativa a validação de dados quando definida como verdadeira. Caso contrário, a validação será desabilitada para a tarefa. O valor padrão é falso.
- `ValidationMode`: controla como o DMS validará os dados na tabela de destino em relação à tabela de origem. O AWS DMS fornece essa configuração para futura capacidade de extensão. Atualmente, o valor padrão e único válido é`ROW_LEVEL`. AWS DMS valida todas as linhas entre as tabelas de origem e de destino.
- `FailureMaxCount`: especifica o número máximo de registros que podem falhar na validação antes que a validação seja suspensa para a tarefa. O valor padrão é 10.000. Para que a validação continue, independentemente do número de registros que fizeram com que a validação falhasse, defina essa configuração como um valor superior ao número de registros na origem.
- `HandleCollationDiff`: quando esta opção está definida como `true`, a validação considera as diferenças de agrupamento de colunas nos endpoints do PostgreSQL e do Microsoft SQL Server ao identificar os registros de origem e de destino a serem comparados. Caso contrário, qualquer diferença no agrupamento de colunas será ignorada para validação. Os agrupamentos de colunas podem ditar a ordem das linhas, o que é importante para a validação dos dados. Configurar `HandleCollationDiff` como true resolve essas diferenças de agrupamento automaticamente e impede falsos positivos na validação dos dados. O valor padrão é `false`.
- `RecordFailureDelayInMinutes`: especifica o atraso, em minutos, antes de relatar qualquer detalhe da falha da validação.
- `RecordFailureDelayLimitInMinutes`: especifica o atraso antes de relatar qualquer detalhe da falha da validação. Normalmente, o AWS DMS utiliza a latência da tarefa para reconhecer o atraso real da chegada das alterações ao destino para evitar falsos positivos. Essa configuração substitui o valor de atraso real e permite que você defina um atraso maior antes de relatar qualquer métrica de validação. O valor padrão é 0.
- `RecordSuspendDelayInMinutes`: especifica o tempo de atraso, em minutos, antes que as tabelas sejam suspensas da validação devido ao limite de erro definido em `FailureMaxCount`.
- `SkipLobColumns`— Quando essa opção é definida como`true`, AWS DMS ignora a validação de dados para todas as colunas LOB na parte da tabela da validação da tarefa. O valor padrão é `false`.
- `TableFailureMaxCount`: especifica o número máximo de linhas em uma tabela em que uma validação pode falhar antes que a validação seja suspensa para a tabela. O valor padrão é 1,000.
- `ThreadCount`— Especifica o número de threads de execução AWS DMS usados durante a validação. Cada thread seleciona not-yet-validated dados da origem e do destino para comparar

e validar. O valor padrão é 5. Se você `ThreadCount` definir um número maior, AWS DMS poderá concluir a validação mais rapidamente. No entanto, o AWS DMS executa mais consultas simultâneas que consomem mais recursos na origem e no destino.

- `ValidationOnly`: quando esta opção está definida como `true`, a tarefa executa a validação dos dados sem executar nenhuma migração ou replicação de dados. O valor padrão é `false`. Não é possível modificar a configuração de `ValidationOnly` após a tarefa ser criada.

  Você deve definir como TargetTablePrepModeDO_NOTHING(o padrão para uma tarefa somente de validação) e definir o Tipo de migração como um dos seguintes:

  - Carga total — Defina o tipo de migração da tarefa para migrar dados existentes no AWS DMS console. Ou, na AWS DMS API, defina o tipo de migração como FULL-LOAD.
  - CDC: defina o Tipo de migração da tarefa para Replicar somente alterações de dados no console do AWS DMS . Ou, na AWS DMS API, defina o tipo de migração como CDC.

  Independentemente do tipo de migração escolhido, os dados não são realmente migrados ou replicados durante uma tarefa somente de validação.

  Para ter mais informações, consulte Tarefas somente de validação.

  > **Important**
  >
  > A configuração de `ValidationOnly` é imutável. Ela não pode ser modificada para uma tarefa após a criação dessa tarefa.

- `ValidationPartialLobSize`: especifica se você deseja fazer uma validação parcial das colunas LOB em vez de validar todos os dados armazenados na coluna. Isso é algo que pode ser útil ao migrar apenas parte dos dados de LOB e não todo o conjunto de dados de LOB. O valor está em unidades de KB. O valor padrão é 0, o que significa que o AWS DMS valida todos os dados da coluna LOB. Por exemplo, `"ValidationPartialLobSize": 32` significa que valida AWS DMS somente os primeiros 32 KB dos dados da coluna na origem e no destino.
- `PartitionSize`: especifica o tamanho do lote de registros a serem lidos para comparação da origem e do destino. O padrão é 10.000.
- `ValidationQueryCdcDelaySeconds`: a quantidade de tempo em que a primeira consulta de validação é atrasada na origem e no destino de cada atualização da CDC. Isso pode ajudar a reduzir a contenção de recursos quando a latência da migração é alta. Uma tarefa somente de validação define automaticamente essa opção como 180 segundos. O padrão é 0.

Por exemplo, o JSON a seguir permite a validação de dados com duas vezes o número padrão de threads. Ele também considera as diferenças na ordem dos registros provocadas pelas diferenças no agrupamento de colunas nos endpoints do PostgreSQL. Além disso, ele fornece um atraso nos relatórios de validação para considerar tempo adicional para processar todas as falhas de validação.

```
"ValidationSettings": {
     "EnableValidation": true,
     "ThreadCount": 10,
     "HandleCollationDiff": true,
     "RecordFailureDelayLimitInMinutes": 30
  }
```

> **Note**
>
> Para um endpoint Oracle, AWS DMS usa DBMS_CRYPTO para validar BLOBs. Se o endpoint do Oracle utilizar BLOBs, conceda a permissão `execute` para DBMS_CRYPTO para a conta de usuário que acessa o endpoint do Oracle. Faça isso executando a seguinte instrução.
>
> ```
> grant execute on sys.dbms_crypto to dms_endpoint_user;
> ```

Configurações de tarefa para processamento de DDL de processamento de alterações

As configurações a seguir determinam como AWS DMS manipula as alterações da linguagem de definição de dados (DDL) nas tabelas de destino durante a captura de dados de alteração (CDC). Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

As configurações de tarefas para lidar com DDL de processamento de alterações incluem:

- `HandleSourceTableDropped` – defina esta opção como `true` para abandonar a tabela de destino quando a tabela de origem for abandonada.
- `HandleSourceTableTruncated`: defina esta opção como `true` para truncar a tabela de destino quando a tabela de origem for truncada.

- `HandleSourceTableAltered`: defina esta opção como `true` para alterar a tabela de destino quando a tabela de origem for alterada.

Veja a seguir um exemplo de como as configurações de tarefas que manipulam o DDL de processamento de alterações aparecem em um arquivo JSON de configuração de tarefas:

```
"ChangeProcessingDdlHandlingPolicy": {
   "HandleSourceTableDropped": true,
   "HandleSourceTableTruncated": true,
   "HandleSourceTableAltered": true
},
```

> **Note**
>
> Para obter informações sobre quais instruções DDL são compatíveis com um endpoint específico, consulte o tópico que descreve esse endpoint.

Configurações da tarefa de substituição de caracteres

Você pode especificar que sua tarefa de replicação realize substituições de caracteres no banco de dados de destino para todas as colunas do banco de dados de origem com o tipo de `WSTRING` dados AWS DMS `STRING` ou. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

É possível configurar a substituição de caracteres para qualquer tarefa com endpoints dos seguintes bancos de dados de origem e de destino:

- Bancos de dados de origem:
  - Oracle
  - Microsoft SQL Server
  - MySQL
  - PostgreSQL
  - SAP Adaptive Server Enterprise (ASE)
  - IBM Db2 LUW

- Bancos de dados de destino:
  - Oracle
  - Microsoft SQL Server
  - MySQL
  - PostgreSQL
  - SAP Adaptive Server Enterprise (ASE)
  - Amazon Redshift

É possível especificar substituições de caracteres utilizando o parâmetro `CharacterSetSettings` nas configurações da tarefa. Essas substituições de caracteres ocorrem para caracteres especificados utilizando o valor de ponto de código Unicode em notação hexadecimal. É possível implementar as substituições em duas fases, na seguinte ordem, se ambas forem especificadas:

1. Substituição individual de caracteres — AWS DMS pode substituir os valores dos caracteres selecionados na fonte por valores de substituição especificados dos caracteres correspondentes no destino. Utilize a matriz `CharacterReplacements` no `CharacterSetSettings` para selecionar todos os caracteres de origem com os pontos de código Unicode que você especificar. Utilize essa matriz também para especificar os pontos de código de substituição para os caracteres correspondentes no destino.

   Para selecionar todos os caracteres na origem que têm um determinado ponto de código, defina uma instância de `SourceCharacterCodePoint` na matriz `CharacterReplacements` para esse ponto de código. Depois, especifique o ponto de código de substituição para todos os caracteres de destino equivalentes definindo a instância correspondente de `TargetCharacterCodePoint` nessa matriz. Para excluir caracteres de destino em vez de substituí-los, defina as instâncias apropriadas de `TargetCharacterCodePoint` como zero (0). É possível substituir ou excluir quantos valores diferentes de caracteres de destino desejar especificando pares adicionais de configurações `SourceCharacterCodePoint` e `TargetCharacterCodePoint` na matriz `CharacterReplacements`. Se você especificar o mesmo valor para várias instâncias de `SourceCharacterCodePoint`, o valor da última configuração correspondente de `TargetCharacterCodePoint` se aplicará ao destino.

   Por exemplo, suponha que você especifique os seguintes valores para `CharacterReplacements`.

```
"CharacterSetSettings": {
```

```
    "CharacterReplacements": [ {
        "SourceCharacterCodePoint": 62,
        "TargetCharacterCodePoint": 61
        }, {
        "SourceCharacterCodePoint": 42,
        "TargetCharacterCodePoint": 41
        }
    ]
}
```

Neste exemplo, AWS DMS substitui todos os caracteres com o valor hexadecimal 62 do ponto do código-fonte no destino por caracteres com o valor 61 do ponto de código. Além disso, AWS DMS substitui todos os caracteres com o ponto 42 do código-fonte no destino por caracteres com o valor 41 do ponto de código. Em outras palavras, o AWS DMS substitui todas as instâncias da letra `'b'` no destino pela letra `'a'`. Da mesma forma, AWS DMS substitui todas as instâncias da letra `'B'` no alvo pela letra`'A'`.

2. Validação e substituição do conjunto de caracteres — Após a conclusão de qualquer substituição individual de caracteres, AWS DMS pode garantir que todos os caracteres de destino tenham pontos de código Unicode válidos no único conjunto de caracteres que você especificar. Utilize o `CharacterSetSupport` em `CharacterSetSettings` para configurar essa verificação e modificação de caracteres de destino. Para especificar o conjunto de caracteres de verificação, defina `CharacterSet` em `CharacterSetSupport` como o valor de string do conjunto de caracteres. (Os valores possíveis de `CharacterSet` se seguem.) Você pode AWS DMS modificar os caracteres de destino inválidos de uma das seguintes formas:

   - Especifique um único ponto de código Unicode de substituição para todos os caracteres de destino inválidos, independentemente do ponto de código atual. Para configurar esse ponto de código de substituição, defina `ReplaceWithCharacterCodePoint` em `CharacterSetSupport` como o valor especificado.
   - Configure a exclusão de todos os caracteres de destino inválidos definindo `ReplaceWithCharacterCodePoint` como zero (0).

   Por exemplo, suponha que você especifique os seguintes valores para `CharacterSetSupport`.

```
"CharacterSetSettings": {
    "CharacterSetSupport": {
        "CharacterSet": "UTF16_PlatformEndian",
        "ReplaceWithCharacterCodePoint": 0
    }
```

```
}
```

Neste exemplo, AWS DMS exclui todos os caracteres encontrados no destino que são inválidos no conjunto de `"UTF16_PlatformEndian"` caracteres. Portanto, todos os caracteres especificados com o valor hexadecimal `2FB6` são excluídos. Esse valor é inválido porque esse é um ponto de código Unicode de 4 bytes e os conjuntos de caracteres UTF16 aceitam apenas caracteres com pontos de código de 2 bytes.

> **Note**
>
> A tarefa de replicação conclui todas as substituições de caracteres especificadas antes de iniciar qualquer transformação global ou de tabela especificada por meio do mapeamento de tabela. Para obter mais informações sobre o mapeamento de tabela, consulte Utilizar o mapeamento de tabela para especificar as configurações da tarefa.
>
> A substituição de caracteres não é compatível com tipos de dados LOB. Isso inclui qualquer tipo de dados que o DMS considere um tipo de dados LOB. Por exemplo, o tipo de dados `Extended` no Oracle é considerado LOB. Para obter mais informações sobre tags, consulte Tipos de dados de origem do Oracle:

Os valores que AWS DMS oferecem suporte para `CharacterSet` aparecem na tabela a seguir.

`UTF-8`	`ibm-860_P100-1995`	`ibm-280_P100-1995`
`UTF-16`	`ibm-861_P100-1995`	`ibm-284_P100-1995`
`UTF-16BE`	`ibm-862_P100-1995`	`ibm-285_P100-1995`
`UTF-16LE`	`ibm-863_P100-1995`	`ibm-290_P100-1995`
`UTF-32`	`ibm-864_X110-1999`	`ibm-297_P100-1995`
`UTF-32BE`	`ibm-865_P100-1995`	`ibm-420_X120-1999`
`UTF-32LE`	`ibm-866_P100-1995`	`ibm-424_P100-1995`
`UTF16_PlatformEndian`	`ibm-867_P100-1998`	`ibm-500_P100-1995`

UTF16_OppositeEndian	ibm-868_P100-1995	ibm-803_P100-1999
UTF32_PlatformEndian	ibm-869_P100-1995	ibm-838_P100-1995
UTF32_OppositeEndian	ibm-878_P100-1996	ibm-870_P100-1995
UTF-16BE,version=1	ibm-901_P100-1999	ibm-871_P100-1995
UTF-16LE,version=1	ibm-902_P100-1999	ibm-875_P100-1995
UTF-16,version=1	ibm-922_P100-1999	ibm-918_P100-1995
UTF-16,version=2	ibm-1168_P100-2002	ibm-930_P120-1999
UTF-7	ibm-4909_P100-1999	ibm-933_P110-1995
IMAP-mailbox-name	ibm-5346_P100-1998	ibm-935_P110-1999
SCSU	ibm-5347_P100-1998	ibm-937_P110-1999
BOCU-1	ibm-5348_P100-1997	ibm-939_P120-1999
CESU-8	ibm-5349_P100-1998	ibm-1025_P100-1995
ISO-8859-1	ibm-5350_P100-1998	ibm-1026_P100-1995
US-ASCII	ibm-9447_P100-2002	ibm-1047_P100-1995
gb18030	ibm-9448_X100-2005	ibm-1097_P100-1995
ibm-912_P100-1995	ibm-9449_P100-2002	ibm-1112_P100-1995
ibm-913_P100-2000	ibm-5354_P100-1998	ibm-1122_P100-1999
ibm-914_P100-1995	ibm-1250_P100-1995	ibm-1123_P100-1995
ibm-915_P100-1995	ibm-1251_P100-1995	ibm-1130_P100-1997
ibm-1089_P100-1995	ibm-1252_P100-2000	ibm-1132_P100-1998
ibm-9005_X110-2007	ibm-1253_P100-1995	ibm-1137_P100-1999

ibm-813_P100-1995	ibm-1254_P100-1995	ibm-4517_P100-2005
ibm-5012_P100-1999	ibm-1255_P100-1995	ibm-1140_P100-1997
ibm-916_P100-1995	ibm-5351_P100-1998	ibm-1141_P100-1997
ibm-920_P100-1995	ibm-1256_P110-1997	ibm-1142_P100-1997
iso-8859_10-1998	ibm-5352_P100-1998	ibm-1143_P100-1997
iso-8859_11-2001	ibm-1257_P100-1995	ibm-1144_P100-1997
ibm-921_P100-1995	ibm-5353_P100-1998	ibm-1145_P100-1997
iso-8859_14-1998	ibm-1258_P100-1997	ibm-1146_P100-1997
ibm-923_P100-1998	macos-0_2-10.2	ibm-1147_P100-1997
ibm-942_P12A-1999	macos-6_2-10.4	ibm-1148_P100-1997
ibm-943_P15A-2003	macos-7_3-10.2	ibm-1149_P100-1997
ibm-943_P130-1999	macos-29-10.2	ibm-1153_P100-1999
ibm-33722_P12A_P12A-2009_U2	macos-35-10.2	ibm-1154_P100-1999
ibm-33722_P120-1999	ibm-1051_P100-1995	ibm-1155_P100-1999
ibm-954_P101-2007	ibm-1276_P100-1995	ibm-1156_P100-1999
euc-jp-2007	ibm-1006_P100-1995	ibm-1157_P100-1999
ibm-1373_P100-2002	ibm-1098_P100-1995	ibm-1158_P100-1999
windows-950-2000	ibm-1124_P100-1996	ibm-1160_P100-1999
ibm-950_P110-1999	ibm-1125_P100-1997	ibm-1164_P100-1999
ibm-1375_P100-2008	ibm-1129_P100-1997	ibm-1364_P110-2007
ibm-5471_P100-2006	ibm-1131_P100-1997	ibm-1371_P100-1999

ibm-1386_P100-2001	ibm-1133_P100-1997	ibm-1388_P103-2001
windows-936-2000	ISO_2022,locale=ja,version=0	ibm-1390_P110-2003
ibm-1383_P110-1999	ISO_2022,locale=ja,version=1	ibm-1399_P110-2003
ibm-5478_P100-1995	ISO_2022,locale=ja,version=2	ibm-5123_P100-1999
euc-tw-2014	ISO_2022,locale=ja,version=3	ibm-8482_P100-1999
ibm-964_P110-1999	ISO_2022,locale=ja,version=4	ibm-16684_P110-2003
ibm-949_P110-1999	ISO_2022,locale=ko,version=0	ibm-4899_P100-1998
ibm-949_P11A-1999	ISO_2022,locale=ko,version=1	ibm-4971_P100-1999
ibm-970_P110_P110-2006_U2	ISO_2022,locale=zh,version=0	ibm-9067_X100-2005
ibm-971_P100-1995	ISO_2022,locale=zh,version=1	ibm-12712_P100-1998
ibm-1363_P11B-1998	ISO_2022,locale=zh,version=2	ibm-16804_X110-1999
ibm-1363_P110-1997	HZ	ibm-37_P100-1995,swaplfnl
windows-949-2000	x11-compound-text	ibm-1047_P100-1995,swaplfnl
windows-874-2000	ISCII,version=0	ibm-1140_P100-1997,swaplfnl

ibm-874_P100-1995	ISCII,version=1	ibm-1141_P100-1997,swaplfnl
ibm-1162_P100-1999	ISCII,version=2	ibm-1142_P100-1997,swaplfnl
ibm-437_P100-1995	ISCII,version=3	ibm-1143_P100-1997,swaplfnl
ibm-720_P100-1997	ISCII,version=4	ibm-1144_P100-1997,swaplfnl
ibm-737_P100-1997	ISCII,version=5	ibm-1145_P100-1997,swaplfnl
ibm-775_P100-1996	ISCII,version=6	ibm-1146_P100-1997,swaplfnl
ibm-850_P100-1995	ISCII,version=7	ibm-1147_P100-1997,swaplfnl
ibm-851_P100-1995	ISCII,version=8	ibm-1148_P100-1997,swaplfnl
ibm-852_P100-1995	LMBCS-1	ibm-1149_P100-1997,swaplfnl
ibm-855_P100-1995	ibm-37_P100-1995	ibm-1153_P100-1999,swaplfnl
ibm-856_P100-1995	ibm-273_P100-1995	ibm-12712_P100-1998,swaplfnl
ibm-857_P100-1995	ibm-277_P100-1995	ibm-16804_X110-1999,swaplfnl
ibm-858_P100-1997	ibm-278_P100-1995	ebcdic-xml-us

Configurações de tarefa de imagem anterior

Ao gravar atualizações de CDC em um destino de streaming de dados, como o Kinesis ou o Apache Kafka, é possível exibir os valores originais de uma linha do banco de dados de origem antes da alteração por uma atualização. Para tornar isso possível, AWS DMS preenche uma imagem anterior dos eventos de atualização com base nos dados fornecidos pelo mecanismo do banco de dados de origem. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

Para isso, utilize o parâmetro `BeforeImageSettings` que adiciona um novo atributo JSON a cada operação de atualização com valores coletados do sistema do banco de dados de origem.

Aplique `BeforeImageSettings` somente às tarefas de carga máxima mais CDC ou às tarefas de CDC somente. As tarefas de carga máxima mais CDC migram os dados existentes e replicam as alterações em andamento. As tarefas somente de CDC replicam somente as alterações de dados.

Não aplique `BeforeImageSettings` a tarefas que são somente de carga total.

As opções possíveis para `BeforeImageSettings` são as seguintes:

- `EnableBeforeImage`: ativa a geração de imagem anterior quando definido como `true`. O padrão é `false`.
- `FieldName`: atribui um nome ao novo atributo JSON. Quando `EnableBeforeImage` for `true`, `FieldName` será necessário e não poderá estar vazio.
- `ColumnFilter`: especifica uma coluna a ser adicionada utilizando a geração de imagem anterior. Para adicionar somente colunas que fazem parte das chaves primárias da tabela, use o valor padrão, `pk-only`. Para adicionar qualquer coluna que tenha um valor de imagem anterior, use `all`. Observe que a imagem anterior não é compatível com tipos de dados de objetos binários grandes (LOB), como CLOB e BLOB.

O exemplo a seguir mostra a utilização de `BeforeImageSettings`.

```
"BeforeImageSettings": {
    "EnableBeforeImage": true,
    "FieldName": "before-image",
    "ColumnFilter": "pk-only"
  }
```

Para obter informações sobre as configurações de imagem anterior para o Kinesis, incluindo configurações adicionais de mapeamento de tabela, consulte Utilizar uma imagem anterior para visualizar valores originais de linhas da CDC para um fluxo de dados do Kinesis como destino.

Para obter informações sobre as configurações de imagem anterior para o Kafka, incluindo configurações adicionais de mapeamento de tabela, consulte Utilizar uma imagem anterior para visualizar os valores originais de linhas da CDC para o Apache Kafka como destino.

Configurações de tarefa de tratamento de erros

É possível definir o comportamento do tratamento de erros da tarefa de replicação ao utilizar as seguintes configurações. Para obter informações sobre como utilizar um arquivo de configuração de tarefas para definir as configurações de tarefas, consulte Exemplo de configurações de tarefas.

- `DataErrorPolicy`— Determina a ação que o AWS DMS executa quando há um erro relacionado ao processamento de dados no nível do registro. Erros de conversão, erros de transformação e dados ruins são alguns exemplos de erros de processamento de dados. O padrão é `LOG_ERROR`.
  - `IGNORE_RECORD`: a tarefa continua, e os dados daquele registro são ignorados. O contador de erros para a propriedade `DataErrorEscalationCount` foi incrementado. Portanto, se você definir um limite de erros para uma tabela, este erro será contado para esse limite.
  - `LOG_ERROR`: a tarefa continua, e o erro é gravado no log de tarefas.
  - `SUSPEND_TABLE`: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - `STOP_TASK`: a tarefa é interrompida, e intervenção manual é requerida.
- `DataTruncationErrorPolicy`: determina a ação que o AWS DMS realiza quando os dados estão truncados. O padrão é `LOG_ERROR`.
  - `IGNORE_RECORD`: a tarefa continua, e os dados daquele registro são ignorados. O contador de erros para a propriedade `DataErrorEscalationCount` foi incrementado. Portanto, se você definir um limite de erros para uma tabela, este erro será contado para esse limite.
  - `LOG_ERROR`: a tarefa continua, e o erro é gravado no log de tarefas.
  - `SUSPEND_TABLE`: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - `STOP_TASK`: a tarefa é interrompida, e intervenção manual é requerida.
- `DataErrorEscalationPolicy`: determina a ação que o AWS DMS realiza quando o número máximo de erros (definido no parâmetro `DataErrorEscalationCount`) é atingido. O padrão é `SUSPEND_TABLE`.

  - SUSPEND_TABLE: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - STOP_TASK: a tarefa é interrompida, e intervenção manual é requerida.
- DataErrorEscalationCount: define o número máximo de erros que podem ocorrer nos dados de um registro específico. Quando esse número é atingido, os dados da tabela que contém o registro de erro são tratados de acordo com a política definida em DataErrorEscalationPolicy. O padrão é 0.
- EventErrorPolicy— Determina a ação que o AWS DMS executa quando ocorre um erro ao enviar um evento relacionado à tarefa. Os valores possíveis são
  - IGNORE: a tarefa continua e todos os dados associados a esse evento são ignorados.
  - STOP_TASK: a tarefa é interrompida, e intervenção manual é requerida.
- TableErrorPolicy: determina a ação que o AWS DMS realiza quando um erro ocorre durante o processamento de dados ou de metadados de uma tabela específica. Esse erro só se aplica a dados gerais da tabela e é relacionado a um registro específico. O padrão é SUSPEND_TABLE.
  - SUSPEND_TABLE: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - STOP_TASK: a tarefa é interrompida, e intervenção manual é requerida.
- TableErrorEscalationPolicy: determina a ação que o AWS DMS realiza quando o número máximo de erros (definido utilizando o parâmetro TableErrorEscalationCount). A configuração de usuário padrão, e única, é STOP_TASK, onde a tarefa é interrompida e é necessária intervenção manual.
- TableErrorEscalationCount: o número máximo de erros que podem ocorrer devido a dados ou metadados gerais de uma tabela específica. Quando esse número é atingido, os dados da tabela são manipulados de acordo com a política definida em TableErrorEscalationPolicy. O padrão é 0.
- RecoverableErrorCount: o número máximo de tentativas feitas para reiniciar uma tarefa quando um erro ambiental ocorre. Após o sistema tentar reiniciar a tarefa o número designado de vezes, ela será interrompida e será necessária intervenção manual. O valor padrão é -1, que instrui AWS DMS a tentar reiniciar a tarefa indefinidamente. Quando você define esse valor como -1, o número de novas tentativas que o DMS tenta varia com base no tipo de erro retornado da seguinte forma:
  - Estado de execução, erro recuperável: se ocorrer um erro recuperável, como uma conexão perdida ou uma falha na aplicação de destino, o DMS repetirá a tarefa nove vezes.
  - Estado inicial, erro recuperável: o DMS repete a tarefa seis vezes.

- Estado de execução, erro fatal tratado pelo DMS: o DMS repete a tarefa seis vezes.
- Estado de execução, erro fatal não tratado pelo DMS: o DMS não repete a tarefa.

Defina esse valor como 0 para nunca tentar reiniciar uma tarefa.

Recomendamos que você `RecoverableErrorInterval` defina valores de `RecoverableErrorCount` e para que haja repetições suficientes em intervalos suficientes para que sua tarefa do DMS se recupere adequadamente. Se ocorrer um erro fatal, o DMS interrompe as tentativas de reinicialização na maioria dos cenários.

- `RecoverableErrorInterval`— O número de segundos que o AWS DMS espera entre as tentativas de reiniciar uma tarefa. O padrão é 5.
- `RecoverableErrorThrottling`: quando ativado, o intervalo entre as tentativas de reiniciar uma tarefa é aumentado em uma série com base no valor de `RecoverableErrorInterval`. Por exemplo, se `RecoverableErrorInterval` estiver definido como 5 segundos, a próxima tentativa ocorrerá após 10 segundos, depois 20, depois 40 segundos e assim por diante. O padrão é `true`.
- `RecoverableErrorThrottlingMax`— O número máximo de segundos que o AWS DMS espera entre as tentativas de reiniciar uma tarefa, se `RecoverableErrorThrottling` estiver ativado. O padrão é 1800.
- `RecoverableErrorStopRetryAfterThrottlingMax`— Quando definido como`true`, interrompe a reinicialização da tarefa após atingir o número máximo de segundos de AWS DMS espera entre as tentativas de recuperação, por. `RecoverableErrorThrottlingMax`
- `ApplyErrorDeletePolicy`: determina a ação que o AWS DMS realiza quando há um conflito com uma operação DELETE. O padrão é `IGNORE_RECORD`. Os valores possíveis são os seguintes:
  - `IGNORE_RECORD`: a tarefa continua, e os dados daquele registro são ignorados. O contador de erros para a propriedade `ApplyErrorEscalationCount` foi incrementado. Portanto, se você definir um limite de erros para uma tabela, este erro será contado para esse limite.
  - `LOG_ERROR`: a tarefa continua, e o erro é gravado no log de tarefas.
  - `SUSPEND_TABLE`: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - `STOP_TASK`: a tarefa é interrompida, e intervenção manual é requerida.
- `ApplyErrorInsertPolicy`: determina a ação que o AWS DMS realiza quando há um conflito com uma operação INSERT. O padrão é `LOG_ERROR`. Os valores possíveis são os seguintes:

  - `IGNORE_RECORD`: a tarefa continua, e os dados daquele registro são ignorados. O contador de erros para a propriedade `ApplyErrorEscalationCount` foi incrementado. Portanto, se você definir um limite de erros para uma tabela, este erro será contado para esse limite.
  - `LOG_ERROR`: a tarefa continua, e o erro é gravado no log de tarefas.
  - `SUSPEND_TABLE`: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - `STOP_TASK`: a tarefa é interrompida, e intervenção manual é requerida.
  - `INSERT_RECORD`: se houver um registro de destino existente com a mesma chave primária que o registro de origem inserido, o registro de destino será atualizado.
- `ApplyErrorUpdatePolicy`: determina a ação que o AWS DMS realiza quando há um conflito de dados ausentes com uma operação UPDATE. O padrão é `LOG_ERROR`. Os valores possíveis são os seguintes:
  - `IGNORE_RECORD`: a tarefa continua, e os dados daquele registro são ignorados. O contador de erros para a propriedade `ApplyErrorEscalationCount` foi incrementado. Portanto, se você definir um limite de erros para uma tabela, este erro será contado para esse limite.
  - `LOG_ERROR`: a tarefa continua, e o erro é gravado no log de tarefas.
  - `SUSPEND_TABLE`: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - `STOP_TASK`: a tarefa é interrompida, e intervenção manual é requerida.
  - `UPDATE_RECORD`— Se o registro de destino estiver ausente, o registro de destino ausente será inserido na tabela de destino. AWS DMS desativa completamente o suporte da coluna LOB para a tarefa. Selecionar essa opção requer que o registro em log suplementar total seja ativado para todas as colunas da tabela de origem quando o Oracle é o banco de dados de origem.
- `ApplyErrorEscalationPolicy`— Determina qual ação o AWS DMS executa quando o número máximo de erros (definido usando o `ApplyErrorEscalationCount` parâmetro) é atingido. O padrão é LOG_ERROR:
  - `LOG_ERROR`: a tarefa continua, e o erro é gravado no log de tarefas.
  - `SUSPEND_TABLE`: a tarefa continua, mas os dados da tabela com o registro do erro são movidos para um estado de erro e os dados não são replicados.
  - `STOP_TASK`: a tarefa é interrompida, e intervenção manual é requerida.
- `ApplyErrorEscalationCount`: esta opção define o número máximo de conflitos de APPLY que pode ocorrer para uma tabela específica durante uma operação de processo de alteração. Quando

esse número é atingido, os dados da tabela são manipulados de acordo com a política definida no parâmetro `ApplyErrorEscalationPolicy`. O padrão é 0.

- `ApplyErrorFailOnTruncationDdl`: defina esta opção como `true` para fazer com que a tarefa falhe quando um truncamento for executado em uma das tabelas rastreadas durante a CDC. O padrão é `false`.

  Essa abordagem não funciona com o PostgreSQL versão 11.x ou inferior, ou qualquer outro endpoint de origem que não replica o truncamento da tabela de DDL.

- `FailOnNoTablesCaptured`: defina esta opção como `true` para que uma tarefa falhe quando os mapeamentos de tabelas definidos para uma tarefa não encontrarem tabelas quando a tarefa for iniciada. O padrão é `false`.

- `FailOnTransactionConsistencyBreached`: esta opção se aplica a tarefas que utilizam o Oracle como uma origem com CDC. O padrão é falso. Defina como `true` para fazer com que uma tarefa falhe quando uma transação permanecer aberta por um tempo maior do que o limite especificado e puder ser descartada.

  Quando uma tarefa do CDC começa com o Oracle, AWS DMS aguarda por um tempo limitado até que a transação aberta mais antiga seja fechada antes de iniciar o CDC. Se a transação aberta mais antiga não fechar até que o tempo limite seja atingido, na maioria dos casos AWS DMS iniciará o CDC, ignorando essa transação. Se essa opção for definida como `true`, a tarefa falhará.

- `FullLoadIgnoreConflicts`— Defina essa opção `true` para AWS DMS ignorar os erros de “zero linhas afetadas” e “duplicatas” ao aplicar eventos em cache. Se definido como`false`, AWS DMS relata todos os erros em vez de ignorá-los. O padrão é `true`.

Observe que os erros de carregamento de tabela no Redshift como destino são relatados em `STL_LOAD_ERRORS`. Para obter mais informações, consulte STL_LOAD_ERRORS no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Salvar configurações de tarefa

É possível salvar as configurações de tarefa como um arquivo JSON caso você queira reutilizá-las em outra tarefa. É possível encontrar as configurações de tarefas para copiar para um arquivo JSON na seção Detalhes da visão geral de uma tarefa.

> **Note**
>
> Ao reutilizar as configurações de tarefas para outras tarefas, remova todos os atributos `CloudWatchLogGroup` e `CloudWatchLogStream`. Caso contrário, o seguinte erro será fornecido: MENSAGEM DE ERRO DO SISTEMA:Configurações da tarefa CloudWatchLogGroup ou CloudWatchLogStream não podem ser definidas na criação.

Por exemplo, o seguinte arquivo JSON contém as configurações salvas de uma tarefa.

```
{
    "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
}
```

Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS

Objetos binários grandes (LOBs), às vezes, podem ser difíceis de migrar entre sistemas. O AWS DMS oferece uma variedade de opções para ajudar com o ajuste de colunas de LOBs. Para ver quais tipos de dados e quando são considerados LOBs AWS DMS, consulte a AWS DMS documentação.

ao migrar dados de um banco de dados para outro, aproveite a oportunidade para repensar como os seus LOBS são armazenados, especialmente para migrações heterogêneas. Se quiser fazer isso, não haverá necessidade de migrar os dados de LOB.

Se você quiser incluir LOBs, poderá escolher as outras configurações de LOB:

- O modo LOB determina como os LOBs são processados:
  - Modo LOB completo — No modo LOB completo, AWS DMS migra todos os LOBs da origem para o destino, independentemente do tamanho. Nessa configuração, não AWS DMS há informações sobre o tamanho máximo de LOBs esperado. Assim, migra-se um LOB por vez, parte por parte. Esse modo pode ser bem lento.
  - Modo LOB limitado: no modo LOB limitado, você define o tamanho máximo de LOB que o DMS deve aceitar. Isso permite que o DMS pré-aloque memória e carregue os dados de LOB em massa. Os LOBs que ultrapassam o tamanho máximo são truncados, e é enviado um aviso para o arquivo de log. No modo LOB limitado, há ganhos significativos de desempenho em relação ao modo LOB completo. É recomendável utiliza o modo LOB limitado sempre que possível. O valor máximo recomendado é 102400 KB (100 MB).

    > **Note**
    >
    > Utilizar a opção Tamanho máximo de LOB (K) com um valor maior que 63 KB afeta o desempenho de uma carga máxima configurada para ser executada no modo LOB limitado. Durante uma carga máxima, o DMS aloca memória multiplicando o valor do

tamanho máximo de LOB (k) pela taxa de confirmação, e o produto é multiplicado pelo número de colunas de LOB. Quando o DMS não pode pré-alocar essa memória, o DMS começa a consumir memória de SWAP, e isso afeta o desempenho de uma carga máxima. Portanto, se você tiver problemas de desempenho ao utilizar o modo LOB limitado, considere diminuir a taxa de confirmação até atingir um nível aceitável de desempenho. Também é possível considerar a utilização do modo LOB em linha para endpoints compatíveis depois de compreender a distribuição de LOB para a tabela. Para validar o tamanho de LOB limitado, defina `ValidationPartialLobSize` com o mesmo valor de `LobMaxSize` (K).

- Modo LOB em linha: no modo LOB em linha, defina o tamanho máximo de LOB que o DMS transfere em linha. LOBs menores que o tamanho especificado são transferidos em linha. LOBs maiores que o tamanho especificado são replicados utilizando o modo LOB completo. É possível selecionar essa opção para replicar LOBs pequenos e grandes quando a maioria dos LOBs for pequena. O DMS não é compatível com o modo LOB em linha para endpoints que não são compatíveis com o modo LOB completo, como S3 e Redshift.

> **Note**
>
> Com o Oracle, os LOBs são tratados como tipos de dados VARCHAR sempre que possível. Essa abordagem significa que os AWS DMS obtém do banco de dados em massa, o que é significativamente mais rápido do que outros métodos. O tamanho máximo de um VARCHAR no Oracle é 32 K. Portanto, um tamanho de LOB limitado de menos de 32 K é ideal quando o Oracle é o banco de dados de origem.

- Quando uma tarefa é configurada para ser executada no modo LOB limitado, a opção Max LOB size (K) (Tamanho máx. de LOB (K) define o tamanho máximo de LOB compatível com o AWS DMS . Os LOBs maiores do que esse valor são truncados nesse valor.
- Quando uma tarefa é configurada para usar o modo LOB completo, AWS DMS recupera LOBs em partes. A opção Tamanho do bloco de LOB (K) (Tamanho do bloco de LOB (K) determina o tamanho de cada parte. Ao configurá-la, preste bastante atenção ao tamanho máximo de pacote permitido pela sua configuração de rede. Se o tamanho do bloco de LOB exceder o tamanho máximo de pacote permitido, talvez você veja erros de desconexão. O valor recomendado para `LobChunkSize` é 64 kilobytes. Aumentar o valor de `LobChunkSize` acima de 64 kilobytes pode causar falhas na tarefa.
- Quando uma tarefa é configurada para ser executada no modo LOB em linha, a `InlineLobMaxSize` configuração determina quais LOBs o DMS transfere em linha.

> **Note**
>
> É possível utilizar tipos de dados de LOB somente com tabelas e visualizações que incluem uma chave primária.

Para obter informações sobre as configurações de tarefas para especificar essas opções, consulte Configurações de tarefa de metadados de destino

Criar várias tarefas

Em alguns cenários de migração, talvez seja necessário criar várias tarefas de migração. As tarefas funcionam de forma independente e podem ser executados simultaneamente. Cada tarefa tem seu próprio carregamento inicial, CDC e processo de leitura de log. As tabelas relacionadas à linguagem de manipulação de dados (DML) devem fazer parte da mesma tarefa.○

Veja a seguir alguns motivos para criar várias tarefas para uma migração:

- As tabelas de destino das tarefas residem em bancos de dados diferentes, como quando você realiza fan-out ou divide um sistema em vários.
- Você deseja dividir a migração de uma tabela grande em várias tarefas utilizando filtragem.

> **Note**
>
> Como cada tarefa tem seu próprio processo de leitura de log e de captura de alterações, as alterações não são coordenadas entre elas. Portanto, ao utilizar várias tarefas para executar uma migração, verifique se cada transação de origem está totalmente contida em uma única tarefa. É possível utilizar várias tarefas para executar uma migração se nenhuma transação individual for dividida em tarefas diferentes.

Criar tarefas para replicação contínua utilizando o AWS DMS

É possível criar uma tarefa do AWS DMS que capture as alterações em andamento no armazenamento de origem. É possível fazer essa captura enquanto migra seus dados. Também é possível criar uma tarefa que capture alterações em andamento depois de concluir a migração

inicial (carga máxima) para um armazenamento de dados de destino compatível. Esse processo é chamado de replicação contínua ou captura de dados de alteração (CDC). O AWS DMS utiliza esse processo ao replicar alterações em andamento de um armazenamento de dados de origem. Esse processo funciona coletando as alterações nos logs de banco de dados utilizando a API nativa do mecanismo de banco de dados.

> **Note**
>
> É possível migrar visualizações utilizando apenas tarefas de carga máxima. Se a tarefa for somente CDC ou uma tarefa de carga máxima que inicie a CDC após a conclusão, a migração incluirá apenas tabelas da origem. Usando uma tarefa somente de carga máxima, é possível migrar exibições ou uma combinação de tabelas e exibições. Para obter mais informações, consulte Especificar a seleção de tabelas e as regras de transformação utilizando JSON.

Cada mecanismo de origem tem requisitos de configuração específicos para expor esse fluxo de alterações a uma conta de usuário específica. A maioria dos mecanismos exige algumas configurações adicionais para possibilitar que o processo de captura consuma os dados de alteração de forma significativa, sem perda de dados. Por exemplo, a Oracle exige a adição do registro em log suplementar e o MySQL exige o registro em log binário em a nível de linha (log binário).

Para fazer a leitura das alterações em andamento no banco de dados de origem, o AWS DMS utiliza ações da API específicas do mecanismo para ler alterações dos logs de transações no mecanismo de origem. Veja a seguir alguns exemplos de como o AWS DMS faz isso:

- Para o Oracle, o AWS DMS utiliza a API Oracle LogMiner ou a API do leitor binário (API bfile) para ler alterações em andamento. O AWS DMS faz a leitura de alterações em andamento a partir dos logs online ou redo de arquivamento com base no número de alterações do sistema (SCN).
- Para o Microsoft SQL Server, o AWS DMS utiliza MS-Replication ou MS-CDC para gravar informações no log de transações do SQL Server. Ele utiliza o perfil `fn_dblog()` ou `fn_dump_dblog()` no SQL Server para ler as alterações no log de transações com base no número de sequência do log (LSN).
- Para o MySQL, o AWS DMS lê as alterações dos logs binários (binlogs) baseados em linhas e migra essas alterações para o destino.
- Para o PostgreSQL, o AWS DMS define slots de replicação lógica e utiliza o plugin test_decoding para ler as alterações da origem e migrá-las para o destino.

- Para o Amazon RDS como origem, é recomendável garantir que os backups estejam ativados para configurar a CDC. Também é recomendável garantir que o banco de dados de origem esteja configurado para reter os logs de alterações por um tempo suficiente, 24 horas geralmente é suficiente. Para obter as configurações específicas de cada endpoint, consulte o seguinte:
  - Amazon RDS para Oracle: Configurando uma fonte Oracle AWS gerenciada para AWS DMS
  - Amazon RDS para MySQL e Aurora MySQL: Usando um banco AWS de dados compatível com MySQL gerenciado como fonte para AWS DMS.
  - Amazon RDS para SQL Server: Configurar a replicação contínua em uma instância de banco de dados SQL Server na nuvem
  - Amazon RDS para PostgreSQL e Aurora PostgreSQL: o PostgreSQL mantém automaticamente o WAL necessário.

Existem dois tipos de tarefas de replicação contínua:

- Carga máxima mais CDC: a tarefa migra os dados existentes e atualiza o banco de dados de destino com base nas alterações do banco de dados de origem.
- Somente CDC: a tarefa migra alterações em andamento depois que os dados estão no banco de dados de destino.

Executar a replicação a partir de um ponto de início de CDC

É possível iniciar uma tarefa de replicação contínua do AWS DMS (somente para captura de dados de alteração) de diversos pontos. Incluindo o seguinte:

- Em um horário de início de CDC personalizado: é possível utilizar a AWS CLI ou o AWS DMS com um timestamp em que você deseja que a replicação seja iniciada. O AWS DMS iniciará uma tarefa de replicação contínua a partir desse horário de início da CDC personalizada. O AWS DMS converterá o timestamp fornecido (em UTC) para um ponto de início nativo, como um LSN para SQL Server ou um SCN para Oracle. O AWS DMS usará métodos específicos do mecanismo para determinar onde iniciar a tarefa de migração com base no fluxo de alterações do mecanismo de origem.

> **Note**
>
> Somente definindo o atributo de conexão `StartFromContext` com o timestamp necessário, o Db2 como origem oferece um horário de início personalizado da CDC.

O PostgreSQL como origem não é compatível com um horário de início personalizado da CDC. Isso ocorre porque o mecanismo de banco de dados do PostgreSQL não consegue mapear um timestamp para um LSN ou SCN, como fazem o Oracle e o SQL Server.

- Em um ponto de início nativo de CDC: também é possível iniciar em um ponto nativo no log de transações do mecanismo de origem. Em alguns casos, é possível preferir essa abordagem porque um timestamp pode indicar vários pontos nativos no log de transações. O AWS DMS é compatível com esse recurso para os seguintes endpoints de origem:
  - SQL Server
  - PostgreSQL
  - Oracle
  - MySQL
  - MariaDB

Quando a tarefa é criada, o AWS DMS marca o ponto de início da CDC e não pode ser alterada. Para utilizar um ponto de início diferente da CDC, crie uma nova tarefa.

Determinar um ponto de início nativo de CDC

Um ponto de início nativo de CDC é um ponto no log do mecanismo do banco de dados que define um horário para começar a CDC. Como um exemplo, suponha que um despejo de dados em massa tenha sido aplicado ao destino. É possível pesquisar o ponto de início nativo para a tarefa somente de replicação contínua. Para evitar qualquer inconsistência de dados, escolha cuidadosamente o ponto de início da tarefa somente de replicação. O DMS captura transações iniciadas após o ponto de início da CDC escolhido.

Os exemplos a seguir mostram como encontrar o ponto de início nativo da CDC em mecanismos de origem compatíveis:

SQL Server

No SQL Server, o número de sequência de log (LSN) tem três partes:

- Número de sequência do Arquivo de log virtual (VLF)
- Deslocamento inicial de um bloco de log
- Número do slot

Um exemplo de LSN é o seguinte: `00000014:00000061:0001`

Para obter o ponto de início para uma tarefa de migração do SQL Server com base nas configurações de backup do log de transações, utiliza o perfil `fn_dblog()` ou `fn_dump_dblog()` no SQL Server.

Para utilizar o ponto de início nativo da CDC com o SQL Server, crie uma publicação em qualquer tabela que participe da replicação contínua. O AWS DMS cria a publicação automaticamente quando você utiliza a CDC sem utilizar um ponto de início nativo da CDC.

PostgreSQL

Utilize um ponto de verificação de recuperação de CDC para o banco de dados de origem PostgreSQL. Esse valor de ponto de verificação é gerado em vários pontos à medida que uma tarefa de replicação contínua é executada para o banco de dados de origem (a tarefa pai). Para obter mais informações sobre pontos de verificação em geral, consulte Utilizar um ponto de verificação como um ponto de início da CDC.

Para identificar o ponto de verificação a ser utilizado como ponto de início nativo, utilize a visualização `pg_replication_slots` do banco de dados ou os detalhes da visão geral da tarefa pai no AWS Management Console.

Como encontrar os detalhes da visão geral da tarefa pai no console

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.

   Se estiver conectado como usuário do IAM, verifique se você tem as permissões necessárias para acessar o AWS DMS. Para obter mais informações sobre as permissões necessárias, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, escolha Tarefas de migração do banco de dados.
3. Escolha a tarefa pai na lista na página Database migration tasks (Tarefas de migração de banco de dados). Isso abre a página da tarefa pai mostrando os detalhes da visão geral.
4. Encontre o valor do ponto de verificação em Captura de dados de alteração (CDC), Posição inicial da captura de dados de alteração (CDC) e Ponto de verificação de recuperação da captura de dados de alteração (CDC).

   O valor é semelhante ao valor a seguir:

   ```
   checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0
   ```

Aqui, o componente `4AF/B00000D0` é o que você precisa para especificar esse ponto de início nativo da CDC. Defina o parâmetro `CdcStartPosition` da API do DMS como esse valor ao criar a tarefa de CDC para iniciar a replicação neste ponto de início para a origem do PostgreSQL. Para obter informações sobre como utilizar a AWS CLI para criar a tarefa de CDC, consulte Habilitando o CDC com uma instância de banco AWS de dados PostgreSQL gerenciada com AWS DMS.

Oracle

Um número de alterações do sistema (SCN) é um timestamp interno e lógico utilizado por bancos de dados Oracle. Os SCNs ordenam os eventos que ocorrem dentro do banco de dados, o que é necessário para satisfazer as propriedades ACID de uma transação. Os bancos de dados Oracle utilizam SCNs para marcar o local em que todas as alterações foram gravadas em disco para que uma ação de recuperação não seja aplicada nas alterações já gravadas. A Oracle também utiliza SCNs para marcar o ponto em que não existem redos para um conjunto de dados, para que a recuperação possa ser interrompida.

Para obter o SCN atual em um banco de dados Oracle, execute o seguinte comando.

```
SELECT CURRENT_SCN FROM V$DATABASE
```

Se você utilizar o SCN ou o timestamp para iniciar uma tarefa de CDC, perderá os resultados de qualquer transação aberta e haverá falha na migração. Transações abertas são transações que foram iniciadas antes da posição de início da tarefa e confirmadas após a posição de início da tarefa. É possível identificar o SCN e o timestamp para iniciar uma tarefa de CDC em um ponto que inclua todas as transações abertas. Para obter mais informações, consulte Transações na documentação on-line do Oracle. Com a versão 3.5.1 e superior, o AWS DMS é compatível com transações abertas para uma tarefa somente de CDC utilizando a configuração `openTransactionWindow` do endpoint se você utilizar o SCN ou o timestamp para iniciar a tarefa.

Ao utilizar a configuração `openTransactionWindow`, forneça a janela, em número de minutos, para tratar as transações abertas. O AWS DMS muda a posição de captura e encontra a nova posição para iniciar a captura de dados. O AWS DMS utiliza a nova posição de início para verificar todas as transações abertas dos redos necessários do Oracle ou dos redo logs arquivados.

MySQL

Antes do lançamento do MySQL versão 5.6.3, o número de sequência de log (LSN) para MySQL era um inteiro não assinado de 4 bytes. No MySQL versão 5.6.3, quando o limite de tamanho de arquivo de log redo aumentou de 4 GB para 512 GB, o LSN se tornou um inteiro não assinado de 8 bytes. O aumento reflete que bytes adicionais foram necessários para armazenar informações de tamanho extra. As aplicações criadas no MySQL 5.6.3 ou superior que utilizam valores de LSN devem utilizar variáveis de 64 bits em vez de 32 bits para armazenar e comparar valores de LSN. Para obter mais informações LSNs do MySQL, consulte a Documentação do MySQL.

Para obter o LSN atual em um banco de dados MySQL, execute o seguinte comando.

```
mysql> show master status;
```

A consulta retorna o nome e a posição do arquivo de log binário, além de vários outros valores. O ponto de início nativo de CDC é uma combinação do nome e da posição do arquivo de log binário, por exemplo, `mysql-bin-changelog.000024:373`. Nesse exemplo, `mysql-bin-changelog.000024` é o nome do arquivo de log binário e `373` é a posição onde o AWS DMS precisa iniciar a captura de alterações.

Utilizar um ponto de verificação como um ponto de início da CDC

Uma tarefa de replicação contínua migrará as alterações, e o AWS DMS armazenará em cache as informações sobre ponto de verificação específicos do AWS DMS, periodicamente. O ponto de verificação criado pelo AWS DMS contém informações para que o mecanismo de replicação conheça o ponto de recuperação para o fluxo de alterações. É possível utilizar o ponto de verificação para voltar no cronograma de alterações e recuperar uma tarefa de migração com falha. Também é possível utilizar um ponto de verificação para iniciar outra tarefa de replicação contínua para outro destino em qualquer ponto determinado no tempo.

É possível obter as informações do ponto de verificação de uma das seguintes formas:

- Execute a operação da API `DescribeReplicationTasks` e visualize os resultados. É possível filtrar as informações por tarefa e pesquisar o ponto de verificação. É possível recuperar o último ponto de verificação quando a tarefa está no estado interrompida ou com falha. Essas informações serão perdidas se a tarefa for excluída.
- Visualize a tabela de metadados chamada `awsdms_txn_state` na instância de destino. É possível consultar a tabela para obter informações do ponto de verificação. Para criar a tabela de

metadados, defina o parâmetro `TaskRecoveryTableEnabled` como `Yes` ao criar uma tarefa. Essa configuração faz com que o AWS DMS grave continuamente as informações do ponto de verificação na tabela de metadados de destino. Essas informações serão perdidas se uma tarefa for excluída.

Por exemplo, o seguinte é uma verificação de exemplo na tabela de metadados: `checkpoint:V1#34#00000132/0F000E48#0#0#*#0#121`

- No painel de navegação, escolha Tarefas de migração de banco de dados e escolha a tarefa pai na lista que aparece na página Tarefas de migração de banco de dados. A página da tarefa pai é aberta, mostrando os detalhes da visão geral. Encontre o valor do ponto de verificação em Captura de dados de alteração (CDC), Posição inicial da captura de dados de alteração (CDC) e Ponto de verificação de recuperação da captura de dados de alteração (CDC). O valor do ponto de verificação é semelhante ao seguinte:

  `checkpoint:V1#1#000004AF/B00000D0#0#0#*#0#0`

Interromper uma tarefa em um ponto de tempo de confirmação ou servidor

Com a introdução dos pontos de início nativos de CDC, o AWS DMS também pode interromper uma tarefa nos seguintes pontos:

- Um tempo de confirmação na origem
- Um tempo do servidor na instância de replicação

É possível modificar uma tarefa e definir uma hora em UTC para interrompê-la, conforme necessário. A tarefa é interrompida automaticamente com base no tempo da confirmação ou do servidor definido. Ou, se você souber um tempo adequada para interromper a tarefa de migração durante a criação da tarefa, defina também um tempo de interrupção ao criar a tarefa.

> **Note**
>
> Pode demorar até 40 minutos para inicializar todos os recursos na primeira vez que você inicia uma nova replicação do AWS DMS com Tecnologia Sem Servidor. Observe que a opção `server_time` só é aplicável após a conclusão da inicialização do recurso.

Executar replicação bidirecional

É possível utilizar as tarefas do AWS DMS para executar a replicação bidirecional entre dois sistemas. Na replicação bidirecional, replique os dados da mesma tabela (ou conjunto de tabelas) entre dois sistemas nas duas direções.

Por exemplo, é possível copiar uma tabela EMPLOYEE do banco de dados A para o banco de dados B e replicar as alterações na tabela do banco de dados A para o banco de dados B. Também é possível replicar alterações na tabela EMPLOYEE do banco de dados B de volta para A. Assim, você estará executando a replicação bidirecional.

> **Note**
>
> A replicação bidirecional do AWS DMS não pretende ser uma solução completa de vários mestres, incluindo nó primário, resolução de conflitos e assim por diante.

Utilize a replicação bidirecional para situações em que os dados em diferentes nós são segregados operacionalmente. Em outras palavras, vamos supor que você tenha um elemento de dados alterado por um aplicativo operando no nó A e que o nó A execute a replicação bidirecional com o nó B. Esse elemento de dados no nó A nunca será alterado por nenhum aplicativo que opere no nó B.

O AWS DMS é compatível com a replicação bidirecional nos seguintes mecanismos de banco de dados:

- Oracle
- SQL Server
- MySQL
- PostgreSQL
- Amazon Aurora Edição compatível com MySQL
- Aurora Edição Compatível com PostgreSQL

Criar tarefas de replicação bidirecional

Para habilitar a replicação bidirecional do AWS DMS, configure os endpoints de origem e de destino para os dois bancos de dados (A e B). Por exemplo, configure um endpoint de origem para o banco

de dados A, um endpoint de origem para o banco de dados B, um endpoint de destino para o banco de dados A e um endpoint de destino para o banco de dados B.

Crie, então, duas tarefas: uma tarefa para a origem A mover dados para o destino B e outra para a origem B mover dados para o destino A. Além disso, verifique se cada tarefa está configurada com prevenção de loopback. Fazer isso impede que alterações idênticas sejam aplicadas aos destinos das duas tarefas, corrompendo os dados de, pelo menos, uma delas. Para obter mais informações, consulte Evitar loopback.

Para obter a abordagem mais fácil, comece com conjuntos de dados idênticos no banco de dados A e B. Crie duas tarefas somente CDC, uma tarefa para replicar dados de A para B e outra tarefa para replicar dados de B para A.

Para utilizar o AWS DMS para instanciar um novo conjunto de dados (banco de dados) no nó B, do nó A, faça o seguinte:

1. Utilize uma tarefa de carga máxima e CDC para mover dados do banco de dados A para B. Verifique se nenhuma aplicação esteja modificando dados no banco de dados B durante esse período.
2. Quando a carga máxima estiver concluída e antes que as aplicações possam modificar os dados no banco de dados B, observe o horário ou a posição de início da CDC do banco de dados B. Para obter instruções, consulte Executar a replicação a partir de um ponto de início de CDC.
3. Crie uma tarefa somente CDC que mova dados do banco de dados B de volta para A utilizando esse horário de início ou posição inicial de CDC.

> **Note**
>
> Apenas uma tarefa em um par bidirecional pode ser tanto carga máxima quanto CDC.

Evitar loopback

Para exibir o loopback de prevenção, vamos supor que em uma tarefa T1 o AWS DMS leia logs de alteração do banco de dados de origem A e aplique as alterações no banco de dados de destino B.

Uma segunda tarefa, T2, lê logs de alterações do banco de dados de origem B e aplica-as de volta no banco de dados de destino A. Antes que a T2 faça isso, o DMS deve verificar se as mesmas alterações feitas no banco de dados de destino B a partir do banco de dados de origem A não sejam

feitas no banco de dados de origem A. Em outras palavras, o DMS deve verificar se essas alterações não são ecoadas (looped) de volta para o banco de dados de destino A. Caso contrário, os dados no banco de dados A poderão ser corrompidos.

Para evitar o loopback de alterações, adicione as seguintes configurações de tarefa a cada tarefa de replicação bidirecional. Isso garante que os dados de loopback não sejam corrompidos em nenhuma das direções.

```
{
. . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": Boolean,
    "SourceSchema": String,
    "TargetSchema": String
  },

. . .
}
```

As configurações de tarefa `LoopbackPreventionSettings` determinam se uma transação é nova ou um eco da tarefa de replicação oposta. Quando o AWS DMS aplica uma transação a um banco de dados de destino, ele atualiza uma tabela do DMS (`awsdms_loopback_prevention`) com uma indicação da alteração. Antes de aplicar cada transação a um destino, o DMS ignora qualquer transação que inclua referência a essa tabela `awsdms_loopback_prevention`. Portanto, ele não aplica a alteração.

Inclua essas configurações de tarefa com cada tarefa de replicação em um par bidirecional. Essas configurações habilitam a prevenção de loopback. Elas também especificam o esquema para cada banco de dados de origem e destino na tarefa que inclui a tabela `awsdms_loopback_prevention` para cada endpoint.

Para permitir que cada tarefa identifique esse eco e o descarte, defina `EnableLoopbackPrevention` como `true`. Para especificar um esquema na origem que inclua `awsdms_loopback_prevention`, defina `SourceSchema` para o nome desse esquema no banco de dados de origem. Para especificar um esquema no destino que inclua a mesma tabela, defina `TargetSchema` para o nome desse esquema no banco de dados de destino.

No exemplo a seguir, as configurações `SourceSchema` e `TargetSchema` para uma tarefa de replicação T1 e sua tarefa de replicação oposta T2 são especificadas com configurações opostas.

As configurações para a tarefa T1 são as seguintes.

```
{
. . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "LOOP-DATA",
    "TargetSchema": "loop-data"
  },

. . .
}
```

As configurações para a tarefa oposta T2 são as seguintes.

```
{
. . .

  "LoopbackPreventionSettings": {
    "EnableLoopbackPrevention": true,
    "SourceSchema": "loop-data",
    "TargetSchema": "LOOP-DATA"
  },

. . .
}
```

> **Note**
>
> Ao utilizar a AWS CLI, utilize somente os comandos `create-replication-task` ou `modify-replication-task` para configurar `LoopbackPreventionSettings` em suas tarefas de replicações bidirecionais.

Limitações da replicação bidirecional

A replicação bidirecional para AWS DMS tem as seguintes limitações:

- A prevenção de loopback rastreia apenas instruções de linguagem de manipulação de dados (DML). O AWS DMS não é compatível com a prevenção de loopback da linguagem de definição

de dados (DDL). Para fazer isso, configure uma das tarefas em um par bidirecional para filtrar instruções DDL.

- As tarefas que utilizam prevenção de loopback não são compatíveis com a confirmação de alterações em lote. Para configurar uma tarefa com prevenção de loopback, defina `BatchApplyEnabled` como `false`.
- A replicação bidirecional do DMS não inclui detecção ou resolução de conflitos. Para detectar inconsistências de dados, utilize a validação de dados nas duas tarefas.

Modificar uma tarefa

É possível modificar as configurações de uma tarefa, o mapeamento de tabela e outras configurações. Também é possível ativar e executar avaliações de pré-migração antes de executar a tarefa modificada. É possível modificar uma tarefa no console, selecionando a tarefa e escolhendo Modificar. Também é possível usar o comando da CLI ou a operação da API ModifyReplicationTask.

Há poucas limitações para modificar uma tarefa. Incluindo o seguinte:

- Não é possível modificar o endpoint de origem ou de destino de uma tarefa.
- Você não pode alterar o tipo de migração de uma tarefa.
- As tarefas executadas devem ter o status Interrompida ou Com falha para serem modificadas.

Mover uma tarefa

É possível mover uma tarefa para uma instância de replicação diferente quando qualquer uma das situações a seguir se aplicar ao seu caso de uso.

- No momento, você está utilizando uma instância de um determinado tipo e quer mudar para um tipo de instância diferente.
- A instância atual está sobrecarregada por muitas tarefas de replicação, e você quer dividir a carga em várias instâncias.
- O armazenamento da instância está cheio, e você quer mover as tarefas dessa instância para uma instância com mais capacidade como alternativa ao ajuste da escala do armazenamento ou da computação.

- Você quer usar um recurso recém-lançado do AWS DMS, mas não quer criar uma nova tarefa e reiniciar a migração. Em vez disso, você prefere criar uma instância de replicação com uma nova versão do AWS DMS compatível com o recurso, e mover a tarefa existente para essa instância.

É possível mover uma tarefa no console, selecionando a tarefa e escolhendo Mover. Também é possível utilizar o comando da CLI ou a operação da API `MoveReplicationTask` para mover a tarefa. É possível mover uma tarefa que tenha qualquer mecanismo de banco de dados como o endpoint de destino.

Verifique se a instância de replicação de destino tem espaço de armazenamento suficiente para acomodar a tarefa que está sendo movida. Caso contrário, ajuste a escala do armazenamento para liberar espaço para a instância de replicação de destino antes de mover a tarefa.

Além disso, verifique se a instância de replicação de destino é criada com a mesma versão ou superior do mecanismo do AWS DMS da instância de replicação atual.

> **Note**
>
> - Não é possível mover uma tarefa para a mesma instância de replicação em que ela reside atualmente.
> - Não é possível modificar as configurações de uma tarefa enquanto ela está sendo movida.
> - Uma tarefa executada deve ter um status de Interrompida, Com falha ou Falha ao mover antes que você possa movê-la.

Há dois status de tarefa relacionados à movimentação de uma tarefa do DMS, Movendo e Falha ao mover. Para obter mais informações sobre esses status de tarefas, consulte Status da tarefa.

Depois de mover uma tarefa, é possível ativar e executar avaliações de pré-migração para verificar se há problemas de bloqueio antes de executar a tarefa movida.

Recarregar tabelas durante uma tarefa

Enquanto uma tarefa é executada, é possível recarregar uma tabela de banco de dados de destino utilizando os dados da origem. Se, durante a tarefa, ocorrer um erro ou forem alterados dados devido a operações de partição, você poderá recarregar uma tabela (por exemplo, utilizando o Oracle). É possível recarregar até 10 tabelas de uma tarefa.

O recarregamento de tabelas não interrompe a tarefa.

Para recarregar uma tabela, as seguintes condições devem ser verdadeiras:

- A tarefa deve estar em execução.
- O método de migração da tarefa deve ser de carregamento completo ou de carregamento completo com CDC.
- Não são permitidas tabelas duplicadas.
- O AWS DMS mantém a definição da tabela lida anteriormente e não a recria durante a operação de recarregamento. As instruções de DDL, como ALTER TABLE ADD COLUMN ou DROP COLUMN, feitas na tabela antes de ela ser recarregada, podem causar uma falha na operação de recarregamento.

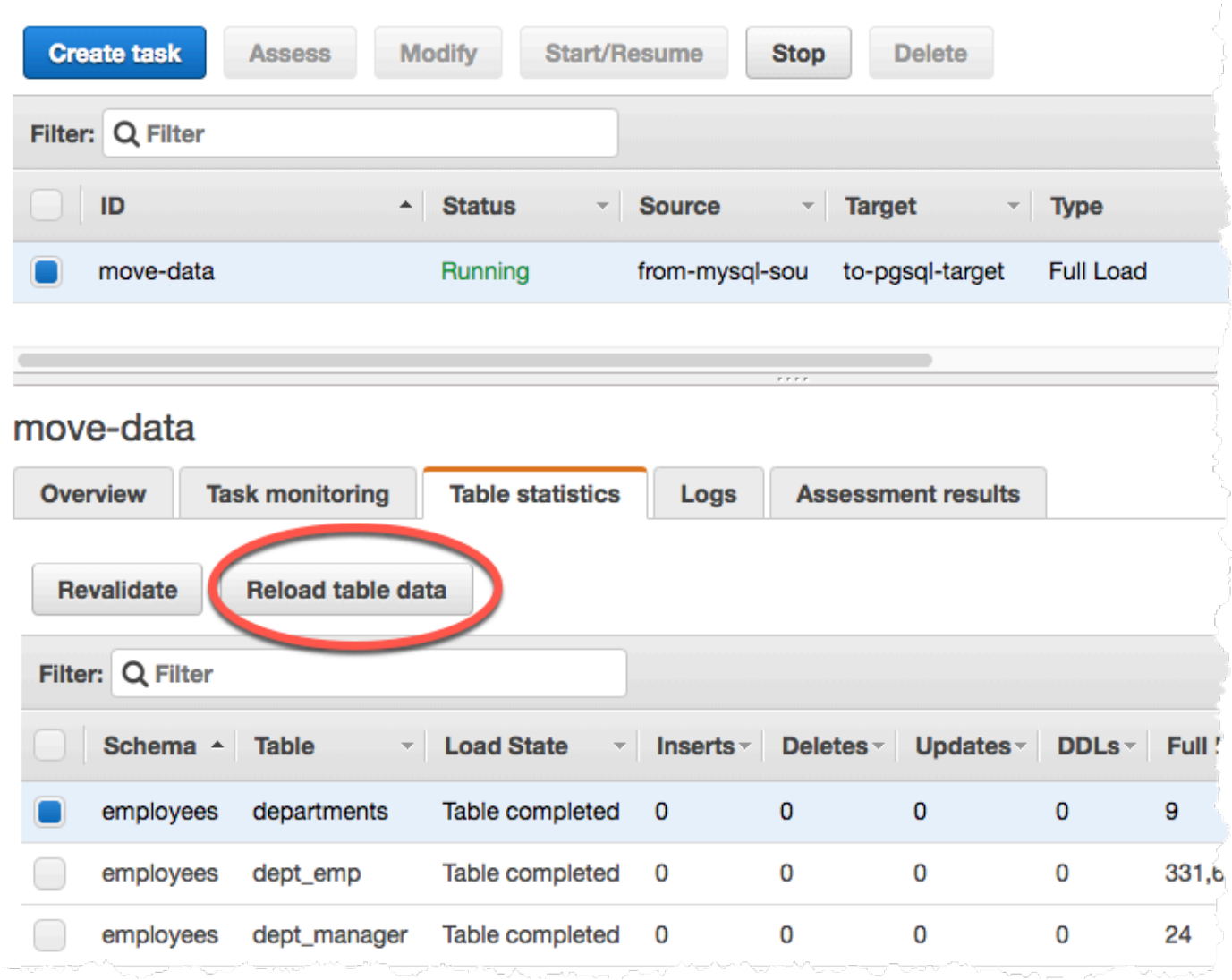
> **Note**
>
> O DMS aplica a configuração `TargetTablePrepMode` antes de recarregar a tabela. Se você definir `TargetTablePrepMode` como `DO_NOTHING`, primeiro deverá truncar manualmente a tabela.

AWS Management Console

Como recarregar uma tabela utilizando o console do AWS DMS

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.

   Se estiver conectado como usuário do IAM, verifique se você tem as permissões necessárias para acessar o AWS DMS. Para obter mais informações sobre as permissões necessárias, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, selecione Tasks.
3. Escolha a tarefa em execução que tem a tabela que você deseja recarregar.
4. Escolha a guia Table Statistics (Estatísticas da tabela).



5. Escolha a tabela que você deseja recarregar. Se a tarefa não está mais em execução, você não pode recarregar a tabela.
6. Selecione Reload table data (Recarregar dados da tabela).

Quando o AWS DMS está se preparando para recarregar uma tabela, o console altera o status dela para Table is being reloaded (A tabela está sendo recarregada).

Utilizar o mapeamento de tabela para especificar as configurações da tarefa

O mapeamento de tabela utiliza vários tipos de regras para especificar a fonte de dados, o esquema de origem, os dados e todas transformações que devem ocorrer durante a tarefa. É possível utilizar

o mapeamento de tabela para especificar as tabelas individuais em um banco de dados a serem migradas e o esquema a ser utilizado na migração.

Ao trabalhar com mapeamento de tabela, é possível utilizar filtros para especificar os dados que deseja replicar de colunas de tabela. Além disso, é possível utilizar transformações para modificar esquemas, tabelas ou visualizações selecionadas antes que sejam gravadas no banco de dados de destino.

Tópicos

> **Note**
>
> Ao trabalhar com mapeamento de tabela para um endpoint de origem do MongoDB, é possível utilizar filtros para especificar os dados que deseja replicar e especificar um nome de banco de dados no lugar do `schema_name`. Ou, é possível utilizar o valor padrão `"%"`.

Especificar a seleção de tabelas e as regras de transformação no console

Você pode usar o AWS Management Console para realizar o mapeamento da tabela, incluindo a especificação da seleção e das transformações da tabela. No console, utilize a seção Onde para especificar o esquema, a tabela e a ação (incluir ou excluir). Utilize a seção Filtrar para especificar o nome da coluna em uma tabela e as condições que você deseja aplicar a uma tarefa de replicação. Juntas, essas duas ações criam uma regra de seleção.

As transformações podem ser incluídas no mapeamento de tabela após você especificar pelo menos uma regra de seleção. As transformações podem ser usadas para renomear um esquema ou uma tabela, adicionar um prefixo ou sufixo a um esquema ou uma tabela ou remover a coluna de uma tabela.

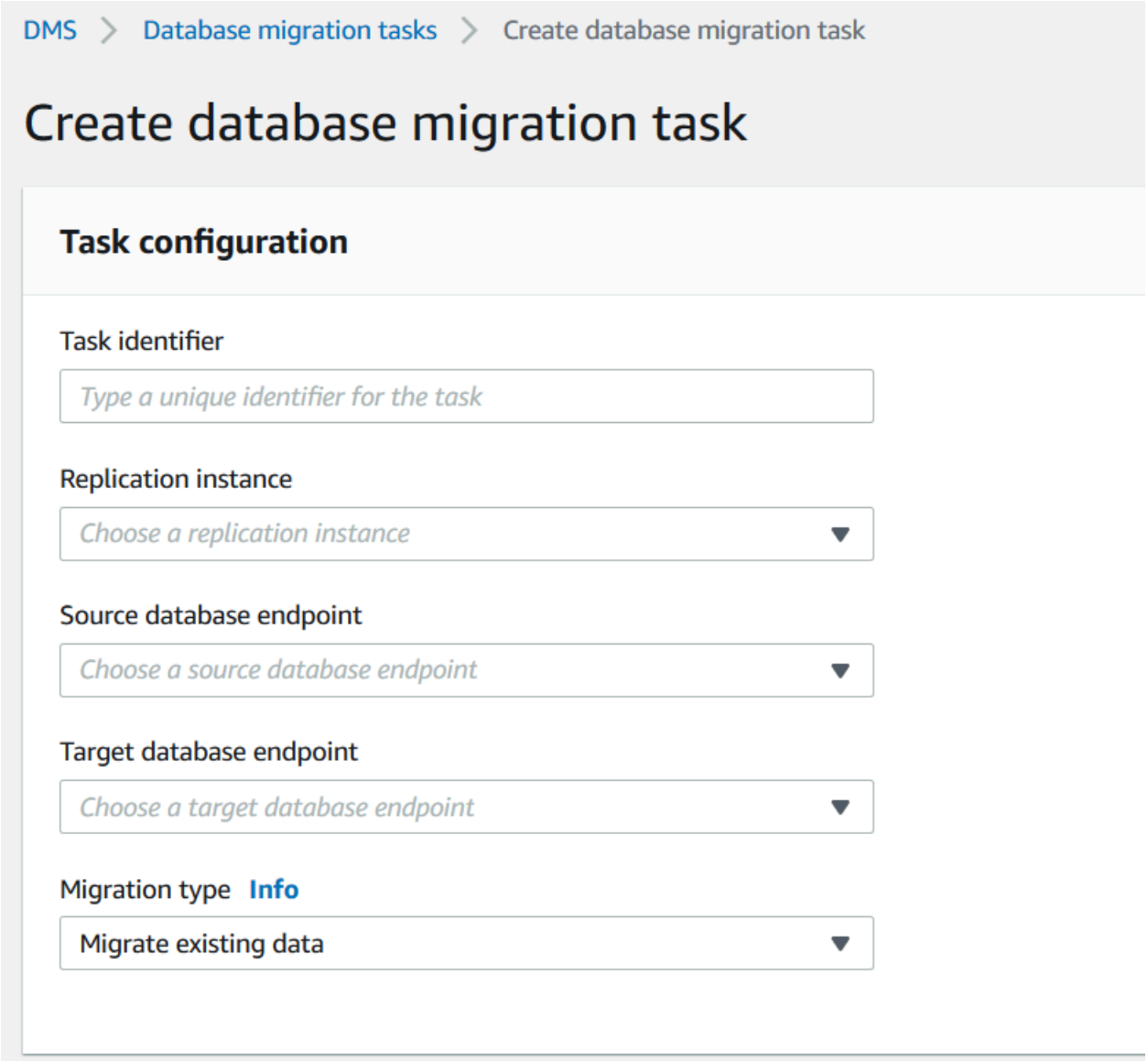
> **Note**
>
> AWS DMS não oferece suporte a mais de uma regra de transformação por nível de esquema, nível de tabela ou nível de coluna.

O procedimento a seguir mostra como configurar regras de seleção com base em uma tabela chamada **`Customers`** em um esquema chamado **`EntertainmentAgencySample`**.

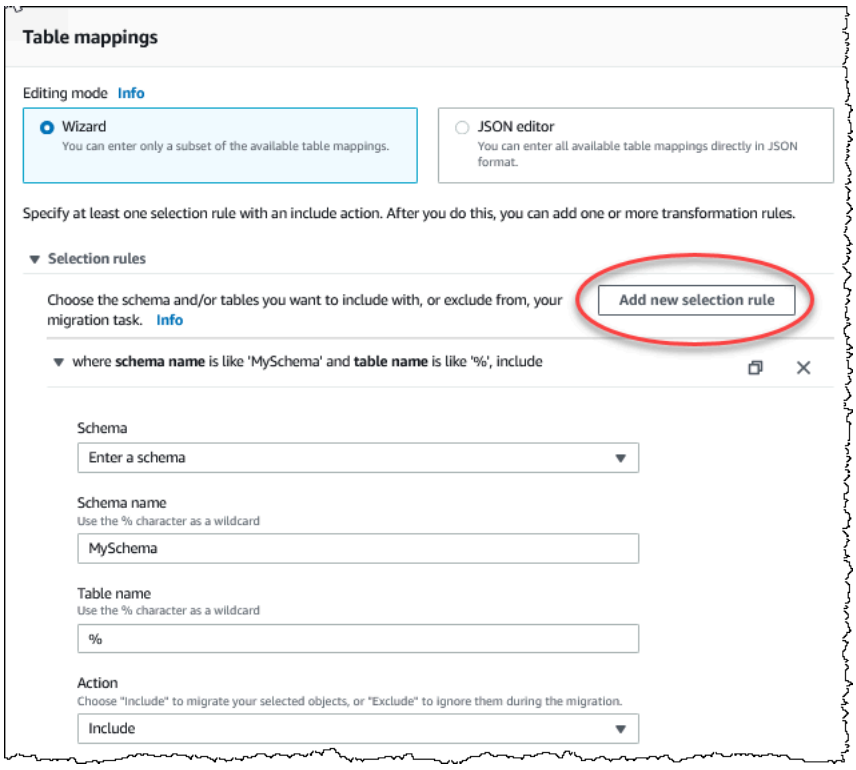
Como especificar uma seleção de tabela, critérios de filtro e transformações utilizando o console

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Se estiver conectado como usuário do IAM, verifique se você tem as permissões necessárias para acessar o AWS DMS. Para obter mais informações sobre as permissões necessárias, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. Na página Painel, escolha Tarefas de migração de banco de dados.
3. Escolha Create Task.
4. Na seção Configuração da tarefa, insira as informações da tarefa, incluindo o Identificador da tarefa, a Instância de replicação, o Endpoint do banco de dados de origem, o Endpoint do banco de dados de destino e o Tipo de migração.



5. Na seção Mapeamento de tabela, insira o nome do esquema ou o nome da tabela. É possível utilizar "%" como um valor curinga ao especificar o nome do esquema ou o nome da tabela. Para obter informações sobre outros curingas que é possível utilizar, consulte the section called “Curingas no mapeamento de tabela”. Especifique a ação a ser executada para incluir ou excluir os dados definidos pelo filtro.

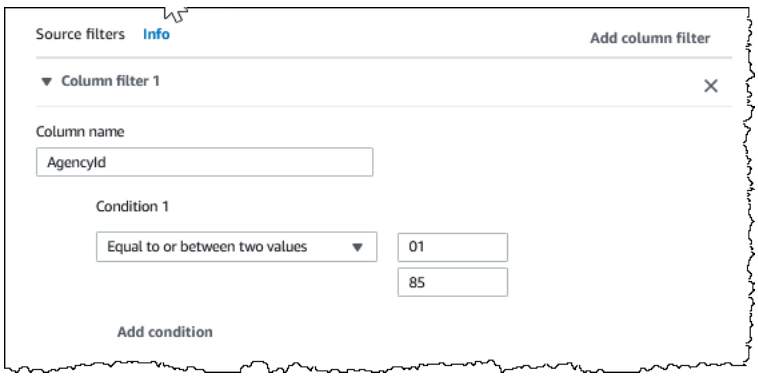


6. Especifique as informações de filtro utilizando os links Add column filter (Adicionar filtro de coluna) e Add condition (Adicionar condição).

   a. Selecione Add column filter (Adicionar filtro de coluna) para especificar uma coluna e as condições.

   b. Selecione Adicionar condition (Adicionar condição) para adicionar mais condições.

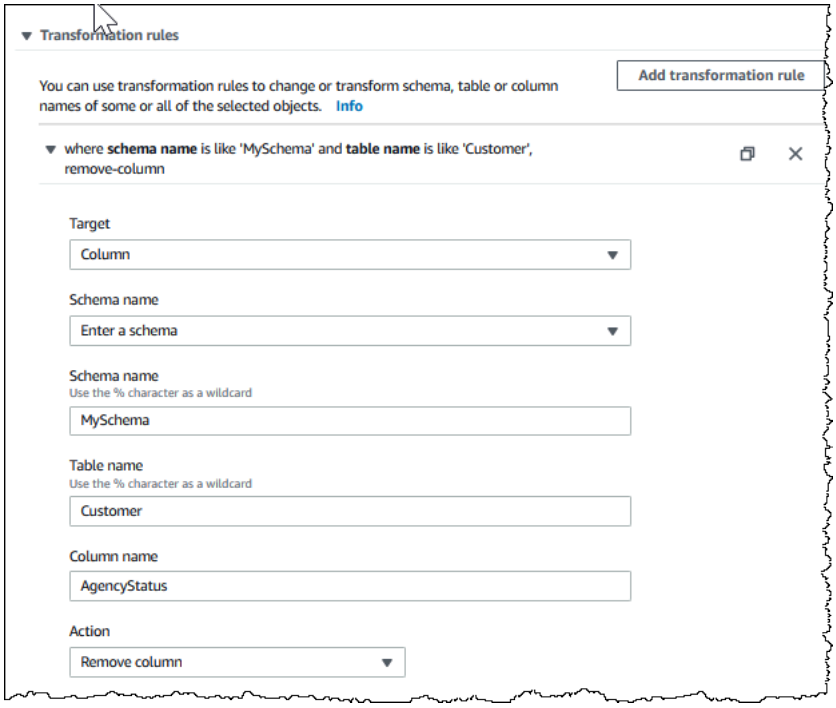
   O exemplo a seguir mostra um filtro para a tabela **`Customers`** que inclui **`AgencyIDs`** entre **`01`** e **`85`**.



7. Quando tiver criado as seleções desejadas, selecione Adicionar nova regra de seleção.

8. Depois de criar pelo menos uma regra de seleção, é possível adicionar uma transformação à tarefa. Selecione Add transformation rule (Adicionar regra de transformação).

9. Selecione o destino que deseja transformar e insira as informações adicionais solicitadas. O exemplo a seguir mostra uma transformação que exclui a coluna **`AgencyStatus`** da tabela **`Customer`**.



10. Selecione Add transformation rule.
11. Escolha Criar tarefa.

> **Note**
>
> AWS DMS não oferece suporte a mais de uma regra de transformação por nível de esquema ou por nível de tabela.

Especificar a seleção de tabelas e as regras de transformação utilizando JSON

Para especificar os mapeamentos de tabela que deseja aplicar durante a migração, é possível criar um arquivo JSON. Se você criar uma tarefa de migração utilizando o console, poderá procurar esse arquivo JSON ou inserir o JSON diretamente na caixa de mapeamento de tabela. Se utilizar a CLI ou a API para executar migrações, é possível especificar esse arquivo utilizando o parâmetro `TableMappings` da operação de API `CreateReplicationTask` ou `ModifyReplicationTask`.

AWS DMS só pode processar arquivos JSON de mapeamento de tabelas de até 2 MB. É recomendável manter o tamanho do arquivo JSON da regra de mapeamento abaixo do limite de 2 MB ao trabalhar com tarefas do DMS. Isso evita erros inesperados durante a criação ou a modificação da tarefa. Quando um arquivo de regra de mapeamento excede o limite de 2 MB, é recomendável dividir as tabelas em várias tarefas para reduzir o tamanho do arquivo de regras de mapeamento para que ele permaneça abaixo desse limite.

É possível especificar com quais tabelas, visualizações e esquemas você deseja trabalhar. Também é possível executar transformações de tabela, visualização e esquema e especificar configurações de como o AWS DMS carrega tabelas e visualizações individuais. Crie regras de mapeamento de tabela para essas opções utilizando os seguintes tipos de regra:

- Regras de `selection`: identificam os tipos e nomes das tabelas de origem, visualizações e esquemas a serem carregados. Para ter mais informações, consulte Regras de seleção e ações.
- Regras de `transformation`: especificam determinadas alterações ou adições a tabelas de origem e esquemas específicos na origem antes de serem carregados no destino. Para ter mais informações, consulte Regras de transformação e ações.

  Além disso, para definir o conteúdo de colunas novas e existentes, é possível utilizar uma expressão em uma regra de transformação. Para ter mais informações, consulte Utilizar expressões de regra de transformação para definir o conteúdo da coluna.
- Regras de `table-settings`: especificam como as tarefas do DMS carregam os dados de tabelas individuais. Para ter mais informações, consulte Regras e operações de configurações de tabelas e coleções.

> **Note**
>
> Para destinos do Amazon S3, também é possível marcar objetos do S3 mapeados para tabelas e esquemas selecionados utilizando o tipo de regra `post-processing` e a ação da regra `add-tag`. Para ter mais informações, consulte Marcação de objetos do Amazon S3. Para os destinos a seguir, é possível especificar como e onde os esquemas e tabelas selecionados são migrados para o destino utilizando o tipo de regra `object-mapping`:
>
> - Amazon DynamoDB: para obter mais informações, consulte Utilizar o mapeamento de objetos para migrar dados para o DynamoDB.
> - Amazon Kinesis: para obter mais informações, consulte Utilizar o mapeamento de objetos para migrar dados para um fluxo de dados do Kinesis.
> - Apache Kafka: para obter mais informações, consulte Utilizar o mapeamento de objetos para migrar dados para um tópico do Kafka.

Regras de seleção e ações

Usando o mapeamento de tabela, é possível especificar as tabelas, as exibições e os esquemas com os quais deseja trabalhar utilizando ações e regras de seleção. Os seguintes valores podem ser aplicados a regras de mapeamento de tabela que utilizam o tipo de regra de seleção.

Parâmetro	Possíveis valores	Descrição
`rule-type`	`selection`	Uma regra de seleção. Defina pelo menos uma regra de seleção ao especificar um mapeamento de tabelas.
`rule-id`	Um valor numérico.	Um valor numérico exclusivo para identificar a regra.
`rule-name`	Um valor alfanumérico.	Um nome exclusivo para identificar a regra.
`rule-action`	`include`, `exclude`, `explicit`	Um valor que inclui ou exclui o objeto ou objetos selecionados pela regra. Se `explicit` for especificado, será

Parâmetro	Possíveis valores	Descrição
		possível selecionar e incluir apenas um objeto que corresponda a uma tabela e a um esquema especific ados explicitamente.

Parâmetro	Possíveis valores	Descrição
`object-locator`	Um objeto com os seguintes parâmetros: • `schema-name` : o nome do esquema. • `table-name` : o nome da tabela. • (Opcional) `table-type` : `table | view | all`, para indicar se `table-name` se refere apenas a tabelas, visualizações ou ambas. O padrão é `table`. AWS DMS carrega visualizações somente em uma tarefa de carga completa. Se você tiver apenas tarefas de carga completa e captura de dados alterados (CDC), configure pelo menos uma full-load-only tarefa para carregar suas visualizações. Nem todos os endpoints de destino aceitam visualizações como fonte de replicação, mesmo com carga total (por exemplo, Amazon OpenSearch Service). Verifique as limitações do endpoint de destino.	O nome de cada esquema e tabela, ou exibição, aos quais a regra se aplica. Também é possível especificar se uma regra inclui apenas tabelas, somente visualizações ou ambas. Se `rule-action` for `include` ou `exclude`, será possível utilizar o sinal de porcentagem "%" como um curinga para todo ou parte do valor do parâmetro `schema-name` e `table-name` . Para obter informações sobre outros curingas que é possível utilizar, consulte the section called “Curingas no mapeamento de tabela”. Assim, você pode corresponder estes itens: • Uma única tabela, visualização ou coleção em um único esquema • Uma única tabela, visualização ou coleção em alguns ou todos os esquemas • Algumas ou todas as tabelas e visualizações em um único esquema ou coleções em um único banco de dados • Algumas ou todas as tabelas e visualizações em alguns ou todos os esquemas ou coleções em alguns ou todos os bancos de dados Se `rule-action` for `explicit`, você só poderá especificar o

Parâmetro	Possíveis valores	Descrição
		nome exato de uma única tabela e esquema (sem curingas). As origens compatíveis com visualizações incluem: • Oracle • Microsoft SQL Server • PostgreSQL • IBM Db2 LUW • IBM Db2 z/OS • SAP Adaptive Server Enterprise (ASE) • MySQL • AURORA • AURORA Sem Servidor • MariaDB **Note** AWS DMS nunca carrega uma visualização de origem em uma visualização de destino. Uma exibição de origem é carregada em uma tabela equivalente no destino com o mesmo nome que a exibição na origem. As origens compatíveis com bancos de dados que contêm coleções incluem:

Parâmetro	Possíveis valores	Descrição
		• MongoDB • Amazon DocumentDB
`load-order`	Um inteiro positivo. O valor máximo é 2.147.483.647.	Indica a prioridade para carregar tabelas e exibições. Tabelas e exibições com valores mais altos são carregadas primeiro.
`filters`	Uma matriz de objetos .	Um ou mais objetos para filtrar a origem. Você especifica parâmetros de objetos para filtragem em uma única coluna na origem. Você especifica vários objetos para filtragem em várias colunas. Para ter mais informações, consulte Usar filtros de origem.

Example Migrar todas as tabelas em um esquema

O exemplo a seguir migra todas as tabelas de um esquema chamado `Test` da origem para o endpoint de destino.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "%"
            },
            "rule-action": "include"
        }
    ]
}
```

Example Migrar algumas tabelas em um esquema

O exemplo a seguir migra todas as tabelas, exceto as que começam com `DMS`, de um esquema chamado `Test` na origem para o endpoint de destino.

```
{
    "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 Migrar uma única tabela especificada em um único esquema

O exemplo a seguir migra a tabela `Customer` do esquema `NewCust` na origem para o endpoint de destino.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "NewCust",
```

```
                "table-name": "Customer"
            },
            "rule-action": "explicit"
        }
    ]
}
```

> **Note**
>
> É possível selecionar explicitamente várias tabelas e esquemas especificando várias regras de seleção.

Example Migrar tabelas em uma ordem definida

O exemplo a seguir migra duas tabelas. A tabela `loadfirst` (com prioridade 1) é inicializada antes da tabela`loadsecond`.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "loadsecond"
            },
            "rule-action": "include",
            "load-order": "2"
        },
        {
            "rule-type": "selection",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "Test",
                "table-name": "loadfirst"
            },
            "rule-action": "include",
            "load-order": "1"
        }
```

```
    ]
}
```

> **Note**
>
> `load-order` é aplicável à inicialização da tabela. A carga de uma tabela sucessiva não aguardará a conclusão de uma carga de tabela anterior se `MaxFullLoadSubTasks` for maior que 1.

Example Migrar algumas exibições em um esquema

O exemplo a seguir migra algumas exibições de um esquema chamado `Test` na origem para tabelas equivalentes no destino.

```
{
   "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 Migrar todas as tabelas e exibições em um esquema

O exemplo a seguir migra todas as tabelas e exibições de um esquema chamado `report` na origem para tabelas equivalentes no destino.

```
{
   "rules": [
        {
           "rule-type": "selection",
           "rule-id": "3",
           "rule-name": "3",
```

```
            "object-locator": {
                "schema-name": "report",
                "table-name": "%",
                "table-type": "all"
             },
            "rule-action": "include"
        }
    ]
}
```

Curingas no mapeamento de tabela

Esta seção descreve os curingas que é possível utilizar ao especificar o esquema e os nomes das tabelas para o mapeamento de tabela.

Curinga	Correspondências
%	Zero ou mais caracteres
_	Um único caractere
[_]	Um caractere de sublinhado literal
[ab]	Um conjunto de caracteres. Por exemplo, [ab] correspon de a 'a' ou 'b'.
[a-d]	Uma variedade de personage ns. Por exemplo, [a-d] corresponde a 'a', 'b', 'c' ou 'd'.

Para endpoints de origem e de destino do Oracle, é possível utilizar o atributo de conexão adicional `escapeCharacter` para especificar um caractere de escape. Um caractere de escape permite que você utilize um caractere curinga especificado em expressões como se não fosse curinga. Por exemplo, `escapeCharacter=#` permite que você utilize '#' para fazer com que um caractere curinga atue como um caractere comum em uma expressão, como neste código de exemplo.

```
{
```

```
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "542485267",
            "rule-name": "542485267",
            "object-locator": { "schema-name": "R00T", "table-name": "TEST#_T%" },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

Aqui, o caractere de escape '#' faz com que o caractere curinga '_' atue como um caractere normal. AWS DMS seleciona tabelas no esquema chamadoR00T, em que cada tabela tem um nome com seu TEST_T prefixo.

Regras de transformação e ações

As ações de transformação são utilizadas para especificar as transformações que você deseja aplicar ao esquema, à tabela ou à visualização selecionada. As regras da transformação são opcionais.

Limitações

- Não é possível aplicar mais de uma ação de regra de transformação ao mesmo objeto (esquema, tabela, coluna, espaço para tabela de tabela ou espaço para tabela de índice). É possível aplicar várias ações de regra de transformação em qualquer nível, desde que cada ação de transformação seja aplicada em um objeto diferente.
- Os nomes de tabelas e de colunas nas regras de transformação diferenciam maiúsculas de minúsculas. Por exemplo, você deve fornecer nomes de tabelas e nomes de colunas para um banco de dados Oracle ou Db2 em letras maiúsculas.
- As transformações não são compatíveis com nomes de colunas com idiomas da direita para a esquerda.
- As transformações não podem ser executadas em colunas que contenham caracteres especiais (por exemplo, #, \, /, -) em seu nome.
- A única transformação compatível com colunas mapeadas para tipos de dados BLOB/CLOB é descartar a coluna no destino.
- AWS DMS não oferece suporte à replicação de duas tabelas de origem em uma única tabela de destino. AWS DMS replica registros de tabela em tabela e de coluna em coluna, de acordo com as

regras de transformação da tarefa de replicação. Os nomes dos objetos devem ser exclusivos para evitar sobreposição.

Por exemplo, uma tabela de origem tem uma coluna chamada `ID` e a tabela de destino correspondente tem uma coluna preexistente chamada `id`. Se uma regra utilizar uma instrução `ADD-COLUMN` para adicionar uma nova coluna chamada `id`, e uma instrução SQLite para preencher a coluna com valores personalizados, isso criará um objeto duplicado e ambíguo chamado `id` e não é compatível.

Valores

Os seguintes valores podem ser aplicados a regras de mapeamento de tabela que utilizam o tipo de regra de transformação

Parâmetro	Possíveis valores	Descrição
`rule-type`	`transformation`	Um valor que aplica a regra a cada objeto especificado pela regra de seleção. Utilize `transformation` a menos que especificado de outra forma.
`rule-id`	Um valor numérico.	Um valor numérico exclusivo para identificar a regra. Se você especific ar várias regras de transformação para o mesmo objeto (esquema, tabela, coluna, espaço entre tabelas ou espaço de tabela de índice), AWS DMS aplique a regra de transform ação com o ID de regra inferior.
`rule-name`	Um valor alfanumérico.	Um nome exclusivo para identificar a regra.
`object-lo cator`	Um objeto com os seguintes parâmetros:	O nome de cada esquema, tabela ou visualização, espaço de tabela de tabela, espaço de tabela de índice e coluna aos quais a regra se aplica. É

Parâmetro	Possíveis valores	Descrição
	• `schema-name` : o nome do esquema. Para endpoints MongoDB e Amazon DocumentDB, esse é o nome do banco de dados que contém um conjunto de coleções. • `table-name` : o nome da tabela, visualização ou coleção. • `table-tablespace-name` : o nome de um espaço para tabela de tabela existente. • `index-tablespace-name` : o nome de um espaço para tabela de índice existente. • `column-name` : o nome de uma coluna existente. • `data-type` : o nome de um tipo de dados de coluna existente.	possível utilizar o sinal de porcentag em "%" como um curinga para todo ou parte do valor de cada parâmetro `object-locator` , exceto `data-type` . Assim, você pode correspon der estes itens: • Uma única tabela ou exibição em um único esquema • Uma única tabela ou exibição em alguns ou todos os esquemas • Algumas ou todas as tabelas e exibições em um único esquema • Algumas ou todas as tabelas e exibições em alguns ou todos os esquemas • Uma ou mais colunas na tabela ou tabelas especificadas, na exibição ou exibições e no esquema ou esquemas. • As colunas com um determina do `data-type` quando várias colunas são especificadas. Para obter os valores possíveis de `data-type` , consulte `data-type` descrito a seguir nesta tabela. Além disso, o parâmetro `index-tab lespace-name` ou `table-tab lespace-name` está disponíve l apenas para corresponder a um endpoint de origem do Oracle. É possível especificar `table-tab`

Parâmetro	Possíveis valores	Descrição
		`lespace-name` ou `index-tab lespace-name` em uma única regra, mas não os dois. Portanto, é possível corresponder qualquer um dos itens a seguir: • Um, alguns ou todos os espaços de tabela de tabela • Um, alguns ou todos os espaços de tabela de índice

Parâmetro	Possíveis valores	Descrição

`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`
		A transformação que você quer aplicar ao objeto. Todas as ações de regra de transformação diferenciam maiúsculas e minúsculas.
		
		O valor de `add-column` do parâmetro `rule-action` adiciona uma coluna a uma tabela. Mas você não pode adicionar uma nova coluna com o mesmo nome de uma coluna já existente da mesma tabela.
		
		Quando utilizado com os parâmetros `expression` e `data-type` , `add-column` especifica o valor de novos dados da coluna.
		
		O valor de `change-data-type` para `rule-action` só está disponível para destinos de regras de `column`.
		
		O valor `include-column` do parâmetro `rule-action` altera o modo da tabela para descartar todas as colunas por padrão e incluir as colunas especificadas. Várias colunas são incluídas no destino com a invocação da regra `include-column` várias vezes.
		
		Não é possível utilizar uma regra `define-primary-key` quando a regra tem um curinga (`%`) em um nome de esquema ou de tabela.

Parâmetro	Possíveis valores	Descrição
		Para uma tarefa existente, as ações da regra de transformação que alteram o esquema da tabela de destino, como `remove-column` , `rename` ou `add-prefix` , não entrarão em vigor enquanto você não reiniciar a tarefa. Se você retomar a tarefa depois de adicionar a regra de transformação, poderá observar um comportamento inesperad o na coluna alterada, que pode incluir dados ausentes da coluna. É necessário reiniciar a tarefa para garantir que a regra de transform ação funcione corretamente.
`rule-target`	`schema`, `table`, `column`, `table-tablespace` , `index-tab lespace`	O tipo de objeto que você está transformando. Os valores `index-tablespace` e `table-tablespace` estão disponíveis somente para um endpoint de destino do Oracle. Especifique um valor para o parâmetro especificado como parte do `object-locator` : nome `table-tablespace-name` ou `index-tablespace-name` .
`value`	Um valor alfanumérico que segue as regras de nomenclatura do tipo de destino.	O novo valor de ações que exigem entrada, como `rename`.

Parâmetro	Possíveis valores	Descrição
`old-value`	Um valor alfanumérico que segue as regras de nomenclatura do tipo de destino.	O antigo valor de ações que exigem substituição, como `replace-prefix` .

Parâmetro	Possíveis valores	Descrição
`data-type`	`type`: o tipo de dados a ser utilizado se `rule-action` for `add-column` ou o tipo de dados de substituição se `rule-action` for `change-data-type`. Ou, o nome do tipo de dados de substituição quando `rule-action` for `change-data-type`, o valor de `column-name` é `"%"`, e um parâmetro `data-type` adicional para identificar o tipo de dados existente está incluído no `object-locator`. AWS DMS suporta transformações de tipo de dados de coluna para os seguintes tipos de dados do DMS: `"bytes", "date", "time", "datetime", "int1", "int2", "int4", "int8", "numeric", "real4", "real8", "string", "uint1", "uint2", "uint4", "uint8", "wstring", "blob", "nclob", "clob", "boolean", "set", "list" "map", "tuple"` `precision`: se a coluna adicionada ou o tipo de dados de substituição tiver uma precisão, um valor inteiro para especificar a precisão. `scale`: se a coluna adicionada ou o tipo de dados de substituição tiver	Veja a seguir um exemplo de um parâmetro `data-type` para especificar o tipo de dados existente a ser substituído. `{ "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": "" } }] }`

Parâmetro	Possíveis valores	Descrição
	uma escala, um valor inteiro ou um valor de data e hora para especificar a escala. `length`: o tamanho dos dados da nova coluna (quando utilizado com `add-column`)	Aqui, a coluna `col10` da tabela `table_t` é alterada para o tipo de dados `string`.

Parâmetro	Possíveis valores	Descrição
`expression`	Um valor alfanumérico que segue a sintaxe SQLite.	Quando utilizado com a `rule-action` definida como `rename-schema` , o parâmetro `expression` especifica um novo esquema. Quando utilizado com a `rule-action` definida como `rename-table` , `expression` especifica uma nova tabela. Quando utilizado com a `rule-action` definida como `rename-column` , `expression` especifica um novo valor de nome de coluna. Quando utilizado com a `rule-action` definida como `add-column` , `expression` especifica dados que compõem uma nova coluna. Observe que somente expressões são compatíveis com esse parâmetro . Os operadores e os comandos são incompatíveis. Para obter mais informações sobre como utilizar expressões para regras de transformação, consulte Utilizar expressões de regra de transformação para definir o conteúdo da coluna. Para obter mais informações sobre expressões SQLite, consulte Utilizar perfis do SQLite para criar expressões.

Parâmetro	Possíveis valores	Descrição
`primary-key-def`	Um objeto com os seguintes parâmetros: • `name`: o nome de uma nova chave primária ou índice exclusivo para a tabela ou visualização. • (Opcional) `origin`: o tipo de chave exclusiva a ser definida: `primary-key` (o padrão) ou `unique-index` . • `columns`: uma matriz de strings que lista os nomes de colunas na ordem em que aparecem na chave primária ou índice exclusivo.	Esse parâmetro pode definir o nome, o tipo e o conteúdo de uma chave exclusiva na tabela ou visualização transformada. Ele faz isso quando a `rule-action` é definida como `define-primary-key` e o `rule-target` é definido como `table`. Por padrão, a chave exclusiva é definida como uma chave primária.

Parâmetro	Possíveis valores	Descrição
`before-image-def`	Um objeto com os seguintes parâmetros: • `column-prefix` : um valor que precede um nome de coluna. O valor padrão é `BI_`. • `column-suffix` : um valor acrescentado ao nome da coluna. O padrão é vazio. • `column-filter` : requer um dos seguintes valores: `pk-only` (padrão), `non-lob` (opcional) e `all` (opcional).	Esse parâmetro define uma convenção de nomenclatura para identificar as colunas de imagem anterior e especifica um filtro para identificar quais colunas de origem podem ter colunas de imagem anterior criadas para elas no destino. É possível especificar esse parâmetro quando a `rule-action` é definida como `add-before-image-columns` e o `rule-target` é definido como `column`. Não defina `column-prefix` e `column-suffix` como strings vazias. Para `column-filter` , selecione: • `pk-only`: para adicionar somente colunas que façam parte das chaves primárias da tabela. • `non-lob`: para adicionar somente colunas que não sejam do tipo LOB. • `all`: para adicionar qualquer coluna que tenha um valor de imagem anterior. Para obter mais informações sobre suporte a imagem anterior para endpoints de destino do AWS DMS , consulte:

Parâmetro	Possíveis valores	Descrição
		• Utilizar uma imagem anterior para visualizar valores originais de linhas da CDC para um fluxo de dados do Kinesis como destino • Utilizar uma imagem anterior para visualizar os valores originais de linhas da CDC para o Apache Kafka como destino

Exemplos

Example Renomear um esquema

O exemplo a seguir renomeia um esquema de `Test` na origem para `Test1` no destino.

```
{

    "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 Renomeação de uma tabela

O exemplo a seguir renomeia uma tabela de `Actor` na origem para `Actor1` no destino.

```
{
    "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 Renomeação de uma coluna

O exemplo a seguir renomeia uma coluna na tabela `Actor` de `first_name` na origem para `fname` no destino.

```
{
    "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 Renomear um espaço de tabela da tabela do Oracle

O exemplo a seguir renomeia o espaço de tabela denominado `SetSpace` para uma tabela denominada `Actor` na origem do Oracle para `SceneTblSpace` no endpoint de destino do Oracle.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
```

```
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "rename",
            "rule-target": "table-tablespace",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "Actor",
                "table-tablespace-name": "SetSpace"
            },
            "value": "SceneTblSpace"
        }
    ]
}
```

Example Renomear um espaço de tabela de índice do Oracle

O exemplo a seguir renomeia o espaço de tabela de índice denominado `SetISpace` para uma tabela denominada `Actor` na origem do Oracle para `SceneIdxSpace` no endpoint de destino do Oracle.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "rename",
            "rule-target": "table-tablespace",
            "object-locator": {
                "schema-name": "Play",
                "table-name": "Actor",
                "table-tablespace-name": "SetISpace"
            },
```

```
            "value": "SceneIdxSpace"
        }
    ]
}
```

Example Adicionar uma coluna

O exemplo a seguir adiciona uma coluna `datetime` à tabela `Actor` no esquema `test`.

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "test",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "transformation",
            "rule-id": "2",
            "rule-name": "2",
            "rule-action": "add-column",
            "rule-target": "column",
            "object-locator": {
                "schema-name": "test",
                "table-name": "actor"
            },
            "value": "last_updated",
            "data-type": {
                "type": "datetime",
                "precision": 6
            }
        }
    ]
}
```

Example Remover uma coluna

O exemplo a seguir transforma a tabela `Actor` na origem para remover todas as colunas que começam com os caracteres `col` no destino.

```
{
  "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 Converter para minúsculas

O exemplo a seguir converte o nome de uma tabela de `ACTOR` na origem em `actor` no destino.

```
{
 "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 Converter em maiúsculas

O exemplo a seguir converte todas as colunas em todas as tabelas e todos os esquemas de minúsculas, na origem, para maiúsculas, no destino.

```
{
    "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 Adicionar um prefixo

O exemplo a seguir transforma todas as tabelas na origem para adicionar o prefixo `DMS_` a elas no destino.

```
{
  "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 Substituir um prefixo

O exemplo a seguir transforma todas as colunas contendo o prefixo `Pre_` na origem para substituí-lo por `NewPre_` no destino.

```
{
    "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 Remover um sufixo

O exemplo a seguir transforma todas as tabelas na origem para remover o sufixo `_DMS` delas no destino.

```
{
 "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 Definir uma chave primária

O exemplo a seguir define uma chave primária denominada `ITEM-primary-key` em três colunas da tabela `ITEM` migradas para o endpoint de destino.

```
{
 "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 Definir um índice exclusivo

O exemplo a seguir define um índice exclusivo denominado `ITEM-unique-idx` em três colunas da tabela `ITEM` migradas para o endpoint de destino.

```
{
 "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 Alterar o tipo de dados da coluna de destino

O exemplo a seguir altera o tipo de dados de uma coluna de destino chamada `SALE_AMOUNT` de um tipo de dados existente para `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 Adicionar uma coluna de imagem anterior

Para uma coluna de origem chamada emp_no, a regra de transformação no exemplo a seguir adiciona uma nova coluna chamada BI_emp_no no destino.

```
{
 "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"
   }
  }
 ]
}
```

Aqui, a instrução a seguir preenche uma coluna `BI_emp_no` na linha correspondente com 1.

```
UPDATE employees SET emp_no = 3 WHERE BI_emp_no = 1;
```

Ao escrever atualizações do CDC em AWS DMS destinos compatíveis, a `BI_emp_no` coluna possibilita saber quais linhas têm valores atualizados na `emp_no` coluna.

Utilizar expressões de regra de transformação para definir o conteúdo da coluna

Para definir o conteúdo de colunas novas e existentes, é possível utilizar uma expressão em uma regra de transformação. Por exemplo, utilizando expressões, é possível adicionar uma coluna ou replicar cabeçalhos de tabela de origem para um destino. Também é possível utilizar expressões para sinalizar registros em tabelas de destino como inseridos, atualizados ou excluídos na origem.

Tópicos

- Adicionar uma coluna utilizando uma expressão
- Sinalizar registros de destino utilizando uma expressão
- Replicar cabeçalhos de tabela de origem utilizando expressões
- Utilizar perfis do SQLite para criar expressões
- Adicionar metadados a uma tabela de destino utilizando expressões

Adicionar uma coluna utilizando uma expressão

Para adicionar colunas a tabelas utilizando uma expressão em uma regra de transformação, utilize uma ação de regra `add-column` e um destino de regra `column`.

O exemplo a seguir adiciona uma nova coluna à tabela `ITEM`. Ele define o nome da nova coluna como `FULL_NAME`, com um tipo de dados de `string`, com 50 caracteres. A expressão concatena

os valores de duas colunas existentes, `FIRST_NAME` e `LAST_NAME`, para avaliar para `FULL_NAME`. Os parâmetros `schema-name` e `table-name` e de expressão se referem aos objetos na tabela do banco de dados de origem. `Value` e o bloco `data-type` se referem aos objetos na tabela do banco de dados de destino.

```
{
    "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
            }
        }
    ]
}
```

Sinalizar registros de destino utilizando uma expressão

Para sinalizar registros em tabelas de destino como inseridos, atualizados ou excluídos na tabela de origem, utilize uma expressão em uma regra de transformação. A expressão utiliza um perfil `operation_indicator` para sinalizar registros. Os registros excluídos da origem não são

excluídos do destino. Em vez disso, o registro de destino é sinalizado com um valor fornecido pelo usuário para indicar que ele foi excluído da origem.

> **Note**
>
> O perfil `operation_indicator` funciona somente em tabelas que têm uma chave primária no banco de dados de origem e de destino.

Por exemplo, a regra de transformação a seguir primeiro adiciona uma nova coluna `Operation` à uma tabela de destino. Ela atualiza a coluna com o valor `D` sempre que um registro for excluído de uma tabela de origem.

```
{
      "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
      }
}
```

Replicar cabeçalhos de tabela de origem utilizando expressões

Por padrão, os cabeçalhos das tabelas de origem não são replicados no destino. Para indicar quais cabeçalhos a serem replicados, utilize uma regra de transformação com uma expressão que inclua o cabeçalho da coluna da tabela.

É possível utilizar os cabeçalhos de coluna a seguir em expressões.

Cabeçalho	Valor na replicação contínua	Valor na carga completa	Tipo de dados
AR_H_STREAM_POSITION	O valor da posição do streaming da origem. Esse valor pode ser o número de alterações do sistema (SCN) ou o número de sequência de log (LSN), dependendo do endpoint de origem.	Uma string vazia.	STRING
AR_H_TIMESTAMP	Um time stamp indicando a hora da alteração.	Um timestamp indicando a hora atual em que os dados chegam ao destino.	DATETIME (escala=7)
AR_H_COMMIT_TIMESTAMP	Um time stamp indicando a hora da confirmação.	Um time stamp indicando a hora atual.	DATETIME (escala=7)
AR_H_OPERATION	INSERT, UPDATE ou DELETE	INSERT	STRING
AR_H_USER	O nome de usuário, ID ou qualquer outra informação fornecida pela origem sobre o usuário que fez a alteração. Esse cabeçalho tem suporte somente nos endpoints de origem SQL Server e Oracle	A transformação que você deseja aplicar ao objeto. As ações de regra de transformação diferenciam maiúsculas e minúsculas.	STRING

Cabeçalho	Valor na replicação contínua	Valor na carga completa	Tipo de dados
	(versão 11.2.0.3 e posterior).		
AR_H_CHANGE_SEQ	Um número de incremento exclusivo do banco de dados de origem que consiste em um timestamp e em um número de incremento automático. O valor depende do sistema do banco de dados de origem.	Uma string vazia.	STRING

O exemplo a seguir adiciona uma nova coluna ao destino utilizando o valor da posição do fluxo da origem. Para o SQL Server, o valor da posição do fluxo é o LSN do endpoint de origem. Para o Oracle, o valor da posição do fluxo é o SCN do endpoint de origem.

```
{
      "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
      }
    }
```

O exemplo a seguir adiciona uma nova coluna ao destino que tem um número incremental exclusivo da origem. Esse valor representa um número exclusivo de 35 dígitos no nível da tarefa. Os primeiros 16 dígitos fazem parte de um timestamp e os últimos 19 dígitos são o número de record_id incrementado pelo DBMS.

```
{
"rule-type": "transformation",
"rule-id": "2",
"rule-name": "2",
"rule-target": "column",
"object-locator": {
"schema-name": "%",
"table-name": "%"
},
"rule-action": "add-column",
"value": "transact_id",
"expression": "$AR_H_CHANGE_SEQ",
"data-type": {
"type": "string",
"length": 50
}
}
```

Utilizar perfis do SQLite para criar expressões

Utilize as configurações da tabela para especificar quaisquer configurações que deseja aplicar à tabela ou à visualização selecionada para uma operação especificada. As regras de configuração de tabela são opcionais.

> **Note**
>
> Em vez do conceito de tabelas e visualizações, os bancos de dados MongoDB e DocumentDB armazenam os registros de dados como documentos reunidos em coleções. Portanto, ao migrar de uma origem do MongoDB ou do DocumentDB, considere o tipo de segmentação por intervalo das configurações de carga paralela para coleções selecionadas, em vez de tabelas e visualizações.

Tópicos

- Utilizar uma expressão CASE
- Exemplos

A seguir, você encontrará perfis de string que podem ser utilizadas para criar expressões de regras de transformação.

Funções de string	Descrição
`lower(x)`	O perfil `lower(x)` retorna uma cópia da string *x* com todos os caracteres convertidos em minúsculas. O padrão, o perfil `lower` integrado funciona somente para caracteres ASCII.
`upper(x)`	O perfil `upper(x)` retorna uma cópia da string *x* com todos os caracteres convertidos em maiúsculas. O padrão, o perfil `upper` integrado funciona somente para caracteres ASCII.
`ltrim(x,y)`	O perfil `ltrim(x,y)` retorna uma string formada pela remoção de todos os caracteres que aparecem em y do lado esquerdo de x. Se não houver nenhum valor para y, `ltrim(x)` removerá os espaços do lado esquerdo de x.
`replace(x,y,z)`	O perfil `replace(x,y,z)` retorna uma string formada pela substituição da string z de cada ocorrência da string y na string x.
`rtrim(x,y)`	O perfil `rtrim(x,y)` retorna uma string formada pela remoção de todos os caracteres que aparecem em y do lado direito de x. Se não houver nenhum valor para y, `rtrim(x)` removerá os espaços do lado direito de x.
`substr(x,y,z)`	O perfil `substr(x,y,z)` retorna uma substring da string de entrada *x* que começa com o *y*º caractere e tem *z* caracteres. Se *z* for omitido, `substr(x,y)` retornará todos os caractere s até o final da string *x* começando com o caractere *y*º. O caractere mais à esquerda de *x* é o número 1. Se *y* for negativo, o primeiro caractere da substring será encontrado contando a partir da direita em vez da esquerda. Se *z* for negativo, os

Funções de string	Descrição
	caracteres `abs(z)` anteriores ao *y*º caractere serão retornados. Se *x* for uma string, os índices de caracteres se referem aos caracteres UTF-8 reais. Se *x* for um BLOB, os índices se referirão a bytes.
`trim(x,y)`	O perfil `trim(x,y)` retorna uma string formada pela remoção de todos os caracteres que aparecem em *y* do dois lados de *x*. Se não houver nenhum valor para *y*, `trim(x)` removerá os espaços dos dois lados de *x*.

A seguir, é possível encontrar perfis de LOB que podem ser utilizados para criar expressões de regras de transformação.

Perfis de LOB	Descrição
`hex(x)`	O perfil `hex` recebe um BLOB como argumento e retorna uma versão de string hexadecimal em maiúsculas do conteúdo do BLOB.
`randomblob ( N)`	O perfil `randomblob( N)` retorna um BLOB de *N*-bytes que contém bytes pseudoaleatórios. Se *N* for menor que 1, um BLOB aleatório de 1 byte será retornado.
`zeroblob(N)`	O perfil `zeroblob( N)` retorna um BLOB que consiste em *N* bytes de 0x00.

A seguir, é possível encontrar perfis numéricos que podem ser utilizados para criar expressões de regras de transformação.

Perfis numéricos	Descrição
`abs(x)`	O perfil `abs(x)` retorna o valor absoluto do argumento numérico *x*. O perfil `abs(x)` retornará NULL se *x* for NULL. O perfil

Perfis numéricos	Descrição
	`abs(x)` retornará 0,0 se *x* for uma string ou BLOB que não pode ser convertido em um valor numérico.
`random()`	O perfil `random` retorna um número inteiro pseudoaleatório entre -9.223.372.036.854.775.808 e +9.223.372.036.854 .775.807.
`round (x,y)`	O perfil `round (x,y)` retorna um valor de ponto flutuante *x* arredondado para *y* dígitos à direita do ponto decimal. Se não houver nenhum valor para *y*, presume-se que ele seja 0.
`max (x,y...)`	O perfil `max` de multiargumento retorna o argumento com o valor máximo ou retorna NULL se algum argumento for NULL. O perfil `max` pesquisa seus argumentos da esquerda para a direita em busca de um argumento que defina um perfil de agrupamento. Se um for encontrado, ele utilizará esse perfil de agrupamento para todas as comparações de strings. Se nenhum dos argumentos para `max` definir um perfil de agrupamento, o perfil de agrupamento `BINARY` será utilizado. O perfil `max` é simples quando há dois ou mais argumentos, mas funciona como um perfil agregado se tiver um único argumento.
`min (x,y...)`	O perfil `min` de multiargumento retorna o argumento com o valor mínimo. O perfil `min` pesquisa seus argumentos da esquerda para a direita em busca de um argumento que defina um perfil de agrupamento. Se um for encontrado, ele utilizará esse perfil de agrupamento para todas as comparações de strings. Se nenhum dos argumentos para `min` definir um perfil de agrupamento, o perfil de agrupamento `BINARY` será utilizado. O perfil `min` é simples quando há dois ou mais argumentos, mas funciona como um perfil agregado se tiver um único argumento.

A seguir, é possível encontrar perfis de verificação de NULL que podem ser utilizados para criar expressões de regras de transformação.

Perfis de verificação NULL	Descrição
`coalesce (x,y...)`	O perfil `coalesce` retorna uma cópia do primeiro argumento não NULL, mas retornará NULL se todos os argumentos forem NULL. O perfil de agrupamento tem pelo menos dois argumento s.
`ifnull(x,y)`	O perfil `ifnull` retorna uma cópia do primeiro argumento não NULL, mas retornará NULL se os dois argumentos forem NULL. O perfil `ifnull` tem exatamente dois argumentos. O perfil `ifnull` é igual a `coalesce` com dois argumentos.
`nullif(x,y)`	O perfil `nullif(x,y)` retornará uma cópia do primeiro argumento se os argumentos forem diferentes, mas retornará NULL se os argumentos forem iguais. O perfil `nullif(x,y)` pesquisa seus argumentos da esquerda para a direita em busca de um argumento que defina um perfil de agrupamento. Se um for encontrado, ele utilizará esse perfil de agrupamento para todas as comparações de strings. Se nenhum dos argumentos nullif definir um perfil de agrupamento, o perfil de agrupamento `BINARY` será utilizado.

A seguir, é possível encontrar perfis de data e hora que podem ser utilizados para criar expressões de regras de transformação.

Perfis de data e hora	Descrição
`date(timestring , modifier, modifier...)`	O perfil `date` retorna a data no formato DD-MM-AAAA.
`time(timestring , modifier, modifier...)`	O perfil `time` retorna a hora no formato HH:MM:SS.

Perfis de data e hora	Descrição
`datetime( timestring , modifier, modifier...)`	O perfil `datetime` retorna a data e a hora no formato DD-MM-AAAA HH:MM:SS.
`julianday( timestring , modifier, modifier...)`	O perfil `julianday` retorna o número de dias desde o meio-dia em Greenwich em 24 de novembro de 4714 a.C.
`strftime( format, timestring , modifier, modifier...)`	O perfil `strftime` retorna a data de acordo com a string de formato especificada como o primeiro argumento, utilizando uma das seguintes variáveis: `%d`: dia do mês `%H`: hora 00 a 24 `%f`: ** segundos fracionários SS.SSS `%j`: dia do ano 001 a 366 `%J`: ** número do dia juliano `%m`: mês 1 a 12 `%M`: minuto 00 a 59 `%s`: segundos desde 1-1-1970 `%S`: segundos 00 a 59 `%w`: dia da semana 0 a 6 domingo==0 `%W`: semana do ano 00 a 53 `%Y`: ano 0000 a 9999 `%%`: %

A seguir, é possível encontrar um perfil de hash que pode ser utilizado para criar expressões de regras de transformação.

Função de hash	Descrição
`hash_sha256( x)`	O perfil `hash` gera um valor de hash para uma coluna de entrada (utilizando o algoritmo SHA-256) e retorna o valor hexadecimal do valor de hash gerado. Para utilizar o perfil `hash` em uma expressão, adicione `hash_sha256( x)` à expressão e substitua *x* pelo nome da coluna de origem.

Utilizar uma expressão CASE

A expressão `CASE` do SQLite avalia uma lista de condições e retorna uma expressão com base no resultado. A sintaxe é mostrada a seguir.

```
    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
```

Exemplos

Example de adição de uma nova coluna de string à tabela de destino utilizando uma condição de caso

O seguinte exemplo de regra de transformação adiciona uma nova coluna string, `emp_seniority`, à tabela de destino, `employee`. Ele utiliza o perfil `round` do SQLite na coluna de salário, com uma condição de caso para verificar se o salário é igual ou superior a 20.000. Se isso acontecer, a coluna obterá o valor `SENIOR` e qualquer outra coisa terá o valor `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 da adição de uma nova coluna de data à tabela de destino

O exemplo a seguir adiciona uma nova coluna de data, `createdate`, à tabela de destino, `employee`. Quando você utiliza o perfil `datetime` de data do SQLite, a data é adicionada à tabela recém-criada para cada linha inserida.

```
 {
     "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 da adição de uma nova coluna numérica à tabela de destino

O exemplo a seguir adiciona uma nova coluna numérica, `rounded_emp_salary`, à tabela de destino, `employee`. Ele utiliza o perfil `round` do SQLite para adicionar o salário arredondado.

```
{
    "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 da adição de uma nova coluna string à tabela de destino utilizando o perfil hash

O exemplo a seguir adiciona uma nova coluna string, `hashed_emp_number`, à tabela de destino, `employee`. O perfil `hash_sha256(x)` do SQLite cria valores com hash no destino para a coluna de origem, `emp_number`.

```
{
    "rule-type": "transformation",
    "rule-id": "2",
    "rule-name": "2",
    "rule-action": "add-column",
    "rule-target": "column",
    "object-locator": {
      "schema-name": "public",
      "table-name": "employee"
    },
    "value": "hashed_emp_number",
    "expression": "hash_sha256($emp_number)",
```

```
    "data-type": {
      "type": "string",
      "length": 64
    }
}
```

Adicionar metadados a uma tabela de destino utilizando expressões

É possível adicionar informações de metadados à tabela de destino utilizando as seguintes expressões:

- `$AR_M_SOURCE_SCHEMA`: o nome do esquema de origem.
- `$AR_M_SOURCE_TABLE_NAME`: o nome da tabela de origem.
- `$AR_M_SOURCE_COLUMN_NAME`: o nome de uma coluna na tabela de origem.
- `$AR_M_SOURCE_COLUMN_DATATYPE`: o tipo de dados de uma coluna na tabela de origem.

Example da adição de uma coluna para um nome de esquema utilizando o nome do esquema da origem

O exemplo a seguir adiciona uma nova coluna `schema_name` ao destino utilizando o nome do esquema da origem.

```
{
    "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
    }
}
```

Regras e operações de configurações de tabelas e coleções

Utilize as configurações de tabela para especificar quaisquer configurações que deseja aplicar a uma tabela ou visualização selecionada para uma operação especificada. As regras de configuração de tabela são opcionais, dependendo do endpoint e dos requisitos da migração.

Em vez de utilizar tabelas e visualizações, os bancos de dados MongoDB e Amazon DocumentDB armazenam registros de dados como documentos que são reunidos em coleções. Um único banco de dados de qualquer endpoint do MongoDB ou do Amazon DocumentDB é um conjunto específico de coleções identificadas pelo nome do banco de dados.

Ao migrar de uma origem do MongoDB ou do Amazon DocumentDB, você trabalha com configurações de carregamento paralelo de forma um pouco diferente. Nesse caso, considere o tipo de segmentação automática ou de segmentação por intervalo das configurações de carga paralela para coleções selecionadas, em vez de tabelas e visualizações.

Tópicos

Para as regras de mapeamento de tabela que utilizam o tipo de regra de configurações de tabela, é possível aplicar os parâmetros a seguir.

Parâmetro	Possíveis valores	Descrição
`rule-type`	`table-settings`	Um valor que aplica a regra a uma tabela, visualização ou coleção especificada pela regra de seleção.
`rule-id`	Um valor numérico.	Um valor numérico exclusivo para identificar a regra.

Parâmetro	Possíveis valores	Descrição
`rule-name`	Um valor alfanumérico.	Um nome exclusivo para identificar a regra.
`object-locator`	Um objeto com os seguintes parâmetros: • `schema-name` : o nome do esquema. Para endpoints MongoDB e Amazon DocumentDB, esse é o nome do banco de dados que contém um conjunto de coleções. • `table-name` : o nome da tabela, visualização ou coleção.	O nome de um esquema e tabela ou visualização específicas ou o nome de um banco de dados e coleção específicos (sem curingas).

Parâmetro	Possíveis valores	Descrição
`parallel-load`	Um objeto com os seguintes parâmetros: • `type`: especifica se a carga paralela está ativada. Se estiver, esse parâmetro também especificará o mecanismo para identificar as partições da tabela ou da exibição, as subpartiç ões ou outros segmentos a serem carregados em paralelo. Partições são segmentos que já estão definidos e identificados pelo nome da tabela ou da exibição de origem. Para endpoints do MongoDB e do Amazon DocumentDB, as partições são segmentos . AWS DMS pode calculá-los automaticamente de acordo com os parâmetros de segmentaç ão automática associados. Ou é possível especificá-los manualmen te utilizando parâmetros de segmentação por intervalo. Somente para endpoints do Oracle, as subpartições são um nível adicional de segmentos que já estão definidos e identific ados pelo nome na tabela ou na exibição de origem. É possível identificar outros segmentos na regra `table-settings`	Um valor que especifica uma operação de carga paralela (multi-threaded) na tabela ou na exibição identificada pela opção `object-locator` . Nesse caso, é possível carregar em paralelo das seguintes maneiras: • Por segmentos especific ados por todas as partições ou subpartiç ões disponíveis. • Por partições e subpartiç ões selecionadas. • Por segmentaç ão automática ou segmentos com base em intervalos especificados. Para obter mais informaçõ es sobre carga paralela, consulte Utilizar carga paralela para tabelas, visualizações e coleções selecionadas.

Parâmetro	Possíveis valores	Descrição
	especificando limites no intervalo de valores para uma ou mais colunas de tabela ou de visualização. • `partitions` : quando `type` for `partitions-list` , esse valor especificará todas as partições a serem carregadas em paralelo. • `subpartitions` : somente para endpoints do Oracle, quando `type` for `partitions-list` esse valor especificará todas as subpartições a serem carregadas em paralelo. • `columns`: quando `type` for `ranges`, esse valor especificará os nomes das colunas utilizadas para identificar os segmentos com base em intervalo a serem carregados em paralelo. • `boundaries` : quando `type` for `ranges`, esse valor especificará os valores das `columns` utilizadas para identificar segmentos com base em intervalos a serem carregados em paralelo.	

Parâmetro	Possíveis valores	Descrição
`type`	Um dos seguintes para `parallel-load` : • `partitions-auto` : todas as partições da tabela ou da visualização são carregadas em paralelo. Cada partição é alocada a seu próprio thread. Essa é uma configuração necessária para que os endpoints de origem do MongoDB e do Amazon DocumentDB utilizem a opção de segmentação automática de uma carga máxima paralela. • `subpartitions-auto` : (somente endpoints do Oracle) todas as subpartições da tabela ou da visualização são carregadas em paralelo. Cada subpartição é alocada para seu próprio thread. • `partitions-list` : todas as partições especificadas da tabela ou da visualização são carregadas em paralelo. Somente para endpoints do Oracle, todas as subpartições especificadas da tabela ou da exibição são carregadas em paralelo. Cada partição e subpartição que você especificar será alocada para seu próprio thread. Você especifica as partições e subpartições a serem carregadas em paralelo pelos nomes das partições (`partition`	O mecanismo para identificar as partições ou subpartições ou segmentos de tabela, visualização ou coleção a serem carregados em paralelo.

Parâmetro	Possíveis valores	Descrição
	`s`) e nomes das subpartições (`subpartitions`). • `ranges`: todos os segmentos especificados do intervalo da tabela, visualização ou coleção são carregados em paralelo. Cada segmento de tabela, visualização ou coleção que você identificar será alocado para seu próprio thread. Você especifica esses segmentos por nomes de coluna (`columns`) e valores da coluna (`boundaries`). Os endpoints do PostgreSQL são compatíveis somente com esse tipo de carga paralela. O MongoDB e o Amazon DocumentDB como endpoints de origem são compatíveis com esse tipo de segmentação por intervalo e com o tipo de segmentação automática de uma carga máxima paralela (`partitions-auto`). • `none`: a tabela, visualização ou coleção é carregada em uma tarefa de thread único (o padrão), independentemente das suas partições ou subpartições. Para ter mais informações, consulte Criar uma tarefa.	

Parâmetro	Possíveis valores	Descrição
`number-of-partitions`	(Opcional) Quando `type` é `partitions-auto` para coleções específicas de um endpoint do MongoDB ou do Amazon DocumentDB, esse parâmetro especifica o número total de partições (segmentos) utilizadas para a migração. O padrão é 16.	Especifica o número exato de partições a serem carregadas em paralelo.
`collection-count-from-metadata`	(Opcional) Quando `type` é `partitions-auto` para coleções especificadas de um endpoint MongoDB ou Amazon DocumentDB e esse parâmetro é definido como AWS DMS , usa uma contagem de coleta estimada `true` para determinar o número de partições. Se esse parâmetro for definido como`false`, AWS DMS usa a contagem real da coleta. O padrão é `true`.	Especifica se uma contagem estimada de coleção ou a contagem real de coleção deve ser utilizada para calcular o número de partições a serem carregadas em paralelo.
`max-records-skip-per-page`	(Opcional) Quando `type` for `partitions-auto` de coleções específicas de um endpoint do MongoDB ou do Amazon DocumentDB, esse é o número de registros a serem ignorados de uma vez ao determinar os limites de cada partição. O AWS DMS utiliza uma abordagem de salto paginado para determinar o limite mínimo de uma partição. O padrão é 10.000.	Especifica o número de registros a serem ignorados de uma vez ao determinar os limites de cada partição. A definição de um valor relativamente grande do padrão pode resultar em tempos limite do cursor e falhas na tarefa. A definição de um valor relativamente resulta em mais operações por página e em uma carga máxima mais lenta.

Parâmetro	Possíveis valores	Descrição
`batch-size`	(Opcional) Quando `type` é `partitions-auto` para coleções especificadas de um endpoint do MongoDB ou do Amazon DocumentDB, esse valor inteiro limita o número de documentos retornados em um lote de ida e volta. Se o tamanho do lote for zero (0), o cursor utilizará o tamanho máximo do lote definido pelo servidor. O padrão é 0.	Especifica o número máximo de documentos retornados em um lote. Cada lote requer uma viagem de ida e volta ao servidor.
`partitions`	Quando `type` for `partitions-list`, isso será uma matriz de strings que especificam os nomes das partições a serem carregadas em paralelo.	Os nomes das partições a serem carregadas em paralelo.
`subpartitions`	(Somente para endpoints do Oracle) Quando `type` for `partitions-list`, isso será uma matriz de strings que especifica os nomes das subpartições a serem carregadas em paralelo.	Os nomes de subpartições a serem carregadas em paralelo.
`columns`	Quando `type` for `ranges`, uma matriz de strings definidas como os nomes das colunas que identificam os segmentos baseados em intervalo de tabela, visualização ou coleção a serem carregados em paralelo.	Os nomes das colunas que identificam os segmentos baseados em intervalo de tabela, visualização ou coleção a serem carregados em paralelo.

Parâmetro	Possíveis valores	Descrição
`boundaries`	Quando `type` for `ranges`, uma matriz de matrizes de coluna e vamor. Cada matriz de valores de coluna contém valores de colunas na quantidade e na ordem especific adas por `columns`. Uma matriz de coluna-valor especifica o limite superior de um segmento de uma tabela, visualização ou coleção. Cada matriz de coluna-valor extra adiciona o limite superior para um segmento adicional de tabela, visualização ou coleção. Todos esses segmentos baseados em intervalos de tabela, visualização ou coleção são carregados em paralelo.	Os valores de colunas que identificam as partições baseada em intervalo de tabela, visualização ou coleção são carregados em paralelo.

Parâmetro	Possíveis valores	Descrição
`lob-settings`	Um objeto com os seguintes parâmetros: • `mode`: especifica o modo de tratamento da migração para LOBs. • `bulk-max-size` : especifica o tamanho máximo de LOBs, dependendo da configuração de `mode`.	Um valor que especific a o tratamento de LOB para a tabela ou a exibição identificada pela opção `object-locator` . O tratamento de LOB especificado substitui qualquer configuração de tarefa de LOB somente para essa tabela ou exibição. Para obter mais informações sobre como utilizar os parâmetro s de configurações de LOB, consulte Especificar configurações de LOB para uma tabela ou exibição selecionada.

Parâmetro	Possíveis valores	Descrição
`mode`	Especifica o tratamento da migração para LOBs na tabela ou visualização especificada utilizando os seguintes valores: • `limited`: (padrão) esse valor define a migração para o modo LOB limitado, com todos os LOBs migrados em linha junto com todos os outros tipos de dados de coluna na tabela ou visualização. Utilize esse valor ao replicar principal mente LOBs pequenos (100 MB ou menos). Além disso, especifiq ue um valor `bulk-max-size` (zero é inválido). Todos os LOBs migrados maiores que `bulk-max-size` são truncados para o tamanho que você definir. • `unlimited` : este valor define a migração como modo LOB completo. Utilize esse valor quando todos ou a maioria dos LOBs que você deseja replicar forem maiores que 1 GB. Se você definir `bulk-max-size` com valor zero, todos os LOBs serão migrados no modo LOB completo padrão. Nesta forma de modo `unlimited` , todos os LOBs são migrados separadam ente de outros tipos de dados de coluna utilizando uma pesquisa da tabela ou da exibição de origem. Se você especificar um valor	O mecanismo utilizado para migrar LOBs.

Parâmetro	Possíveis valores	Descrição
	de `bulk-max-size` maior que zero, todos os LOBs serão migrados em combinação no modo LOB completo total. Nessa forma de modo `unlimited` , os LOBs maiores que `bulk-max-size` são migrados utilizando uma tabela ou uma exibição de origem, semelhante à pesquisa padrão do modo LOB completo. Caso contrário, os LOBs até esse tamanho e incluindo esse tamanho serão migrados em linha, semelhante ao modo LOB limitado. Nunca nenhum LOB é truncado no modo `unlimited` , independentemente do formato que você usa. • `none`: todos os LOBs da tabela ou da visualização são migrados de acordo com as configurações de LOB da tarefa. Para obter mais informações sobre as configurações de LOB de tarefa, consulte Configurações de tarefa de metadados de destino. Para obter mais informações sobre como migrar LOBs e como especificar essas configurações de LOB de tarefa, consulte Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS.	

Parâmetro	Possíveis valores	Descrição
`bulk-max-size`	O efeito desse valor depende do `mode`.	O tamanho máximo de LOBs em incrementos de kilobytes. Especifique essa opção somente se você precisar replicar pequenos LOBs ou se o endpoint de destino não for compatíve l com o tamanho de LOB ilimitado.

Curingas nas configurações de tabela são restritos

A utilização do curinga de porcentagem (`"%"`) em regras `"table-settings"` não é compatível com bancos de dados de origem, conforme mostrado a seguir.

```
{
    "rule-type": "table-settings",
    "rule-id": "8",
    "rule-name": "8",
    "object-locator": {
        "schema-name": "ipipeline-prod",
        "table-name": "%"
    },
    "parallel-load": {
        "type": "partitions-auto",
        "number-of-partitions": 16,
        "collection-count-from-metadata": "true",
        "max-records-skip-per-page": 1000000,
        "batch-size": 50000
    }
  }
```

Se você usar `"%"` as `"table-settings"` regras conforme mostrado, AWS DMS retornará a exceção a seguir.

```
Error in mapping rules. Rule with ruleId = x failed validation. Exact
schema and table name required when using table settings rule.
```

Além disso, AWS recomenda que você não carregue um grande número de coleções grandes usando uma única tarefa com`parallel-load`. Observe que o AWS DMS limita a contenção de recursos, bem como o número de segmentos carregados em paralelo pelo valor do parâmetro de configurações de tarefa `MaxFullLoadSubTasks`, com um valor máximo de 49.

Em vez disso, especifique todas as coleções do banco de dados de origem para as maiores coleções especificando cada `"schema-name"` e `"table-name"` individualmente. Além disso, aumente a escala verticalmente da migração de forma adequada. Por exemplo, execute várias tarefas em um número suficiente de instâncias de replicação para tratar um grande número de coleções grandes no banco de dados.

Utilizar carga paralela para tabelas, visualizações e coleções selecionadas

Para acelerar a migração e torná-la mais eficiente, é possível utilizar a carga paralela para tabelas, visualizações e coleções selecionadas. Em outras palavras, é possível migrar uma única tabela, visualização ou coleção utilizando vários threads em paralelo. Para fazer isso, AWS DMS divide uma tarefa de carga completa em segmentos, com cada segmento da tabela alocado em seu próprio encadeamento.

Utilizando esse processo de carregamento paralelo, é possível primeiro fazer o upload de vários threads de várias tabelas, visualizações e coleções em paralelo no endpoint de origem. É possível que vários threads migrem e carreguem as mesmas tabelas, visualizações e coleções em paralelo para o endpoint de destino. Para alguns mecanismos de banco de dados, é possível segmentar as tabelas e as exibições pelas partições ou subpartições existentes. Para outros mecanismos de banco de dados, você pode segmentar AWS DMS automaticamente as coleções de acordo com parâmetros específicos (segmentação automática). Caso contrário, é possível segmentar qualquer tabela, visualização ou coleção pelos intervalos de valores de coluna especificados.

O carregamento paralelo é compatível com os seguintes endpoints de origem:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2 LUW
- SAP Adaptive Server Enterprise (ASE)
- MongoDB (compatível somente com as opções de segmentação automática e com a segmentação por intervalo de uma carga máxima paralela)

- Amazon DocumentDB (compatível somente com as opções de segmentação automática e com a segmentação por intervalo de uma carga máxima paralela)

Para endpoints MongoDB e Amazon DocumentDB AWS DMS , suporta os seguintes tipos de dados para colunas que são chaves de partição para a opção de segmentação de intervalo de uma carga completa paralela.

- Double
- String
- ObjectId
- Inteiro de 32 bits
- Inteiro de 64 bits

A carga paralela para uso com regras de configuração de tabela é suportada para os seguintes endpoints de destino:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Amazon S3
- SAP Adaptive Server Enterprise (ASE)
- Amazon Redshift
- MongoDB (compatível somente com as opções de segmentação automática e com a segmentação por intervalo de uma carga máxima paralela)
- Amazon DocumentDB (compatível somente com as opções de segmentação automática e com a segmentação por intervalo de uma carga máxima paralela)
- Db2 LUW

Para especificar o número máximo de tabelas e visualizações para carga em paralelo, utilize a configuração de tarefa `MaxFullLoadSubTasks`.

Para especificar o número máximo de threads por tabela ou visualização para destinos compatíveis de uma tarefa de carga paralela, defina mais segmentos utilizando limites de valor-coluna.

> **Important**
>
> `MaxFullLoadSubTasks` controla o número de tabelas ou segmentos de tabela a serem carregados em paralelo. `ParallelLoadThreads` controla o número de threads utilizados por uma tarefa de migração para executar as cargas em paralelo. Essas configurações são multiplicativas. Dessa forma, o número total de threads utilizados durante uma tarefa de carga máxima é aproximadamente o resultado do valor de `ParallelLoadThreads` multiplicado pelo valor de `MaxFullLoadSubTasks` (`ParallelLoadThreads*MaxFullLoadSubtasks`).
> Se você criar tarefas com um grande número de subtarefas de carga máxima e um grande número de threads de carga paralela, a tarefa poderá consumir muita memória e falhar.

Para especificar o número máximo de threads por tabela para destinos do Amazon DynamoDB, do Amazon Kinesis Data Streams, do Apache Kafka ou do Amazon Elasticsearch Service, utilize a configuração da tarefa de metadados de destino `ParallelLoadThreads`.

Para especificar o tamanho do buffer para uma tarefa de carga paralela quando `ParallelLoadThreads` é utilizada, utilize a configuração da tarefa de metadados `ParallelLoadBufferSize`.

A disponibilidade e as configurações de `ParallelLoadThreads` e `ParallelLoadBufferSize` dependem do endpoint de destino.

Para obter mais informações sobre `ParallelLoadThreads` e `ParallelLoadBufferSize`, consulte Configurações de tarefa de metadados de destino. Para obter mais informações sobre a configuração de `MaxFullLoadSubTasks`, consulte Configurações de tarefa de carregamento completo. Para obter informações específicas de endpoints de destino, consulte os tópicos relacionados.

Para utilizar a carga paralela, crie uma regra de mapeamento de tabelas do tipo `table-settings` com a opção `parallel-load`. Na regra `table-settings`, é possível especificar os critérios de segmentação para uma única tabela, visualização ou coleção para carregamento em paralelo. Para fazer isso, defina o parâmetro `type` da opção `parallel-load` como uma das várias opções.

Como fazer isso depende de como você deseja segmentar a tabela, visualização ou coleção para carga paralela:

- Por partições (ou segmentos): carregue todas as partições da tabela ou da visualização existentes (ou segmentos) utilizando o tipo `partitions-auto`. Ou carregue somente as partições selecionadas utilizando o tipo `partitions-list` com uma matriz de partições especificada.

  Somente para endpoints do MongoDB e do Amazon DocumentDB, carregue todas as coleções ou as coleções especificadas por segmentos AWS DMS que calculam automaticamente também usando o tipo e os parâmetros opcionais adicionais. `partitions-auto` `table-settings`
- (Somente endpoints do Oracle) Por subpartições: carregue todas as subpartições da tabela ou da visualização existente utilizando o tipo `subpartitions-auto`. Ou carregue somente as subpartições selecionadas utilizando o tipo `partitions-list` com uma matriz de `subpartitions` especificada.
- Pelos segmentos que você define: carregue os segmentos da tabela, visualização ou coleção que você define utilizando limites de coluna-valor. Para fazer isso, utilize o tipo `ranges` com as matrizes `columns` e `boundaries` especificadas.

> **Note**
>
> Os endpoints do PostgreSQL são compatíveis somente com esse tipo de carga paralela. O MongoDB e o Amazon DocumentDB como endpoints de origem são compatíveis com esse tipo de segmentação por intervalo e com o tipo de segmentação automática de uma carga máxima paralela (`partitions-auto`).

Para identificar tabelas, visualizações ou coleções adicionais a serem carregadas em paralelo, especifique os objetos `table-settings` adicionais com as opções de `parallel-load`.

Nos procedimentos a seguir, é possível saber como codificar JSON para cada tipo de carga paralela, da mais simples à mais complexa.

Como especificar todas as partições de tabela, visualização ou coleção ou todas as subpartições de tabela ou visualização

- Especifique `parallel-load` com o tipo `partitions-auto` ou o tipo `subpartitions-auto` (mas não com os dois).

  Cada partição ou subpartição da tabela, visualização ou coleção é automaticamente alocada para seu próprio thread.

Em alguns endpoints, a carga paralela incluirá partições ou subpartições somente se elas já estiverem definidas para a tabela ou visualização. Para endpoints de origem do MongoDB e do Amazon DocumentDB, você pode AWS DMS calcular automaticamente as partições (ou segmentos) com base em parâmetros adicionais opcionais. Entre elas estão `number-of-partitions`, `collection-count-from-metadata`, `max-records-skip-per-page` e `batch-size`.

Como especificar partições, subpartições selecionadas da tabela ou da exibição ou ambas

1. Especifique `parallel-load` com o tipo `partitions-list`.
2. (Opcional) Inclua as partições, especificando uma matriz de nomes de partição como o valor de `partitions`.

   Cada partição especificada será então alocada para seu próprio thread.

   > **Important**
   >
   > Para endpoints do Oracle, verifique se as partições e subpartições não estão sobrepostas ao escolhê-las para a carga paralela. Se você utilizar partições e subpartições sobrepostas para carregar dados em paralelo, isso duplicará as entradas ou falhará devido a uma violação de duplicação da chave primária.

3. (Opcional) Somente para endpoints do Oracle inclua as subpartições especificando uma matriz de nomes de subpartições como o valor de `subpartitions`.

   Cada subpartição especificada será então alocada para seu próprio thread.

   > **Note**
   >
   > A carga paralela inclui partições ou subpartições somente se elas já estão definidas para a tabela ou a exibição.

É possível especificar os segmentos de tabela ou visualização como intervalos de valores de colunas. Ao fazer isso, lembre-se destas características da coluna:

- A especificação de colunas indexadas melhora significativamente o desempenho.

- É possível especificar até 10 colunas.
- Você não pode usar colunas para definir limites de segmentos com os seguintes tipos de AWS DMS dados: DOUBLE, FLOAT, BLOB, CLOB e NCLOB
- Os registros com valores nulos não são replicados.

Como especificar os segmentos de tabela, visualização ou coleção como intervalos de valores de colunas

1. Especifique `parallel-load` com o tipo `ranges`.
2. Defina um limite entre os segmentos da tabela ou da exibição, especificando uma matriz de nomes de coluna da tabela como o valor de `columns`. Faça isso para cada coluna para a qual você deseja definir um limite entre os segmentos da tabela ou da exibição.

   Observe que a ordem das colunas é significativa. A primeira coluna é a mais significativa e a última coluna é a menos significativa na definição de cada limite, conforme descrito a seguir.
3. Defina os intervalos de dados para todos os segmentos da tabela ou da exibição especificando uma matriz de limites como o valor de `boundaries`. Uma matriz de limites é uma matriz de matrizes de coluna/valor. Para fazer isso, siga as estas etapas:

   a. Especifique cada elemento de uma matriz de coluna/valor como um valor que corresponda a cada coluna. Uma matriz de coluna/valor representa o limite superior de cada segmento da tabela ou da exibição que você deseja definir. Especifique cada coluna na mesma ordem em que você especificou essa coluna na matriz `columns`.

      Insira valores para as colunas DATE no formato com suporte pela origem.

   b. Especifique cada matriz de valores de coluna como o limite superior, em ordem, de cada segmento da parte inferior até o next-to-top segmento da tabela ou exibição. Se houver linhas acima do limite superior especificado, essas linhas completarão o segmento superior da tabela ou da exibição. Portanto, o número de segmentos com base em intervalo será potencialmente mais um que o número de limites de segmentos na matriz de limites. Cada segmento com base em intervalo é alocado para seu próprio thread.

      Todos os dados não nulos serão replicados, mesmo que você não defina intervalos de dados para todas as colunas da tabela ou da exibição.

Por exemplo, suponha que você defina três matrizes de coluna-valor para as colunas COL1, COL2 e COL3, da seguinte forma.

COL1	COL2	COL3
10	30	105
20	20	120
100	12	99

Você definiu três limites de segmento para um total possível de quatro segmentos.

Para identificar os intervalos de linhas a serem replicados para cada segmento, a instância de replicação realiza uma pesquisa com essas três colunas para cada um dos quatro segmentos: A pesquisa é semelhante à seguinte:

Segmento 1

Replicar todas as linhas onde o seguinte é verdadeiro: os primeiros valores de duas colunas são menores ou iguais aos seus valores de limite superior do segmento 1 correspondentes. Além disso, os valores da terceira coluna são menores que o valor limite superior do segmento 1.

Segmento 2

Replique todas as linhas (exceto as linhas do segmento 1) em que o seguinte é verdadeiro: os primeiros valores de duas colunas são menores ou iguais aos valores de limite superior do segmento 2 correspondentes. Além disso, os valores da terceira coluna são menores que o valor limite superior do segmento 2.

Segmento 3

Replique todas as linhas (exceto as linhas do segmento 2) em que o seguinte é verdadeiro: os primeiros valores de duas colunas são menores ou iguais aos valores de limite superior do segmento 3 correspondentes. Além disso, os valores da terceira coluna são menores que o valor limite superior do segmento 3.

Segmento 4

Replicar todas as linhas restantes (exceto as linhas do Segmento 1, 2 e 3).

Nesse caso, a instância de replicação cria uma cláusula `WHERE` para carregar cada segmento da seguinte forma:

Segmento 1

```
((COL1 < 10) OR ((COL1 = 10) AND (COL2 < 30)) OR ((COL1 = 10) AND
(COL2 = 30) AND (COL3 < 105)))
```

Segmento 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)))
```

Segmento 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)))
```

Segmento 4

```
NOT ((COL1 < 100) OR ((COL1 = 100) AND (COL2 < 12)) OR ((COL1 = 100)
AND (COL2 = 12) AND (COL3 < 99)))
```

Especificar configurações de LOB para uma tabela ou exibição selecionada

É possível definir as configurações de LOB de tarefa para uma ou mais tabelas criando uma regra de mapeamento de tabela do tipo `table-settings` com a opção `lob-settings` para um ou mais objetos `table-settings`.

A especificação das configurações de LOB para tabelas ou exibições selecionadas tem suporte para os seguintes endpoints de origem:

- Oracle
- Microsoft SQL Server
- MySQL

- PostgreSQL
- IBM Db2, dependendo do `mode` e das configurações de `bulk-max-size`, como descrito a seguir
- SAP Adaptive Server Enterprise (ASE), de acordo com as configurações de `bulk-max-size` e `mode`, como descrito a seguir

A especificação das configurações de LOB para tabelas ou exibições selecionadas tem suporte para os seguintes endpoints de destino:

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- SAP ASE, dependendo do `mode` e das configurações de `bulk-max-size`, como descrito a seguir

> **Note**
>
> É possível utilizar tipos de dados de LOB somente com tabelas e visualizações que incluem uma chave primária.

Para utilizar as configurações de LOB para uma tabela ou visualização selecionada, crie uma regra de mapeamento de tabela do tipo `table-settings` com a opção `lob-settings`. Isso especifica o tratamento de LOB para a tabela ou a exibição identificada pela opção `object-locator`. Na regra `table-settings`, é possível especificar um objeto de `lob-settings` com os seguintes parâmetros:

- `mode`: especifica o mecanismo para tratar a migração de LOB para a tabela ou a visualização selecionada da seguinte forma:
  - `limited`: o modo LOB limitado padrão é o mais rápido e mais eficiente. Utilize esse modo somente se todos os seus LOBs forem pequenos (até 100 MB de tamanho) ou se o endpoint de destino não oferecer suporte a um tamanho de LOB ilimitado. Além disso, se você utilizar `limited`, todos os LOBs deverão estar dentro do tamanho que você definir para `bulk-max-size`.

Nesse modo para uma tarefa de carga máxima, a instância de replicação migra todos os LOBs em linha junto com outros tipos de dados de coluna como parte do armazenamento da tabela ou da exibição principal. No entanto, a instância trunca qualquer LOB maior que o valor de `bulk-max-size` para o tamanho especificado. Para uma tarefa de carregamento de captura de dados de alteração (CDC), a instância migra todos os LOBs usando uma pesquisa da tabela de origem, como no modo LOB completo padrão.

> **Note**
>
> Você pode migrar exibições somente para tarefas de carregamento completo.

- `unlimited`: o mecanismo de migração para o modo LOB completo depende do valor definido para `bulk-max-size` da seguinte forma:
  - Modo LOB completo padrão: quando você define `bulk-max-size` como zero, a instância de replicação migra todos os LOBs utilizando o modo LOB completo padrão. Esse modo exige uma pesquisa na tabela ou na exibição de origem para migrar cada LOB, independentemente do tamanho. Isso normalmente resulta em uma migração muito mais lenta do que no modo LOB limitado. Utilize esse modo somente se todos ou a maioria dos LOBs forem grandes (1 GB ou maior).
  - Combinação do modo LOB completo: quando você define `bulk-max-size` como um valor diferente de zero, esse modo LOB completo utiliza uma combinação de modo LOB limitado e do modo LOB completo padrão. Isso é para uma tarefa de carga máxima, se o tamanho de um LOB estiver dentro de seu valor de `bulk-max-size`, a instância migrará o LOB em linha como no modo LOB limitado. Se o tamanho do LOB for maior que esse valor, a instância migrará o LOB utilizando uma pesquisa da tabela ou da exibição de origem como no modo LOB completo padrão. Para uma tarefa de carregamento de captura de dados de alteração (CDC), a instância migra todos os LOBs usando uma pesquisa da tabela de origem, como no modo LOB completo padrão. Ela faz isso independentemente do tamanho do LOB.

    > **Note**
    >
    > Você pode migrar exibições somente para tarefas de carregamento completo.

    Esse modo resulta em uma velocidade de migração que é um compromisso entre o mais rápido, modo LOB limitado, e o mais lento, modo LOB completo padrão. Utilize esse modo

somente quando você tiver uma mistura de LOBs pequenos e grandes, e a maioria dos LOBs forem pequenos.

Essa combinação de modo LOB completo está disponível somente para os seguintes endpoints:

- IBM Db2 como origem
- SAP ASE como origem ou destino

Independentemente de como você especifica o modo `unlimited`, a instância migrará todos os LOBs completamente, sem truncamento.

- `none`: a instância de replicação migra LOBs na tabela ou visualização selecionada utilizando as configurações de LOB da tarefa. Utilize essa opção para ajudar a comparar os resultados de migração com e sem configurações de LOB para a tabela ou a exibição selecionada.

Se a tabela ou a exibição especificada tiver LOBs incluídos na replicação, é possível definir a configuração da tarefa `BatchApplyEnabled` `true` somente quando utilizar o modo LOB `limited`.

Em alguns casos, é possível definir `BatchApplyEnabled` como `true` e `BatchApplyPreserveTransaction` como `false`. Nesses casos, a instância definirá `BatchApplyPreserveTransaction` como `true` se a tabela ou a exibição tiver LOBs e os endpoints de origem e de destino forem Oracle.

- `bulk-max-size`: defina esse valor como zero ou como um valor diferente de zero em quilobytes, dependendo do `mode`, conforme descrito nos itens anteriores. No modo `limited`, você deve definir um valor diferente para esse parâmetro.

  A instância converte LOBs em formato binário. Portanto, para especificar o maior LOB que você precisa replicar, multiplique seu tamanho por três. Por exemplo, se o seu maior LOB for de 2 MB, defina `bulk-max-size` como 6.000 (6 MB).

Exemplos de configurações de tabela

Veja a seguir alguns exemplos que demonstram a utilização de configurações de tabela.

Example Carregar uma tabela segmentada por partições

O exemplo a seguir carrega uma tabela `SALES` em sua origem de forma mais eficiente carregando-a em paralelo com base em todas as partições.

```
{
   "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 Carregar uma tabela segmentada por subpartições

O exemplo a seguir carrega uma tabela SALES em sua origem do Oracle de forma mais eficiente carregando-a em paralelo com base em todas as suas subpartições.

```
{
   "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 Carregar uma tabela segmentada por uma lista de partições

O exemplo a seguir carrega uma tabela SALES em sua origem carregando-a em paralelo por uma lista específica de partições. Aqui, as partições especificadas são nomeadas de acordo com valores começando com partes do alfabeto, por exemplo, ABCD, EFGH e assim por diante.

```
{
    "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 Carregar uma tabela segmentada do Oracle por uma lista selecionada de partições e subpartições

O exemplo a seguir carrega uma tabela SALES na origem do Oracle carregando-a em paralelo por uma lista selecionada de partições e subpartições. Aqui, as partições especificadas são nomeadas de acordo com valores começando com partes do alfabeto, por exemplo, ABCD, EFGH e assim por diante. As subpartições especificadas são nomeadas de acordo com os valores começando com numerais, por exemplo, 01234 e 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 Carregar uma tabela segmentada por intervalos de valores de coluna

O exemplo a seguir carrega uma tabela SALES em sua origem, carregando-a em paralelo por segmentos especificados por intervalos dos valores das colunas SALES_NO e REGION.

```
{
    "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "HR",
                "table-name": "SALES"
            },
            "parallel-load": {
                "type": "ranges",
                "columns": [
                    "SALES_NO",
```

```
                    "REGION"
                ],
                "boundaries": [
                    [
                        "1000",
                        "NORTH"
                    ],
                    [
                        "3000",
                        "WEST"
                    ]
                ]
            }
        }
    ]
}
```

Aqui, duas colunas são especificadas para os intervalos dos segmentos com os nomes `SALES_NO` e `REGION`. Dois limites são especificados com dois conjuntos de valores de colunas (`["1000","NORTH"]` e `["3000","WEST"]`).

Esses dois limites, portanto, identificam os seguintes três segmentos da tabela a serem carregados em paralelo:

Segmento 1

As linhas com `SALES_NO` menor ou igual a 1.000 e `REGION` menor que "NORTH". Em outras palavras, o números de vendas até 1.000 na região EAST.

Segmento 2

As linhas que não são do segmento 1 com `SALES_NO` menor ou igual a 3.000 e `REGION` menor que "WEST". Em outras palavras, os números de vendas acima de 1.000 até 3.000 nas regiões NORTH e SOUTH.

Segmento 3

Todas as demais linhas que não sejam do Segmento 1 e do Segmento 2. Em outras palavras, os números de vendas acima de 3.000 na região "WEST".

Example Carregar duas tabelas: uma segmentada por intervalos e outra segmentada por partições

O exemplo a seguir carrega uma tabela SALES em paralelo com os limites do segmento que você identificar. Ele também carrega uma tabela ORDERS em paralelo por todas as suas partições, como nos exemplos anteriores.

```
{
    "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 Carregar uma tabela com LOBs utilizando o modo LOB limitado

O exemplo a seguir carrega uma tabela `ITEMS` que inclui LOBs em sua origem utilizando o modo LOB limitado (o padrão) com um tamanho máximo não truncado de 100 MB. Todos os LOBs maiores que esse tamanho são truncados em 100 MB. Todos os LOBs são carregados em linha com todos os outros tipos de dados da coluna.

```
{
   "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 Carregar uma tabela com LOBs utilizando o modo LOB completo padrão

O exemplo a seguir carrega uma tabela `ITEMS` na origem, incluindo todos os LOBs sem truncamento, utilizando o modo LOB completo padrão. Todos os LOBs, independentemente do tamanho, são carregados separadamente dos outros tipos de dados utilizando uma pesquisa de cada LOB na tabela de origem.

```
{
   "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 Carregar uma tabela com LOBs utilizando o modo de combinação de LOB completo

O exemplo a seguir carrega uma tabela `ITEMS` em sua origem, inclusive todos os seus LOBs sem truncamento, utilizando o modo de combinação de LOB completo. Todos os LOBs com até 100 MB de tamanho são carregados em linha junto com os outros tipos de dados, como no modo LOB limitado. Todos os LOBs com mais de 100 MB de tamanho são carregados separadamente de outros tipos de dados. Essa carga separada utiliza uma pesquisa para cada LOB na tabela de origem, como no modo LOB completo padrão.

```
{
   "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 Carregar uma tabela com LOBs utilizando as configurações de LOB de tarefa

O exemplo a seguir carrega uma tabela `ITEMS` em sua origem, inclusive todos os LOBs, utilizando suas configurações de LOB de tarefa. A configuração de `bulk-max-size` de 100 MB é ignorada e abandonada apenas se houver uma redefinição rápida para o modo `limited` ou `unlimited`.

```
{
   "rules": [{
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "%",
                "table-name": "%"
            },
            "rule-action": "include"
        },
        {
            "rule-type": "table-settings",
            "rule-id": "2",
            "rule-name": "2",
            "object-locator": {
                "schema-name": "INV",
                "table-name": "ITEMS"
            },
            "lob-settings": {
                "mode": "none",
                "bulk-max-size": "100000"
            }
        }
     ]
}
```

Usar filtros de origem

É possível usar filtros de origem para limitar o número e o tipo de registros transferidos da origem ao destino. Por exemplo, você pode especificar que somente os funcionários localizados na sede serão movidos para o banco de dados de destino. Os filtros fazem parte de uma regra de seleção. Os filtros são aplicados a uma coluna de dados.

Os filtros de origem devem seguir estas restrições:

- Uma regra de seleção pode não ter filtros ou ter um ou mais filtros.
- Cada filtro pode ter uma ou mais condições de filtro.
- Se for utilizado mais de um filtro, a lista de filtros será combinada como se utilizasse um operador AND entre eles.

- Se for usada mais de uma condição de filtro em um único filtro, a lista de condições de filtro será combinada como se usasse um operador OR entre elas.
- Os filtros só são aplicados quando `rule-action = 'include'`.
- Os filtros exigem um nome de coluna e uma lista de condições de filtro. As condições do filtro devem ter um operador de filtro associado a um valor, dois valores ou nenhum valor, dependendo do operador.
- Os nomes de colunas, tabelas, exibições e de esquema diferenciam maiúsculas e minúsculas. O Oracle e o Db2 devem sempre utilizar letras maiúsculas.
- Os filtros só são compatíveis com tabelas com nomes exatos. Os filtros não são compatíveis com curingas.

As seguintes limitações se aplicam ao uso de filtros de origem:

- Os filtros não calculam colunas de right-to-left idiomas.
- Não aplique filtros a colunas de LOB.
- Aplique filtros somente a colunas imutáveis, que não são atualizadas após a criação. Se os filtros de origem forem aplicados a colunas mutáveis, que podem ser atualizadas após a criação, o resultado pode ser um comportamento adverso.

  Por exemplo, um filtro para excluir ou incluir linhas específicas em uma coluna sempre excluirá ou incluirá as linhas especificadas, mesmo que as linhas sejam alteradas posteriormente. Vamos supor que você exclua ou inclua as linhas 1 a 10 na coluna A, e elas sejam posteriormente alteradas de modo a tornarem-se as linhas 11 a 20. Neste caso, elas continuarão a ser excluídas ou incluídas ainda que os dados não sejam mais os mesmos.

  Da mesma forma, vamos supor que uma linha fora do escopo do filtro seja atualizada posteriormente (ou atualizada e excluída) e deva ser excluída ou incluída conforme definido pelo filtro. Neste caso, ela será replicada no destino.

As seguintes preocupações adicionais se aplicam ao usar filtros de origem:

- Recomendamos que você crie um índice usando as colunas incluídas na definição de filtragem e a chave primária.

Criar regras de filtro de origem em JSON

É possível criar filtros de origem utilizando o parâmetro `filters` do JSON de uma regra de seleção. O parâmetro `filters` especifica uma matriz de um ou mais objetos JSON. Cada objeto tem parâmetros que especificam o tipo do filtro de origem, o nome da coluna e as condições de filtragem. Essas condições de filtro incluem um ou mais operadores de filtro e valores de filtro.

A tabela a seguir mostra os parâmetros utilizados para especificar a filtragem de origem em um objeto `filters`.

Parâmetro	Valor
`filter-type`	`source`
`column-name`	Um parâmetro com o nome da coluna de origem à qual você deseja aplicar o filtro. O nome diferencia maiúsculas e minúsculas.
`filter-conditions`	Uma matriz de um ou mais objetos que contém um parâmetro `filter-operator` e zero ou mais parâmetros de valor associado, dependendo do valor de `filter-operator` .
`filter-operator`	Um parâmetro com um dos seguintes valores: • `lte`: menor ou igual a um valor • `ste`: menor ou igual a um valor (alias `lte`) • `gte`: maior ou igual a um valor • `eq`: igual a um valor • `noteq`: diferente de um valor • `between`: igual a ou entre dois valores • `notbetween` : diferente de ou entre dois valores • `null`: valores `NULL` • `notnull`: sem valores `NULL`
`value` ou `start-value` e `end-value` ou	Zero ou mais parâmetros de valor associado a `filter-operator` :

Parâmetro	Valor
sem valores	• Se `filter-operator` for `lte`, `ste`, `gte`, `eq` ou `noteq`, utilize um valor `value` para especificar um parâmetro de valor. • Se `filter-operator` for `between` ou `notbetween`, utilize `start-value` e `end-value` para especificar dois parâmetros de valor. • Se `filter-operator` for `null` ou `notnull`, não especifique nenhum parâmetro de valor.

Os exemplos a seguir mostram algumas maneiras comuns de utilizar filtros de origem.

Example Filtro único

O seguinte filtro replica todos os funcionários em que `empid >= 100` para o banco de dados de destino.

```
{
    "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 Vários operadores de filtro

O seguinte filtro aplica vários operadores de filtro a uma única coluna de dados. O filtro replica todos os funcionários em que `(empid <= 10)` OU `(empid is between 50 and 75)` OU `(empid >= 100)` para o banco de dados de destino.

```
{
    "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 Vários filtros

O filtro a seguir aplica vários filtros a duas colunas em uma tabela. O filtro replica todos os funcionários em que `(empid <= 100)` AND `(dept = tech)` para o banco de dados de destino.

```
{
```

```
    "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 Filtrar valores NULL

O filtro a seguir mostra como filtrar valores vazios. Ele replica todos os funcionários em que `dept = NULL` para o banco de dados de destino.

```
{
    "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 Filtrar utilizando operadores NOT

Alguns dos operadores podem ser utilizados na forma negativa. O seguinte filtro replica todos os funcionários em que `(empid is < 50) OR (empid is > 75)` para o banco de dados de destino.

```
{
    "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 Utilizar operadores de filtros mistos

A partir da AWS DMS versão 3.5.0, você pode misturar operadores inclusivos e operadores negativos.

O seguinte filtro replica todos os funcionários em que `(empid != 50) AND (dept is not NULL)` para o banco de dados de destino.

```
{
    "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"
            }]
        }]
    }]
}
```

Observe o seguinte ao utilizar `null` com outros operadores de filtro:

- Utilizar condições de filtro inclusivas, negativas e `null` no mesmo filtro não replicará registros com valores `NULL`.

- Utilizar condições de filtro negativas e `null` em conjunto sem condições de filtro inclusivas no mesmo filtro não replicará nenhum dado.
- Utilizar condições de filtro negativas sem uma condição de filtro `null` definida explicitamente não replicará registros com valores `NULL`.

Filtragem por hora e data

Ao selecionar dados a serem importados, você pode especificar uma data ou hora como parte dos seus critérios de filtro. AWS DMS usa o formato de data YYYY-MM-DD e o formato de hora YYYY-MM-DD HH:MM:SS para filtragem. As funções AWS DMS de comparação seguem as convenções do SQLite. Para obter mais informações sobre comparações de datas e tipos de dados SQLite, consulte Datatypes In SQLite Version 3 na documentação do SQLite.

O exemplo a seguir mostra como filtrar por uma data. Ele replica todos os funcionários em que `empstartdate >= January 1, 2002` para o banco de dados de destino.

Example Filtro de data única

```
{
    "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"
            }]
        }]
    }]
}
```

Ativar e trabalhar com avaliações de pré-migração de uma tarefa

A avaliação de pré-migração avalia os componentes especificados de uma tarefa de migração de banco de dados para ajudar a identificar quaisquer problemas que possam impedir que uma tarefa de migração seja executada conforme o esperado. Essa avaliação dá a você a chance de identificar e corrigir problemas antes de executar uma tarefa nova ou modificada. Isso permite que você evite atrasos relacionados a falhas de tarefas causadas por requisitos ausentes ou limitações conhecidas.

AWS DMS fornece acesso a duas opções diferentes para avaliações de pré-migração:

- Avaliação do tipo de dados: um relatório antigo que fornece um escopo limitado de avaliações.
- Execução da avaliação pré-migração: contém vários tipos de avaliações individuais, incluindo resultados da avaliação do tipo de dados.

> **Note**
>
> Se você escolher uma execução de avaliação pré-migração, não precisará escolher uma avaliação de tipo de dados separadamente.

Essas opções estão descritas nos tópicos a seguir:

- Especificar, iniciar e visualizar as execuções de avaliação de pré-migração: uma execução de avaliação de pré-migração (recomendada) especifica uma ou mais avaliações individuais a serem executadas com base em uma configuração de tarefa de migração nova ou existente. Cada avaliação individual avalia um elemento específico de um banco de dados de origem e/ou destino suportado do ponto de vista de critérios como tipo de migração, objetos suportados, configuração de índice e outras configurações de tarefas, como mapeamentos de tabelas que identificam os esquemas e tabelas a serem migrados.

  Por exemplo, uma avaliação individual pode avaliar quais tipos de dados de origem ou formatos de chave primária podem ou não ser migrados, possivelmente com base na versão do AWS DMS mecanismo. Você pode iniciar e visualizar os resultados da última avaliação e visualizar os resultados de todas as execuções de avaliação anteriores de uma tarefa usando o AWS DMS Management Console ou usando os SDKs AWS CLI e para acessar a AWS DMS API. Você também pode visualizar os resultados de avaliações anteriores de uma tarefa em um bucket do Amazon S3 que você selecionou AWS DMS para armazenar esses resultados.

> **Note**
>
> O número e os tipos de avaliações individuais disponíveis podem aumentar com o tempo. Para obter mais informações sobre atualizações periódicas, consulte Especificar avaliações individuais.

- Iniciando e visualizando avaliações de tipo de dados (Legacy): uma avaliação de tipo de dados (legado) retorna os resultados de um único tipo de avaliação de pré-migração em uma única estrutura JSON: os tipos de dados que podem não ter sido migrados corretamente em uma instância de banco de dados de origem relacional compatível. Esse relatório retorna os resultados de todos os tipos de dados problemáticos encontrados em cada esquema e tabela no banco de dados de origem selecionado para migração.

Criação de pré-requisitos para avaliações de pré-migração

Esta seção descreve os recursos do Amazon S3 e do IAM necessários para criar uma avaliação de pré-migração.

Crie um bucket do S3

AWS DMS armazena relatórios de avaliação de pré-migração em um bucket S3. Para criar o bucket do S3, faça o seguinte:

1. Faça login no AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Selecione Criar bucket.
3. *Na página **Criar bucket**, insira um nome globalmente exclusivo que inclua seu nome de login para o bucket, como dms-bucket- yoursignin.*
4. Escolha o Região da AWS para a tarefa de migração do DMS.
5. Deixe as configurações restantes como estão e escolha Criar bucket.

Criar recursos do IAM

O DMS usa uma função e uma política do IAM para acessar o bucket do S3 e armazenar os resultados da avaliação de pré-migração.

Para criar a política do IAM, faça o seguinte:

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies.
3. Escolha Create policy.
4. Na página Criar política, escolha a guia JSON.
5. Cole o seguinte código JSON no editor, substituindo o código de exemplo. Substitua *my-bucket* pelo nome do bucket do Amazon S3 que você criou na seção anterior.

```
{
   "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. Escolha Avançar: Tags e, em seguida, escolha Avançar: Revisão.
7. Insira **`DMSPremigrationAssessmentS3Policy`** em Nome* e escolha Criar política.

Para criar a função do IAM, faça o seguinte:

1. No console do IAM, selecione Perfis no painel de navegação.
2. Selecione Criar função.
3. Na página Selecionar entidade confiável, em Tipo de entidade confiável, escolha Serviço da AWS . Para Casos de uso de outros AWS serviços, escolha DMS.
4. Marque a caixa de seleção DMS e escolha Avançar.
5. Na página Adicionar permissões, escolha DMS PremigrationAssessment S3Policy. Escolha Próximo.
6. Na página Nomear, revisar e criar, insira **`DMSPremigrationAssessmentS3Role`** em Nome do perfil e escolha Criar função.
7. Na página Perfis, insira **`DMSPremigrationAssessmentS3Role`** em Nome do perfil. Escolha DMS PremigrationAssessment S3Role.
8. Na página DMS PremigrationAssessment S3Role, escolha a guia Relações de confiança. Escolha Editar política de confiança.
9. Na página Editar política de confiança, cole o seguinte JSON no editor, substituindo o texto existente.

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

   Essa política concede `sts:AssumeRole` permissão ao DMS para colocar os resultados da execução da avaliação de pré-migração no bucket do S3.
10. Escolha Atualizar política.

Especificar, iniciar e visualizar as execuções de avaliação de pré-migração

Uma avaliação pré-migração especifica uma ou mais avaliações individuais a serem executadas com base em uma configuração de tarefa de migração nova ou existente. Cada avaliação individual avalia um elemento específico do banco de dados relacional de origem ou de destino, dependendo de considerações como o tipo de migração, os objetos compatíveis, a configuração de índice e outras configurações de tarefas, como mapeamentos de tabelas que identificam os esquemas e tabelas a serem migrados. Por exemplo, uma avaliação individual pode avaliar quais tipos de dados de origem ou formatos de chave primária podem ou não ser migrados.

Especificar avaliações individuais

Ao criar uma nova execução de avaliação, você pode optar por executar algumas ou todas as avaliações individuais aplicáveis à configuração da sua tarefa.

AWS DMS suporta execuções de avaliação de pré-migração para os seguintes mecanismos de banco de dados relacionais de origem e destino:

- Avaliações da Oracle
- Avaliações do SQL Server
- Avaliações do MySQL(inclui MariaDB e Amazon Aurora MySQL, edição compatível com o MariaDB)
- Avaliações do PostgreSQL(inclui a edição compatível com Amazon Aurora PostgreSQL)

Iniciar e visualizar execuções de avaliação de pré-migração

Você pode iniciar uma execução de avaliação de pré-migração para uma tarefa de migração nova ou existente usando o AWS DMS Management Console AWS CLI, o e a AWS DMS API.

Para iniciar uma avaliação de pré-migração, execute uma tarefa nova ou existente

1. Na página tarefas de migração de banco de dados no console de gerenciamento do AWS DMS , execute um destes procedimentos:

   - Para criar uma nova tarefa e avaliá-la, escolha Criar tarefa. A página Criar tarefa de migração de banco de dados é aberta.
     1. Insira as configurações de tarefa necessárias para criar a tarefa, incluindo o mapeamento de tabela.

2. Na seção Avaliação de pré-migração, a caixa de seleção Executar avaliação de pré-migração está marcada. Essa página contém as opções para especificar uma execução de avaliação para a nova tarefa.

> **Note**
>
> Ao criar uma tarefa, a ativação da execução de uma avaliação de pré-migração desativa a opção de iniciar a tarefa automaticamente na criação da tarefa. É possível iniciar a tarefa manualmente após a conclusão da avaliação.

- Para avaliar uma tarefa existente, escolha o Identificador de uma tarefa existente na página Tarefas de migração do banco de dados. A página da tarefa existente escolhida é aberta:
  1. Escolha Ações e selecione Criar avaliação de pré-migração. A página Criar avaliação de pré-migração é aberta com opções para especificar uma execução de avaliação da tarefa existente.

2. Insira um nome exclusivo para sua execução de avaliação ou deixe o valor padrão.
3. Selecione as avaliações individuais disponíveis que você quer incluir nessa execução de avaliação. Você só pode selecionar as avaliações individuais disponíveis com base nas configurações da tarefa atual. Por padrão, todas as avaliações individuais disponíveis são habilitadas e selecionadas.
4. Pesquise e escolha um bucket e uma pasta do Amazon S3 na sua conta para armazenar o relatório de resultados da avaliação. Para obter informações sobre como configurar recursos para execuções de avaliação, consulteCriação de pré-requisitos para avaliações de pré-migração.
5. Selecione ou insira um perfil do IAM com acesso total da conta ao bucket e pasta do Amazon S3 escolhidos. Para obter informações sobre como configurar recursos para execuções de avaliação, consulteCriação de pré-requisitos para avaliações de pré-migração.
6. Opcionalmente, escolha uma configuração para criptografar o relatório de resultados da avaliação no bucket do Amazon S3. Para obter informações sobre a criptografia de bucket do S3, consulte Configuração do comportamento padrão de criptografia do lado do servidor para buckets do Amazon S3.
7. Escolha Criar tarefa para uma nova tarefa ou escolha Criar para uma tarefa existente.

   A página Tarefas de migração do banco de dados é aberta listando a tarefa nova ou modificada com o Status de Criando... e uma mensagem de banner indicando que a execução da avaliação de pré-migração começará assim que a tarefa for criada.

AWS DMS fornece acesso às últimas e a todas as execuções anteriores de avaliação de pré-migração usando o AWS DMS Management Console AWS CLI, o ou a AWS DMS API.

Para visualizar os resultados da execução da avaliação

1. No AWS DMS Management Console, escolha o Identificador para sua tarefa existente na página Tarefas de migração do banco de dados. A página da tarefa existente é aberta.
2. Escolha a guia Avaliações de pré-migração na página da tarefa existente. Isso abre uma seção de avaliações de pré-migração nessa página mostrando os resultados das execuções de avaliação, listados por nome, em ordem cronológica inversa. O resultado mais recente aparece no topo da lista. Escolha o nome da execução da avaliação cujos resultados você deseja visualizar.

Esses resultados da execução da avaliação começam com o nome da avaliação mais recente e uma visão geral de seu status, seguida por uma lista das avaliações individuais especificadas e seu status. É possível explorar os detalhes do status de cada avaliação individual escolhendo o nome na lista, com os resultados disponíveis até o nível da coluna da tabela.

Tanto a visão geral do status de uma execução de avaliação quanto de cada avaliação individual mostram um valor de Status. Esse valor indica o status geral da execução da avaliação e um status semelhante para cada avaliação individual. Veja a seguir uma lista dos valores de Status da execução da avaliação:

- `"cancelling"`: a execução da avaliação foi cancelada.
- `"deleting"`: a execução da avaliação foi excluída.
- `"failed"`: pelo menos uma avaliação individual concluída com status `failed`.
- `"error-provisioning"`: ocorreu um erro interno enquanto os recursos eram provisionados (durante o status `provisioning`).
- `"error-executing"`: ocorreu um erro interno durante a execução de avaliações individuais (durante o status `running`).
- `"invalid state"`: a avaliação está em um estado desconhecido.
- `"passed"`: todas as avaliações individuais foram concluídas e nenhuma tem o status `failed`.
- `"provisioning"`: os recursos necessários para executar avaliações individuais estão sendo provisionados.
- `"running"`: avaliações individuais estão sendo executadas.

- `"starting"`: a execução da avaliação está começando, mas os recursos ainda não estão sendo provisionados para avaliações individuais.
- `"warning"`: pelo menos uma avaliação individual concluída com status `warning`.

Veja a seguir uma lista dos valores de Status de cada avaliação individual da execução de avaliação:

- `"cancelled"`: a avaliação individual foi cancelada como parte do cancelamento da execução da avaliação.
- `"error"`: a avaliação individual não foi concluída com sucesso.
- `"failed"`: a avaliação individual foi concluída com sucesso com um resultado de validação reprovado: veja os detalhes do resultado para obter mais informações.
- `"invalid state"`: a avaliação individual está em estado desconhecido.
- `"passed"`: a avaliação individual foi concluída com um resultado da validação bem-sucedido.
- `"pending"`: a avaliação individual está aguardando para ser executada.
- `"running"`: a avaliação individual está em andamento.
- `"warning"`: a avaliação individual foi concluída com sucesso com um resultado de validação de aviso: visualize os detalhes do resultado para obter mais informações.

Também é possível visualizar os arquivos JSON dos resultados da execução da avaliação no Amazon S3.

Como visualizar os arquivos JSON da execução da avaliação no Amazon S3

1. No AWS DMS Management Console, escolha o link do bucket do Amazon S3 mostrado na visão geral do status da execução da avaliação. Isso exibe uma lista de pastas de bucket e outros objetos do Amazon S3 armazenados no bucket. Se os resultados estiverem armazenados em uma pasta de bucket, abra a pasta.
2. É possível encontrar os resultados da execução da avaliação em vários arquivos JSON. Um arquivo `summary.json` contém os resultados gerais da execução da avaliação. Cada um dos arquivos restantes são nomeados para uma avaliação individual que foi especificada para a execução da avaliação, como `unsupported-data-types-in-source.json`. Cada um desses arquivos contém os resultados da avaliação individual correspondente da execução de avaliação escolhida.

Para iniciar e visualizar os resultados das execuções de avaliação de pré-migração para uma tarefa de migração existente, você pode executar os seguintes comandos de CLI AWS DMS e operações de API:

- CLI: `describe-applicable-individual-assessments`, API: `DescribeApplicableIndividualAssessments`: fornece uma lista de avaliações individuais que é possível especificar para uma nova execução de avaliação de pré-migração, considerando um ou mais parâmetros de configuração da tarefa.
- CLI: `start-replication-task-assessment-run`, API: `StartReplicationTaskAssessmentRun`: inicia uma nova execução de avaliação de pré-migração de uma ou mais avaliações individuais de uma tarefa de migração existente.
- CLI: `describe-replication-task-assessment-runs`, API: `DescribeReplicationTaskAssessmentRuns`: retorna uma lista paginada de execuções de avaliação de pré-migração com base nas configurações de filtro.
- CLI: `describe-replication-task-individual-assessments`, API: `DescribeReplicationTaskIndividualAssessments`: retorna uma lista paginada de avaliações individuais com base nas configurações de filtro.
- CLI: `cancel-replication-task-assessment-run`, API: `CancelReplicationTaskAssessmentRun`: cancela, mas não exclui, uma única execução de avaliação de pré-migração.
- CLI: `delete-replication-task-assessment-run`, API: `DeleteReplicationTaskAssessmentRun`: exclui o registro de uma única execução de avaliação de pré-migração.

Avaliações individuais

Esta seção descreve as avaliações individuais de pré-migração.

Para criar uma avaliação individual de pré-migração usando a AWS DMS API, use a chave de API listada para o `IncludeOnly` parâmetro da StartReplicationTaskAssessmentRunação.

Tópicos

- Avaliações para todos os tipos de endpoints
- Avaliações da Oracle
- Avaliações do SQL Server

- Avaliações do MySQL
- Avaliações do MariaDB
- Avaliações do PostgreSQL

Avaliações para todos os tipos de endpoints

Esta seção descreve avaliações individuais de pré-migração para todos os tipos de endpoints.

Tópicos

- Tipos de dados incompatíveis
- Objetos grandes (LOBs) são usados, mas as colunas LOB de destino não são anuláveis
- Tabela de origem com objetos grandes (LOBs), mas sem chaves primárias ou restrições exclusivas
- Tabela de origem sem chave primária para CDC ou carga total e somente tarefas CDC
- Tabela de destino sem chaves primárias somente para tarefas do CDC
- Tipos de chave primária de origem não suportados - chaves primárias compostas

Tipos de dados incompatíveis

Chave da API: `unsupported-data-types-in-source`

Verifica os tipos de dados no endpoint de origem que o DMS não suporta. Nem todos os tipos de dados podem ser migrados entre os mecanismos.

Objetos grandes (LOBs) são usados, mas as colunas LOB de destino não são anuláveis

Chave da API: `full-lob-not-nullable-at-target`

Verifica a nulidade de uma coluna LOB no destino quando a replicação usa o modo LOB completo ou o modo LOB embutido. O DMS exige que uma coluna LOB seja nula ao usar esses modos LOB. Essa avaliação exige que os bancos de dados de origem e destino sejam relacionais.

Tabela de origem com objetos grandes (LOBs), mas sem chaves primárias ou restrições exclusivas

Chave da API: `table-with-lob-but-without-primary-key-or-unique-constraint`

Verifica a presença de tabelas de origem com LOBs, mas sem uma chave primária ou uma chave exclusiva. Uma tabela deve ter uma chave primária ou uma chave exclusiva para que o DMS migre LOBs. Essa avaliação exige que o banco de dados de origem seja relacional.

Tabela de origem sem chave primária para CDC ou carga total e somente tarefas CDC

Chave da API: `table-with-no-primary-key-or-unique-constraint`

Verifica a presença de uma chave primária ou de uma chave exclusiva nas tabelas de origem para uma migração de carga completa e captura de dados de alteração (CDC) ou uma migração somente de CDC. A falta de uma chave primária ou de uma chave exclusiva pode causar problemas de desempenho durante a migração do CDC. Essa avaliação exige que o banco de dados de origem seja relacional e que o tipo de migração inclua o CDC.

Tabela de destino sem chaves primárias somente para tarefas do CDC

Chave da API: `target-table-has-unique-key-or-primary-key-for-cdc`

Verifica a presença de uma chave primária ou de uma chave exclusiva em tabelas de destino já criadas para uma migração somente de CDC. A falta de uma chave primária ou de uma chave exclusiva pode causar varreduras completas da tabela no destino quando o DMS aplica atualizações e exclusões. Isso pode resultar em problemas de desempenho durante a migração do CDC. Essa avaliação exige que o banco de dados de destino seja relacional e que o tipo de migração inclua o CDC.

Tipos de chave primária de origem não suportados - chaves primárias compostas

Chave da API: `unsupported-source-pk-type-for-elasticsearch-target`

Verifica a presença de chaves primárias compostas nas tabelas de origem ao migrar para o Amazon OpenSearch Service. A chave primária da tabela de origem deve ser composta de uma só coluna. Essa avaliação exige que o banco de dados de origem seja relacional e o banco de dados de destino seja o DynamoDB.

> **Note**
>
> O DMS oferece suporte à migração de um banco de dados de origem para um destino OpenSearch de serviço em que a chave primária de origem consiste em várias colunas.

Avaliações da Oracle

Esta seção descreve avaliações de pré-migração individuais para tarefas de migração que utilizam um endpoint de origem do Oracle.

> **Note**
>
> Para utilizar as avaliações de pré-migração dessa seção, adicione as seguintes permissões ao `dms_user`:
>
> ```
> grant select on gv_$parameter to dms_user;
> grant select on v_$instance to dms_user;
> grant select on v_$version to dms_user;
> grant select on gv_$ASM_DISKGROUP to dms_user;
> grant select on gv_$database to dms_user;
> grant select on DBA_DB_LINKS to to dms_user;
> grant select on gv_$log_History to dms_user;
> grant select on gv_$log to dms_user;
> grant select on dba_types to dms_user;
> grant select on dba_users to dms_user;
> grant select on dba_directories to dms_user;
> ```
>
> Para obter mais informações sobre permissões ao utilizar o Oracle como origem, consulte Privilégios de conta de usuário necessários em uma fonte Oracle autogerenciada para AWS DMS.

Tópicos

- Verificar o registro em log suplementar no nível do banco de dados
- Valide se o link para o banco de dados necessário foi criado para espera
- Validação Oracle para o tipo de dados LOB e se o leitor binário estiver configurado
- Validar se o banco de dados é CDB
- Confira o Oracle Database Edition
- Validar o método CDC do DMS para Oracle
- Validar a configuração do Oracle RAC para o DMS
- Valide se o usuário do DMS tem permissões no destino
- Valide se o registro suplementar é necessário para todas as colunas
- Valide se o registro suplementar está ativado em tabelas com chaves primárias ou exclusivas
- Valide se há SecureFile LOBs e se a tarefa está configurada para o modo LOB completo
- Valide se os índices baseados em funções estão sendo usados nas tabelas incluídas no escopo da tarefa.

- Valide se tabelas temporárias globais estão sendo usadas nas tabelas incluídas no escopo da tarefa.
- Valide se tabelas organizadas por índice com um segmento de estouro estão sendo usadas nas tabelas incluídas no escopo da tarefa.
- Valide se tabelas de aninhamento de vários níveis são usadas nas tabelas incluídas no escopo da tarefa.
- Valide se colunas invisíveis são usadas nas tabelas incluídas no escopo da tarefa.
- Valide se visualizações materializadas baseadas em uma coluna ROWID são usadas nas tabelas incluídas no escopo da tarefa.
- Valide se o recurso Active Data Guard DML Redirect é usado.
- Valide se tabelas particionadas híbridas são usadas.
- Validar se contas Oracle somente do esquema são usadas
- Validar se as colunas virtuais são usadas
- Valide se os nomes das tabelas definidos no escopo da tarefa contêm apóstrofos.
- Valide se as colunas definidas no escopo da tarefa têm XMLType ou não Long Raw tipos de dados e verifique a configuração do modo LOB nas configurações da tarefa. Long
- Valide se a versão de origem do Oracle é suportada pelo AWS DMS.
- Valide se a versão de destino do Oracle é suportada pelo AWS DMS.
- Valide se a versão de destino do Oracle é suportada pelo AWS DMS.
- Valide se o usuário do DMS tem as permissões necessárias para usar a validação de dados.
- Validar se o usuário do DMS tem permissões para usar o Binary Reader com o Oracle ASM
- Valide se o usuário do DMS tem permissões para usar o Binary Reader com Oracle não-ASM
- Valide se o usuário do DMS tem permissões para usar o Binary Reader com o método CopyToTempFolder
- Validar se o usuário do DMS tem permissões para usar o Oracle Standby como fonte
- Validar se a fonte do DMS está conectada a um contêiner de aplicativos (PDB)
- Valide se a tabela tem tipos de dados XML incluídos no escopo da tarefa.
- Valide se o modo de registro de arquivamento está ativado no banco de dados de origem.
- Valida a retenção de registros de arquivamento para o RDS Oracle.
- Valide se a tabela tem tipos de dados estendidos incluídos no escopo da tarefa.

- Valide o tamanho do nome do objeto incluído no escopo da tarefa.
- Validar se a origem do DMS está conectada a um Oracle PDB
- Valide se a tabela tem colunas espaciais incluídas no escopo da tarefa.
- Valide se a origem do DMS está conectada a um Oracle standby.
- Valide se o espaço de tabela do banco de dados de origem está criptografado usando o TDE.
- Validar se o banco de dados de origem é Oracle ASM

Verificar o registro em log suplementar no nível do banco de dados

Chave da API: `oracle-supplemental-db-level`

Essa avaliação de pré-migração valida se o registro em log suplementar mínimo está ativado no nível do banco de dados. Ative o registro em log suplementar para utilizar um banco de dados Oracle como origem da migração.

Para ativar o registro em log suplementar, utilize a seguinte consulta:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
```

Para ter mais informações, consulte Configuração de registro em log suplementar.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Valide se o link para o banco de dados necessário foi criado para espera

Chave da API: `oracle-validate-standby-dblink`

Essa avaliação de pré-migração valida se o Dblink foi criado para a origem do banco de dados standby Oracle. AWSDMS_DBLINK é um pré-requisito para usar um banco de dados em espera como fonte. Ao utilizar o Oracle Standby como origem, o AWS DMS não valida transações abertas por padrão.

Para ter mais informações, consulte Trabalhando com um banco de dados Oracle autogerenciado como fonte para AWS DMS.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Validação Oracle para o tipo de dados LOB e se o leitor binário estiver configurado

Chave da API: `oracle-binary-lob-source-validation`

Essa avaliação de pré-migração valida se o Oracle LogMiner é usado para um endpoint de banco de dados Oracle versão 12c ou posterior. AWS DMS não oferece suporte ao Oracle LogMiner para migrações de colunas LOB dos bancos de dados Oracle versão 12c. Essa avaliação também verifica a presença de colunas LOB e fornece recomendações apropriadas.

Para configurar sua migração para não usar o Oracle LogMiner, adicione a seguinte configuração ao seu endpoint de origem:

```
useLogMinerReader=N;useBfile=Y;
```

Para ter mais informações, consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Validar se o banco de dados é CDB

Chave da API: `oracle-validate-cdb`

Essa avaliação de pré-migração valida se o banco de dados é um banco de dados de contêineres. O AWS DMS não é compatível com o banco de dados raiz de contêineres multilocatários (CDB $ROOT).

> **Note**
>
> Essa avaliação só é necessária para as versões 12.1.0.1 ou posteriores do Oracle. Essa avaliação não é aplicável às versões do Oracle anteriores a 12.1.0.1.

Para ter mais informações, consulte Limitações no uso da Oracle como fonte para AWS DMS.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Confira o Oracle Database Edition

Chave da API: `oracle-check-cdc-support-express-edition`

Essa avaliação de pré-migração validará se o banco de dados de origem do Oracle for Express Edition. O AWS DMS não é compatível com a CDC para Oracle Express Edition (Oracle Database XE) versão 18.0 e posterior.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Validar o método CDC do DMS para Oracle

Chave da API: `oracle-recommendation-cdc-method`

Essa avaliação de pré-migração valida a geração de redo logs nos últimos sete dias e faz uma recomendação sobre o uso do AWS DMS Binary Reader ou do Oracle LogMiner para CDC.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para obter mais informações sobre como decidir qual método de CDC utilizar, consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.

Validar a configuração do Oracle RAC para o DMS

Chave da API: `oracle-check-rac`

Essa avaliação de pré-migração valida se o banco de dados Oracle é uma Real Application Cluster. Os bancos de dados Real Application Cluster devem estar configurados corretamente. Se o banco de dados for baseado no RAC, recomendamos que você use o AWS DMS Binary Reader for CDC em vez do Oracle. LogMiner

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.

Valide se o usuário do DMS tem permissões no destino

Chave da API: `oracle-validate-permissions-on-target`

Essa avaliação de pré-migração valida se os usuários do DMS têm todas as permissões necessárias no banco de dados de destino.

Valide se o registro suplementar é necessário para todas as colunas

Chave da API: `oracle-validate-supplemental-logging-all-columns`

Essa avaliação de pré-migração valida, para as tabelas mencionadas no escopo da tarefa, se o registro suplementar foi adicionado a todas as colunas de tabelas sem uma chave primária ou exclusiva. Sem o registro suplementar em todas as colunas de uma tabela sem uma chave primária ou exclusiva, a before-and-after imagem dos dados não estará disponível nos redo logs. O DMS exige registro suplementar para tabelas sem uma chave primária ou exclusiva para gerar instruções DML.

Valide se o registro suplementar está ativado em tabelas com chaves primárias ou exclusivas

Chave da API: `oracle-validate-supplemental-logging-for-pk`

Essa avaliação de pré-migração valida se o registro suplementar está habilitado para tabelas com uma chave primária ou índice exclusivo e também verifica se `AddSupplementalLogging` está habilitado no nível do endpoint. Para garantir que o DMS possa replicar as alterações, você pode adicionar manualmente o registro suplementar no nível da tabela com base na chave primária ou na chave exclusiva ou utilizar a configuração do endpoint `AddSupplementalLogging = true` com um usuário do DMS com a permissão ALTER em qualquer tabela replicada.

Valide se há SecureFile LOBs e se a tarefa está configurada para o modo LOB completo

Chave da API: `oracle-validate-securefile-lobs`

Essa avaliação de pré-migração verifica a presença de SecureFile LOBs em tabelas dentro do escopo da tarefa e verifica suas configurações de LOB. É importante observar que os SecureFile LOBs atualmente são suportados somente durante o modo FULL LOB. Considere atribuir tabelas LOB a uma tarefa separada para melhorar o desempenho, pois a execução de tarefas no modo LOB completo pode resultar em um desempenho mais lento.

Valide se os índices baseados em funções estão sendo usados nas tabelas incluídas no escopo da tarefa.

Chave da API: `oracle-validate-function-based-indexes`

Essa avaliação de pré-migração verifica os índices baseados em funções nas tabelas dentro do escopo da tarefa. Observe que isso AWS DMS não oferece suporte à replicação de índices baseados em funções. Considere criar os índices após a migração no banco de dados de destino.

Valide se tabelas temporárias globais estão sendo usadas nas tabelas incluídas no escopo da tarefa.

Chave da API: `oracle-validate-global-temporary-tables`

Essa avaliação de pré-migração verifica se as tabelas temporárias globais são usadas dentro do escopo do mapeamento de tabelas de tarefas. Observe que isso AWS DMS não oferece suporte à migração ou replicação de tabelas temporárias globais.

Valide se tabelas organizadas por índice com um segmento de estouro estão sendo usadas nas tabelas incluídas no escopo da tarefa.

Chave da API: `oracle-validate-iot-overflow-segments`

Valide se tabelas organizadas por índice com um segmento de estouro estão sendo usadas nas tabelas incluídas no escopo da tarefa. AWS DMS não oferece suporte ao CDC para tabelas organizadas por índice com um segmento de estouro.

Valide se tabelas de aninhamento de vários níveis são usadas nas tabelas incluídas no escopo da tarefa.

Chave da API: `oracle-validate-more-than-one-nesting-table-level`

Essa avaliação de pré-migração verifica o nível de aninhamento da tabela aninhada usada no escopo da tarefa. AWS DMS suporta somente um nível de aninhamento de tabelas.

Valide se colunas invisíveis são usadas nas tabelas incluídas no escopo da tarefa.

Chave da API: `oracle-validate-invisible-columns`

Essa avaliação de pré-migração valida se as tabelas usadas no escopo da tarefa têm colunas invisíveis. AWS DMS não migra dados de colunas invisíveis em seu banco de dados de origem. Para migrar as colunas invisíveis, você precisa modificá-las para ficarem visíveis.

Valide se visualizações materializadas baseadas em uma coluna ROWID são usadas nas tabelas incluídas no escopo da tarefa.

Chave da API: `oracle-validate-rowid-based-materialized-views`

Essa avaliação de pré-migração valida se as visualizações materializadas usadas na migração são criadas com base na coluna ROWID. AWS DMS não suporta o tipo de dados ROWID ou visualizações materializadas com base em uma coluna ROWID.

Valide se o recurso Active Data Guard DML Redirect é usado.

Chave da API: `oracle-validate-adg-redirect-dml`

Essa avaliação de pré-migração valida se o recurso Active Data Guard DML Redirect é usado. Ao usar o Oracle 19.0 como fonte, AWS DMS não oferece suporte ao recurso de redirecionamento de DML do Data Guard.

Valide se tabelas particionadas híbridas são usadas.

Chave da API: `oracle-validate-hybrid-partitioned-tables`

Essa avaliação de pré-migração valida se as tabelas particionadas híbridas são usadas para as tabelas definidas no escopo da tarefa.

Validar se contas Oracle somente do esquema são usadas

Chave da API: `oracle-validate-schema-only-accounts`

Essa avaliação de pré-migração valida se as contas somente do esquema são encontradas dentro do escopo da tarefa.

Validar se as colunas virtuais são usadas

Chave da API: `oracle-validate-virtual-columns`

Essa avaliação de pré-migração valida se a instância Oracle tem colunas virtuais em tabelas dentro do escopo da tarefa.

Valide se os nomes das tabelas definidos no escopo da tarefa contêm apóstrofos.

Chave da API: `oracle-validate-names-with-apostrophes`

Essa avaliação de pré-migração valida se as tabelas usadas no escopo da tarefa contêm apóstrofos. AWS DMS não replica tabelas com nomes contendo apóstrofos. Se identificado, considere renomear essas tabelas. Como alternativa, você pode criar uma visualização ou visualização materializada sem apóstrofos para carregar essas tabelas.

Valide se as colunas definidas no escopo da tarefa têm **`XMLType`** ou não **`Long Raw`** tipos de dados e verifique a configuração do modo LOB nas configurações da tarefa. **`Long`**

Chave da API: `oracle-validate-limited-lob-mode-for-longs`

Essa avaliação de pré-migração valida se as tabelas definidas no escopo da tarefa têm os tipos de dados, ou `XMLType` `LongLong Raw`, e verifica se a configuração da tarefa está configurada para usar o Modo LOB de Tamanho Limitado. AWS DMS não suporta a replicação desses tipos de dados usando o modo FULL LOB. Considere alterar a configuração da tarefa para usar o modo LOB de tamanho limitado ao identificar tabelas com esses tipos de dados.

Valide se a versão de origem do Oracle é suportada pelo AWS DMS.

Chave da API: `oracle-validate-supported-versions-of-source`

Essa avaliação de pré-migração valida se a versão de origem da instância Oracle é suportada pelo. AWS DMS

Valide se a versão de destino do Oracle é suportada pelo AWS DMS.

Chave da API: `oracle-validate-supported-versions-of-target`

Essa avaliação de pré-migração valida se a versão da instância Oracle de destino é suportada pelo. AWS DMS

Valide se a versão de destino do Oracle é suportada pelo AWS DMS.

Chave da API: `oracle-validate-supported-versions-of-target`

Essa avaliação de pré-migração valida se a versão da instância Oracle de destino é suportada pelo. AWS DMS

Valide se o usuário do DMS tem as permissões necessárias para usar a validação de dados.

Chave da API: `oracle-prerequisites-privileges-of-validation-feature`

Essa avaliação de pré-migração valida se o usuário do DMS tem os privilégios necessários para usar a Validação de Dados do DMS. Você pode ignorar a ativação dessa validação se não pretender usar a validação de dados.

Validar se o usuário do DMS tem permissões para usar o Binary Reader com o Oracle ASM

Chave da API: `oracle-prerequisites-privileges-of-binary-reader-asm`

Essa avaliação de pré-migração valida se o usuário do DMS tem os privilégios necessários para usar o Binary Reader na instância do Oracle ASM. Você pode ignorar a ativação dessa avaliação se sua

origem não for uma instância do Oracle ASM ou se você não estiver usando o Binary Reader for CDC.

Valide se o usuário do DMS tem permissões para usar o Binary Reader com Oracle não-ASM

Chave da API: `oracle-prerequisites-privileges-of-binary-reader-non-asm`

Essa avaliação de pré-migração valida se o usuário do DMS tem os privilégios necessários para usar o Binary Reader na instância Oracle não ASM. Essa avaliação só é válida se você tiver uma instância Oracle não ASM.

Valide se o usuário do DMS tem permissões para usar o Binary Reader com o método CopyToTempFolder

Chave da API: `oracle-prerequisites-privileges-of-binary-reader-copy-to-temp-folder`

Essa avaliação de pré-migração valida se o usuário do DMS tem os privilégios necessários para usar o Binary Reader com o método 'Copiar para pasta temporária'. Essa avaliação é relevante somente se você planeja usar CopyToTempFolder para ler as alterações do CDC ao usar o Binary Reader e ter uma instância do ASM conectada à origem. Você pode ignorar a ativação dessa avaliação se não pretender usar o CopyToTempFolder recurso.

Recomendamos não usar o CopyToTempFolder recurso porque ele está obsoleto.

Validar se o usuário do DMS tem permissões para usar o Oracle Standby como fonte

Chave da API: `oracle-prerequisites-privileges-of-standby-as-source`

Essa avaliação de pré-migração valida se o usuário do DMS tem os privilégios necessários para usar uma instância StandBy Oracle como origem. Você pode ignorar a ativação dessa avaliação se não pretender usar uma instância StandBy Oracle como fonte.

Validar se a fonte do DMS está conectada a um contêiner de aplicativos (PDB)

Chave da API: `oracle-check-app-pdb`

Essa avaliação de pré-migração valida se a fonte do DMS está conectada a um PDB do contêiner de aplicativos. O DMS não oferece suporte à replicação de um PDB de contêiner de aplicativos.

Valide se a tabela tem tipos de dados XML incluídos no escopo da tarefa.

Chave da API: `oracle-check-xml-columns`

Essa avaliação de pré-migração valida se as tabelas usadas no escopo da tarefa têm tipos de dados XML. Também verifica se a tarefa está configurada para o modo LOB limitado quando a tabela contém um tipo de dados XML. O DMS suporta somente o modo LOB limitado para migrar colunas Oracle XML.

Valide se o modo de registro de arquivamento está ativado no banco de dados de origem.

Chave da API: `oracle-check-archivelog-mode`

Essa avaliação de pré-migração valida se o modo de registro de arquivamento está ativado no banco de dados de origem. É necessário ativar o modo de registro de arquivamento no banco de dados de origem para que o DMS replique as alterações.

Valida a retenção de registros de arquivamento para o RDS Oracle.

Chave da API: `oracle-check-archivelog-retention-rds`

Essa avaliação de pré-migração valida se a retenção de registros de arquivamento em seu banco de dados Oracle do RDS está configurada por pelo menos 24 horas.

Valide se a tabela tem tipos de dados estendidos incluídos no escopo da tarefa.

Chave da API: `oracle-check-extended-columns`

Essa avaliação de pré-migração valida se as tabelas usadas no escopo da tarefa têm tipos de dados estendidos. Observe que os tipos de dados estendidos são suportados somente com o DMS versão 3.5 em diante.

Valide o tamanho do nome do objeto incluído no escopo da tarefa.

Chave da API: `oracle-check-object-30-bytes-limit`

Essa avaliação de pré-migração valida se o tamanho do nome do objeto excede 30 bytes. O DMS não suporta nomes de objetos longos (mais de 30 bytes).

Validar se a origem do DMS está conectada a um Oracle PDB

Chave da API: `oracle-check-pdb-enabled`

Essa avaliação de pré-migração valida se a fonte do DMS está conectada a um PDB. O DMS suporta CDC somente ao usar o Binary Reader com o Oracle PDB como fonte. A avaliação também avalia se a tarefa está configurada para usar o leitor binário quando o DMS está conectado ao Oracle PDB.

Valide se a tabela tem colunas espaciais incluídas no escopo da tarefa.

Chave da API: `oracle-check-spatial-columns`

Essa avaliação de pré-migração valida se a tabela tem colunas espaciais incluídas no escopo da tarefa. O DMS suporta tipos de dados espaciais somente usando o modo LOB completo. A avaliação também avalia se a tarefa está configurada para usar o modo LOB completo quando o DMS identifica colunas espaciais.

Valide se a origem do DMS está conectada a um Oracle standby.

Chave da API: `oracle-check-standby-db`

Essa avaliação de pré-migração valida se a origem está conectada a um Oracle standby. O DMS suporta CDC somente ao usar o leitor binário com o Oracle Standby como fonte. A avaliação também avalia se a tarefa está configurada para usar o leitor binário quando o DMS está conectado ao Oracle Standby.

Valide se o espaço de tabela do banco de dados de origem está criptografado usando o TDE.

Chave da API: `oracle-check-tde-enabled`

Essa avaliação de pré-migração valida se a fonte tem a criptografia TDE habilitada no espaço de tabela. O DMS suporta TDE somente com espaços de tabela criptografados ao usar Oracle para RDS Oracle. LogMiner

Validar se o banco de dados de origem é Oracle ASM

Chave da API: `oracle-check-asm`

Essa avaliação de pré-migração valida se a fonte usa o ASM. Para melhorar o desempenho com a configuração do ASM, considere adicionar `parallelASMReadThreads` e `readAheadBlocks` às configurações do endpoint de origem.

Avaliações do SQL Server

Esta seção descreve avaliações de pré-migração individuais para tarefas de migração que utilizam um endpoint de origem do Microsoft SQL Server.

Tópicos

- Verificar se o modelo de recuperação do banco de dados é simples
- Verificar se as tabelas no escopo da tarefa contêm colunas computadas

- Verificar se as tabelas no escopo da tarefa têm índices de armazenamento de colunas
- Verificar se as tabelas otimizadas para memória fazem parte do escopo da tarefa
- Verificar se as tabelas temporais fazem parte do escopo da tarefa
- Verificar se a durabilidade atrasada está ativada no nível do banco de dados
- Verificar se a recuperação acelerada de dados está ativada no nível do banco de dados
- Verificar se o mapeamento de tabela tem mais de 10 mil tabelas com chaves primárias
- Verifique se o banco de dados de origem tem nomes de tabelas ou esquemas com caracteres especiais.
- Verifique se o banco de dados de origem tem nomes de colunas com dados mascarados
- Verifique se o banco de dados de origem tem backups criptografados
- Verifique se o banco de dados de origem tem backups armazenados em uma URL ou no Windows Azure.
- Verifique se o banco de dados de origem tem backups em vários discos
- Verifique se o banco de dados de origem tem pelo menos um backup completo
- Verifique se o banco de dados de origem tem colunas esparsas e compressão da estrutura colunar.
- Verifique se a instância do banco de dados de origem tem auditoria em nível de servidor para SQL Server 2008 ou SQL Server 2008 R2
- Verifique se o banco de dados de origem tem colunas de geometria para o modo LOB completo
- Verifique se o banco de dados de origem tem colunas com a propriedade Identity.
- Verifique se o usuário do DMS tem permissões FULL LOAD
- Verifique se o usuário do DMS tem permissões FULL LOAD e CDC ou somente CDC
- Verifique se o ignoreMsReplicationEnablement ECA está configurado ao usar o MS-CDC com bancos de dados locais ou EC2
- Verifique se o usuário do DMS tem a permissão VIEW DEFINITION.
- Verifique se o usuário do DMS tem a permissão VIEW DATABASE STATE no banco de dados MASTER para usuários sem a função Sysadmin.
- Verifique se o usuário do DMS tem a permissão VIEW SERVER STATE.

Verificar se o modelo de recuperação do banco de dados é simples

Chave da API: `sqlserver-check-for-recovery-model`

Essa avaliação de pré-migração valida o modelo de recuperação do endpoint de origem. AWS DMS exige que o modelo de recuperação seja configurado para `Bulk logged` ou `Full` para replicação contínua.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Utilizar replicação contínua (CDC) a partir de uma origem do SQL Server.

Verificar se as tabelas no escopo da tarefa contêm colunas computadas

Chave da API: `sqlserver-check-for-computed-fields`

Essa avaliação de pré-migração verifica a presença de colunas computadas. AWS DMS não oferece suporte à replicação de alterações de colunas computadas do SQL Server.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verificar se as tabelas no escopo da tarefa têm índices de armazenamento de colunas

Chave da API: `sqlserver-check-for-columnstore-indexes`

Essa avaliação de pré-migração verifica a presença de tabelas com índices columnstore. AWS DMS não oferece suporte à replicação de alterações de tabelas do SQL Server com índices columnstore.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verificar se as tabelas otimizadas para memória fazem parte do escopo da tarefa

Chave da API: `sqlserver-check-for-memory-optimized-tables`

Essa avaliação de pré-migração verifica a presença de tabelas otimizadas para memória. AWS DMS não oferece suporte à replicação de alterações de tabelas otimizadas para memória.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verificar se as tabelas temporais fazem parte do escopo da tarefa

Chave da API: `sqlserver-check-for-temporal-tables`

Essa avaliação de pré-migração verifica a presença de tabelas temporais. AWS DMS não suporta a replicação de alterações de tabelas temporais.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verificar se a durabilidade atrasada está ativada no nível do banco de dados

Chave da API: `sqlserver-check-for-delayed-durability`

Essa avaliação de pré-migração verifica a presença de durabilidade retardada. AWS DMS não suporta a replicação de alterações de transações que usam durabilidade retardada.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verificar se a recuperação acelerada de dados está ativada no nível do banco de dados

Chave da API: `sqlserver-check-for-accelerated-data-recovery`

Essa avaliação de pré-migração verifica a presença de recuperação acelerada de dados. AWS DMS não suporta a replicação de alterações de bancos de dados com recuperação acelerada de dados.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verificar se o mapeamento de tabela tem mais de 10 mil tabelas com chaves primárias

Chave da API: `sqlserver-large-number-of-tables`

Essa avaliação de pré-migração verifica a presença de mais de 10.000 tabelas com chaves primárias. Os bancos de dados configurados com o MS-Replication podem apresentar falhas nas tarefas se houver muitas tabelas com chaves primárias.

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para obter mais informações sobre como configurar a MS-Replication, consulte Capturar dados alterados no SQL Server autogerenciado on-premises ou no Amazon EC2.

Verifique se o banco de dados de origem tem nomes de tabelas ou esquemas com caracteres especiais.

Chave da API: `sqlserver-check-for-special-characters`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem nomes de tabela ou esquema que incluem um caractere do seguinte conjunto:

```
\\ -- \n \" \b \r ' \t ;
```

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem nomes de colunas com dados mascarados

Chave da API: `sqlserver-check-for-masked-data`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem dados mascarados. AWS DMS migra dados mascarados sem mascarar.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem backups criptografados

Chave da API: `sqlserver-check-for-encrypted-backups`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem backups criptografados.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem backups armazenados em uma URL ou no Windows Azure.

Chave da API: `sqlserver-check-for-backup-url`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem backups armazenados em uma URL ou no Windows Azure.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem backups em vários discos

Chave da API: `sqlserver-check-for-backup-multiple-stripes`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem backups em vários discos.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem pelo menos um backup completo

Chave da API: `sqlserver-check-for-full-backup`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem pelo menos um backup completo. O SQL Server deve estar configurado para backup completo e você deve executar um backup antes de replicar os dados.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem colunas esparsas e compressão da estrutura colunar.

Chave da API: `sqlserver-check-for-sparse-columns`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem colunas esparsas e compressão da estrutura colunar. O DMS não suporta colunas esparsas e compressão de estrutura colunar.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se a instância do banco de dados de origem tem auditoria em nível de servidor para SQL Server 2008 ou SQL Server 2008 R2

Chave da API: `sqlserver-check-for-audit-2008`

Essa avaliação de pré-migração verifica se o banco de dados de origem habilitou a auditoria em nível de servidor para o SQL Server 2008 ou o SQL Server 2008 R2. O DMS tem um problema conhecido relacionado com o SQL Server 2008 e 2008 R2.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem colunas de geometria para o modo LOB completo

Chave da API: `sqlserver-check-for-geometry-columns`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem colunas de geometria para o modo LOB (Objeto Grande) completo ao usar o SQL Server como fonte. Recomendamos usar o modo LOB limitado ou definir a configuração da `InlineLobMaxSize` tarefa para usar o modo LOB embutido quando seu banco de dados incluir colunas de geometria.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o banco de dados de origem tem colunas com a propriedade Identity.

Chave da API: `sqlserver-check-for-identity-columns`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem uma coluna com a `IDENTITY` propriedade. O DMS não migra essa propriedade para a coluna correspondente do banco de dados de destino.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o usuário do DMS tem permissões FULL LOAD

Chave da API: `sqlserver-check-user-permission-for-full-load-only`

Essa avaliação de pré-migração verifica se o usuário da tarefa do DMS tem permissões para executar a tarefa no modo FULL LOAD.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o usuário do DMS tem permissões FULL LOAD e CDC ou somente CDC

Chave da API: `sqlserver-check-user-permission-for-cdc`

Essa avaliação de pré-migração verifica se o usuário do DMS tem permissões para executar a tarefa nos `FULL LOAD and CDC` modos ou. `CDC only`

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o **`ignoreMsReplicationEnablement`** ECA está configurado ao usar o MS-CDC com bancos de dados locais ou EC2

Chave da API: `sqlserver-check-attribute-for-enable-ms-cdc-onprem`

Verifique se o atributo de conexão `ignoreMsReplicationEnablement` extra (ECA) está definido ao usar o MS-CDC com bancos de dados locais ou EC2.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o usuário do DMS tem a permissão VIEW DEFINITION.

Chave da API: `sqlserver-check-user-permission-on-view-definition`

Essa avaliação de pré-migração verifica se o usuário especificado nas configurações do endpoint tem a permissão. `VIEW DEFINITION` O DMS exige a `VIEW DEFINITION` permissão para visualizar as definições de objetos.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o usuário do DMS tem a permissão VIEW DATABASE STATE no banco de dados MASTER para usuários sem a função Sysadmin.

Chave da API: `sqlserver-check-user-permission-on-view-database-state`

Essa avaliação de pré-migração verifica se o usuário especificado nas configurações do endpoint tem a permissão. `VIEW DATABASE STATE` O DMS exige essa permissão para acessar objetos do banco de dados no banco de dados MASTER. O DMS também exige essa permissão quando o usuário não tem privilégios de administrador de sistema. O DMS exige essa permissão para criar funções, certificados e logins e para conceder credenciais.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Verifique se o usuário do DMS tem a permissão VIEW SERVER STATE.

Chave da API: `sqlserver-check-user-permission-on-view-server-state`

Essa avaliação de pré-migração verifica se o usuário especificado nos atributos extras de conexão (ECA) tem a `VIEW SERVER STATE` permissão. `VIEW SERVER STATE`é uma permissão em nível de servidor que permite ao usuário visualizar informações e estados de todo o servidor. Essa permissão fornece acesso a exibições de gerenciamento dinâmico (DMVs) e funções de gerenciamento dinâmico (DMFs) que expõem informações sobre a instância do SQL Server. Essa permissão é necessária para que o usuário do DMS tenha acesso aos recursos do CDC. Essa permissão é necessária para executar uma tarefa do DMS nos `CDC only` modos `FULL LOAD and CDC` ou.

Para ter mais informações, consulte Limitações no uso do SQL Server como fonte para AWS DMS.

Avaliações do MySQL

Esta seção descreve avaliações individuais de pré-migração para tarefas de migração que usam um endpoint de origem do MySQL.

Tópicos

- Validar se uma tabela usa um mecanismo de armazenamento diferente do Innodb
- Valide se o incremento automático está ativado em qualquer tabela usada para migração
- Validar se a imagem do log binário do banco de dados está configurada FULL para suportar o DMS CDC
- Valide se o banco de dados de origem é uma réplica de leitura do MySQL
- Valide se uma tabela tem partições e recomende configurações de tarefas target_table_prep_mode de carga total
- Validar se o DMS suporta a versão do banco de dados
- Validar se o banco de dados de destino está configurado para ser definido como local_infile 1
- Valide se o banco de dados de destino tem tabelas com chaves estrangeiras
- Valide se as tabelas de origem no escopo da tarefa têm restrições em cascata
- Valide se os valores de tempo limite são apropriados para uma fonte ou destino do MySQL

Validar se uma tabela usa um mecanismo de armazenamento diferente do Innodb

Chave da API: `mysql-check-table-storage-engine`

Essa avaliação de pré-migração valida se o mecanismo de armazenamento usado para qualquer tabela no banco de dados MySQL de origem é um mecanismo diferente do Innodb. O DMS cria tabelas de destino com o mecanismo de armazenamento InnoDB por padrão. Se você precisar usar um mecanismo de armazenamento diferente do InnoDB, deverá criar manualmente a tabela no banco de dados de destino e configurar sua tarefa DMS para uso `TRUNCATE_BEFORE_LOAD` ou `DO_NOTHING` como configuração de tarefa de carga total. Para obter mais informações sobre as configurações de tarefas de carga total, consulteConfigurações de tarefa de carregamento completo.

Para obter mais informações sobre as limitações do endpoint MySQL, consulte. Limitações no uso de um banco de dados MySQL como fonte para AWS DMS

Valide se o incremento automático está ativado em qualquer tabela usada para migração

Chave da API: `mysql-check-auto-increment`

Essa avaliação de pré-migração valida se as tabelas de origem usadas na tarefa têm o incremento automático ativado. O DMS não migra o atributo AUTO_INCREMENT em uma coluna para um banco de dados de destino.

Para obter mais informações sobre as limitações do endpoint MySQL, consulte. Limitações no uso de um banco de dados MySQL como fonte para AWS DMS Para obter informações sobre como lidar

com colunas de identidade no MySQL, consulte Como lidar com colunas de identidade em AWS DMS: Parte 2.

Validar se a imagem do log binário do banco de dados está configurada **FULL** para suportar o DMS CDC

Chave da API: `mysql-check-binlog-image`

Essa avaliação de pré-migração verifica se a imagem do log binário do banco de dados de origem está definida como. `FULL` No MySQL, a `binlog_row_image` variável determina como um evento de log binário é gravado ao usar o `ROW` formato. Para garantir a compatibilidade com o DMS e oferecer suporte ao CDC, defina a `binlog_row_image` variável como. `FULL` Essa configuração garante que o DMS receba informações suficientes para construir a Linguagem de Manipulação de Dados (DML) completa para o banco de dados de destino durante a migração.

Para definir a imagem do log binário como`FULL`, faça o seguinte:

- Para o Amazon RDS, esse valor é `FULL` por padrão.
- Para bancos de dados hospedados localmente ou no Amazon EC2, `binlog_row_image` defina o valor em `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Valide se o banco de dados de origem é uma réplica de leitura do MySQL

Chave da API: `mysql-check-database-role`

Essa avaliação de pré-migração verifica se o banco de dados de origem é uma réplica de leitura. Para ativar o suporte do CDC para DMS quando conectado a uma réplica de leitura, defina o `log_slave_updates` parâmetro como. `True` Para obter mais informações sobre o uso de um banco de dados MySQL autogerenciado, consulte. Usando um banco de dados autogerenciado compatível com MySQL como fonte para AWS DMS

Para definir o `log_slave_updates` valor como`True`, faça o seguinte:

- Para o Amazon RDS, use o grupo de parâmetros do banco de dados. Para obter informações sobre o uso de grupos de parâmetros do banco de dados do RDS, consulte Como trabalhar com grupos de parâmetros no Guia do usuário do Amazon RDS.

- Para bancos de dados hospedados localmente ou no Amazon EC2, `log_slave_updates` defina o valor em `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Valide se uma tabela tem partições e recomende configurações de tarefas **`target_table_prep_mode`** de carga total

Chave da API: `mysql-check-table-partition`

Essa avaliação de pré-migração verifica a presença de tabelas com partições no banco de dados de origem. O DMS cria tabelas sem partições no destino do MySQL. Para migrar tabelas particionadas para uma tabela particionada no destino, você deve fazer o seguinte:

- Pré-crie as tabelas particionadas no banco de dados MySQL de destino.
- Configure sua tarefa DMS para usar `TRUNCATE_BEFORE_LOAD` ou `DO_NOTHING` como configuração de tarefa de carga total.

Para obter mais informações sobre as limitações do endpoint MySQL, consulte. Limitações no uso de um banco de dados MySQL como fonte para AWS DMS

Validar se o DMS suporta a versão do banco de dados

Chave da API: `mysql-check-supported-version`

Essa avaliação de pré-migração verifica se a versão do banco de dados de origem é compatível com o DMS. O CDC não é compatível com as versões 5.5 ou inferiores do MySQL do Amazon RDS nem com versões do MySQL maiores que 8.0.x. O CDC é compatível somente com as versões 5.6, 5.7 ou 8.0 do MySQL. Para obter mais informações sobre as versões compatíveis do MySQL, consulte. Endpoints de origem da migração de dados

Validar se o banco de dados de destino está configurado para ser definido como **`local_infile`** 1

Chave da API: `mysql-check-target-localinfile-set`

Essa avaliação de pré-migração verifica se o `local_infile` parâmetro no banco de dados de destino está definido como 1. O DMS exige que o parâmetro 'local_infile' seja definido como 1 durante a carga total no banco de dados de destino. Para ter mais informações, consulte Migração do MySQL para o MySQL utilizando o AWS DMS.

Essa avaliação só é válida para uma tarefa de carga completa ou carga total e do CDC.

Valide se o banco de dados de destino tem tabelas com chaves estrangeiras

Chave da API: `mysql-check-fk-target`

Essa avaliação de pré-migração verifica se uma tarefa de CDC com carga total ou total migrando para um banco de dados MySQL tem tabelas com chaves estrangeiras. A configuração padrão no DMS é carregar tabelas em ordem alfabética. Tabelas com chaves estrangeiras e restrições de integridade referencial podem causar falha no carregamento, pois as tabelas principal e secundária podem não ser carregadas ao mesmo tempo.

Para obter mais informações sobre integridade referencial no DMS, consulte Trabalho com índices, acionadores e restrições de integridade referencial no tópico. Aprimoramento do desempenho de uma migração do AWS DMS

Valide se as tabelas de origem no escopo da tarefa têm restrições em cascata

Chave da API: `mysql-check-cascade-constraints`

Essa avaliação de pré-migração verifica se alguma das tabelas de origem do MySQL tem restrições em cascata. As restrições em cascata não são migradas ou replicadas pelas tarefas do DMS, porque o MySQL não registra as alterações desses eventos no log binário. Embora AWS DMS não ofereça suporte a essas restrições, você pode usar soluções alternativas para destinos de bancos de dados relacionais.

Para obter informações sobre o suporte a restrições de cascata e outras restrições, consulte o tópico Solução Índices, chaves estrangeiras ou atualizações ou exclusões em cascata não migrados de problemas de migração. AWS DMS

Valide se os valores de tempo limite são apropriados para uma fonte ou destino do MySQL

Chave da API: `mysql-check-network-parameter`

Essa avaliação de pré-migração verifica se o endpoint MySQL de uma tarefa tem `net_read_timeout` as `net_wait_timeout` configurações `wait_timeout` e definidas para pelo menos 300 segundos. Isso é necessário para evitar desconexões durante a migração.

Para ter mais informações, consulte Conexões a uma instância de destino MySQL são desconectadas durante uma tarefa.

Avaliações do MariaDB

Esta seção descreve avaliações individuais de pré-migração para tarefas de migração que usam um endpoint de origem do MariaDB.

Para criar uma avaliação individual de pré-migração usando a AWS DMS API, use a chave de API listada para o `Include` parâmetro da StartReplicationTaskAssessmentRunação.

Tópicos

- Validar se uma tabela usa um mecanismo de armazenamento diferente do Innodb
- Valide se o incremento automático está ativado em qualquer tabela usada para migração
- Valide se o formato do log binário do banco de dados está definido como compatível com ROW DMS CDC
- Validar se a imagem do log binário do banco de dados está configurada FULL para suportar o DMS CDC
- Valide se o banco de dados de origem é uma réplica de leitura do MariaDB
- Valide se uma tabela tem partições e recomende TRUNCATE_BEFORE_LOAD ou DO_NOTHING para configurações de tarefas de carga total
- Validar se o DMS suporta a versão do banco de dados
- Validar se o banco de dados de destino está configurado para ser definido como local_infile 1
- Valide se o banco de dados de destino tem tabelas com chaves estrangeiras
- Valide se as tabelas de origem no escopo da tarefa têm restrições em cascata
- Validar se as tabelas de origem no escopo da tarefa geraram colunas
- Valide se os valores de tempo limite são apropriados para uma fonte do MariaDB
- Valide se os valores de tempo limite são apropriados para um destino do MariaDB

Validar se uma tabela usa um mecanismo de armazenamento diferente do Innodb

Chave da API: `mariadb-check-table-storage-engine`

Essa avaliação de pré-migração valida se o mecanismo de armazenamento usado para qualquer tabela no banco de dados Source MariaDB é um mecanismo diferente do Innodb. O DMS cria tabelas de destino com o mecanismo de armazenamento InnoDB por padrão. Se você precisar usar um mecanismo de armazenamento diferente do InnoDB, deverá criar manualmente a tabela no banco de dados de destino e configurar sua tarefa DMS para uso `TRUNCATE_BEFORE_LOAD` ou

`DO_NOTHING` como configuração de tarefa de carga total. Para obter mais informações sobre as configurações de tarefas de carga total, consulteConfigurações de tarefa de carregamento completo.

Para obter mais informações sobre as limitações dos endpoints do MariaDB, consulte. Limitações no uso de um banco de dados MySQL como fonte para AWS DMS

Valide se o incremento automático está ativado em qualquer tabela usada para migração

Chave da API: `mariadb-check-auto-increment`

Essa avaliação de pré-migração valida se as tabelas de origem usadas na tarefa têm o incremento automático ativado. O DMS não migra o atributo AUTO_INCREMENT em uma coluna para um banco de dados de destino.

Para obter mais informações sobre as limitações dos endpoints do MariaDB, consulte. Limitações no uso de um banco de dados MySQL como fonte para AWS DMS Para obter informações sobre como lidar com colunas de identidade no MariaDB, consulte Handle IDENTITY columns AWS DMS em: Parte 2.

Valide se o formato do log binário do banco de dados está definido como compatível com **`ROW`** DMS CDC

Chave da API: `mariadb-check-binlog-format`

Essa avaliação de pré-migração valida se o formato do log binário do banco de dados de origem está configurado `ROW` para suportar o DMS Change Data Capture (CDC).

Para definir o formato do log binário como`ROW`, faça o seguinte:

- Para o Amazon RDS, use o grupo de parâmetros do banco de dados. Para obter informações sobre o uso de um grupo de parâmetros do RDS, consulte Como configurar o registro binário do MySQL no Guia do usuário do Amazon RDS.
- Para bancos de dados hospedados localmente ou no Amazon EC2, `binlog_format` defina o valor em `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para obter mais informações sobre servidores MariaDB auto-hospedados, consulte. Usando um banco de dados autogerenciado compatível com MySQL como fonte para AWS DMS

Validar se a imagem do log binário do banco de dados está configurada **FULL** para suportar o DMS CDC

Chave da API: `mariadb-check-binlog-image`

Essa avaliação de pré-migração verifica se a imagem do log binário do banco de dados de origem está definida como. `FULL` No MariaDB, `binlog_row_image` a variável determina como um evento de log binário é gravado ao usar o formato. `ROW` Para garantir a compatibilidade com o DMS e oferecer suporte ao CDC, defina a `binlog_row_image` variável como. `FULL` Essa configuração garante que o DMS receba informações suficientes para construir a Linguagem de Manipulação de Dados (DML) completa para o banco de dados de destino durante a migração.

Para definir a imagem do log binário como`FULL`, faça o seguinte:

- Para o Amazon RDS, esse valor é `FULL` por padrão.
- Para bancos de dados hospedados localmente ou no Amazon EC2, `binlog_row_image` defina o valor em `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Para obter mais informações sobre servidores MariaDB auto-hospedados, consulte. Usando um banco de dados autogerenciado compatível com MySQL como fonte para AWS DMS

Valide se o banco de dados de origem é uma réplica de leitura do MariaDB

Chave da API: `mariadb-check-database-role`

Essa avaliação de pré-migração verifica se o banco de dados de origem é uma réplica de leitura. Para ativar o suporte do CDC para DMS quando conectado a uma réplica de leitura, defina o `log_slave_updates` parâmetro como. `True` Para obter mais informações sobre o uso de um banco de dados MySQL autogerenciado, consulte. Usando um banco de dados autogerenciado compatível com MySQL como fonte para AWS DMS

Para definir o `log_slave_updates` valor como`True`, faça o seguinte:

- Para o Amazon RDS, use o grupo de parâmetros do banco de dados. Para obter informações sobre o uso de grupos de parâmetros do banco de dados do RDS, consulte Como trabalhar com grupos de parâmetros no Guia do usuário do Amazon RDS.

- Para bancos de dados hospedados localmente ou no Amazon EC2, `log_slave_updates` defina o valor em `my.ini` (Microsoft Windows) `my.cnf` ou (UNIX).

Essa avaliação só é válida para uma migração de carga máxima e CDC ou para uma migração somente de CDC. Essa avaliação não é válida para uma migração somente de carga máxima.

Valide se uma tabela tem partições e recomende **`TRUNCATE_BEFORE_LOAD`** ou **`DO_NOTHING`** para configurações de tarefas de carga total

Chave da API: `mariadb-check-table-partition`

Essa avaliação de pré-migração verifica a presença de tabelas com partições no banco de dados de origem. O DMS cria tabelas sem partições no destino do MariaDB. Para migrar tabelas particionadas para uma tabela particionada no destino, você deve fazer o seguinte:

- Pré-crie as tabelas particionadas no banco de dados MariaDB de destino.
- Configure sua tarefa DMS para usar `TRUNCATE_BEFORE_LOAD` ou `DO_NOTHING` como configuração de tarefa de carga total.

Para obter mais informações sobre as limitações dos endpoints do MariaDB, consulte. Limitações no uso de um banco de dados MySQL como fonte para AWS DMS

Validar se o DMS suporta a versão do banco de dados

Chave da API: `mariadb-check-supported-version`

Essa avaliação de pré-migração verifica se a versão do banco de dados de origem é compatível com o DMS. O CDC não é compatível com as versões 10.4 ou inferiores do MariaDB do Amazon RDS, nem com versões do MySQL maiores que 10.11. Para obter mais informações sobre as versões compatíveis do MariaDB, consulte. Endpoints de origem da migração de dados

Validar se o banco de dados de destino está configurado para ser definido como **`local_infile`** 1

Chave da API: `mariadb-check-target-localinfile-set`

Essa avaliação de pré-migração verifica se o `local_infile` parâmetro no banco de dados de destino está definido como 1. O DMS exige que o parâmetro 'local_infile' seja definido como 1 durante a carga total no banco de dados de destino. Para ter mais informações, consulte Migração do MySQL para o MySQL utilizando o AWS DMS.

Essa avaliação só é válida para uma tarefa de carga completa.

Valide se o banco de dados de destino tem tabelas com chaves estrangeiras

Chave da API: `mariadb-check-fk-target`

Essa avaliação de pré-migração verifica se uma tarefa de CDC com carga total ou total migrando para um banco de dados MariaDB tem tabelas com chaves estrangeiras. A configuração padrão no DMS é carregar tabelas em ordem alfabética. Tabelas com chaves estrangeiras e restrições de integridade referencial podem causar falha no carregamento, pois as tabelas principal e secundária podem não ser carregadas ao mesmo tempo.

Para obter mais informações sobre integridade referencial no DMS, consulte Trabalho com índices, acionadores e restrições de integridade referencial no tópico. Aprimoramento do desempenho de uma migração do AWS DMS

Valide se as tabelas de origem no escopo da tarefa têm restrições em cascata

Chave da API: `mariadb-check-cascade-constraints`

Essa avaliação de pré-migração verifica se alguma das tabelas de origem do MariaDB tem restrições em cascata. As restrições em cascata não são migradas ou replicadas pelas tarefas do DMS, porque o MariaDB não registra as alterações desses eventos no log binário. Embora AWS DMS não ofereça suporte a essas restrições, você pode usar soluções alternativas para destinos de bancos de dados relacionais.

Para obter informações sobre o suporte a restrições de cascata e outras restrições, consulte o tópico Solução Índices, chaves estrangeiras ou atualizações ou exclusões em cascata não migrados de problemas de migração. AWS DMS

Validar se as tabelas de origem no escopo da tarefa geraram colunas

Chave da API: `mariadb-check-generated-columns`

Essa avaliação de pré-migração verifica se alguma das tabelas de origem do MariaDB gerou colunas. As tarefas do DMS não migram nem replicam as colunas geradas.

Para obter informações sobre como migrar as colunas geradas, consulte???.

Valide se os valores de tempo limite são apropriados para uma fonte do MariaDB

Chave da API: `mariadb-check-source-network-parameter`

Essa avaliação de pré-migração verifica se o endpoint de origem MariaDB de uma tarefa tem as configurações `wait_timeout` e definidas para `net_read_timeout` pelo `net_wait_timeout` menos 300 segundos. Isso é necessário para evitar desconexões durante a migração.

Para ter mais informações, consulte Conexões a uma instância de destino MySQL são desconectadas durante uma tarefa.

Valide se os valores de tempo limite são apropriados para um destino do MariaDB

Chave da API: `mariadb-check-target-network-parameter`

Essa avaliação de pré-migração verifica se o endpoint de destino do MariaDB de uma tarefa tem `net_read_timeout` as `net_wait_timeout` configurações `wait_timeout` e definidas para pelo menos 300 segundos. Isso é necessário para evitar desconexões durante a migração.

Para ter mais informações, consulte Conexões a uma instância de destino MySQL são desconectadas durante uma tarefa.

Avaliações do PostgreSQL

Esta seção descreve avaliações individuais de pré-migração para tarefas de migração que usam um endpoint de origem do PostgreSQL.

Tópicos

- Validar se a versão do banco de dados de origem é compatível com o DMS para migração
- Validar o logical_decoding_work_mem parâmetro no banco de dados de origem
- Valide se o banco de dados de origem tem alguma transação de longa duração
- Validar o parâmetro do banco de dados de origem max_slot_wal_keep_size
- Verifique se o parâmetro do banco de dados de origem postgres-check-maxwalsenders está definido para oferecer suporte ao CDC.
- Verifique se o banco de dados de origem está configurado para PGLOGICAL
- Validar se a chave primária da tabela de origem é do tipo de dados LOB
- Validar se a tabela de origem tem uma chave primária
- Valide se as transações preparadas estão presentes no banco de dados de origem
- Valide se wal_sender_timeout está definido com um valor mínimo exigido para oferecer suporte ao DMS CDC
- Validar se wal_level está definido como lógico no banco de dados de origem

Validar se a versão do banco de dados de origem é compatível com o DMS para migração

Chave da API: `postgres-check-dbversion`

Essa avaliação de pré-migração verifica se a versão do banco de dados de origem é compatível com o. AWS DMS

Validar o **`logical_decoding_work_mem`** parâmetro no banco de dados de origem

Chave da API: `postgres-check-for-logical-decoding-work-mem`

Essa avaliação de pré-migração recomenda ajustar o `logical_decoding_work_mem` parâmetro no banco de dados de origem. Em um banco de dados altamente transacional em que você pode ter transações de longa duração ou muitas subtransações, isso pode resultar no aumento do consumo de memória de decodificação lógica e na necessidade de transferência para o disco. Isso resulta em alta latência na fonte do DMS durante a replicação. Nesses cenários, talvez seja necessário ajustar`logical_decoding_work_mem`. Esse parâmetro é suportado nas versões 13 e posteriores do PostgreSQL.

Valide se o banco de dados de origem tem alguma transação de longa duração

Chave da API: `postgres-check-longrunningtxn`

Essa avaliação de pré-migração verifica se o banco de dados de origem tem alguma transação de longa execução que durou mais de 10 minutos. O início da tarefa pode falhar porque, por padrão, o DMS verifica se há transações abertas ao iniciar a tarefa.

Validar o parâmetro do banco de dados de origem **`max_slot_wal_keep_size`**

Chave da API: `postgres-check-maxslot-wal-keep-size`

Essa avaliação de pré-migração verifica o valor configurado para. `max_slot_wal_keep_size` Quando `max_slot_wal_keep_size` definido como um valor não padrão, a tarefa do DMS pode falhar devido à remoção dos arquivos WAL necessários.

Verifique se o parâmetro do banco de dados de origem **`postgres-check-maxwalsenders`** está definido para oferecer suporte ao CDC.

Chave da API: `postgres-check-maxwalsenders`

Essa avaliação de pré-migração verifica o valor configurado `max_wal_senders` no banco de dados de origem. O DMS `max_wal_senders` precisa ser definido como maior que 1 para suportar o Change Data Capture (CDC).

Verifique se o banco de dados de origem está configurado para **PGLOGICAL**

Chave da API: `postgres-check-pglogical`

Essa avaliação de pré-migração verifica se o `shared_preload_libraries` valor está definido como compatível com `pglogical` o CDCPGLOGICAL. Observe que você pode ignorar essa avaliação se estiver planejando usar a decodificação de teste para replicação lógica.

Validar se a chave primária da tabela de origem é do tipo de dados LOB

Chave da API: `postgres-check-pk-lob`

Essa avaliação de pré-migração verifica se a chave primária de uma tabela é do tipo de dados Large Object (LOB). O DMS não oferece suporte à replicação se a tabela de origem tiver uma coluna LOB como chave primária.

Validar se a tabela de origem tem uma chave primária

Chave da API: `postgres-check-pk`

Essa avaliação de pré-migração verifica se existem chaves primárias para as tabelas usadas no escopo da tarefa. O DMS não oferece suporte à replicação de tabelas sem chaves primárias, a menos que a identidade da réplica esteja definida `full` na tabela de origem.

Valide se as transações preparadas estão presentes no banco de dados de origem

Chave da API: `postgres-check-preparedtxn`

Essa avaliação de pré-migração verifica se há alguma transação preparada presente no banco de dados de origem. A criação do slot de replicação pode parar de responder se houver alguma transação preparada no banco de dados de origem.

Valide se **`wal_sender_timeout`** está definido com um valor mínimo exigido para oferecer suporte ao DMS CDC

Chave da API: `postgres-check-walsenderstimeout`

Essa avaliação de pré-migração verifica se `wal_sender_timeout` está definida para um mínimo de 10.000 milissegundos (10 segundos). Uma tarefa do DMS com CDC requer um mínimo de 10.000 milissegundos (10 segundos) e falhará se o valor for menor que 10000.

Validar se **`wal_level`** está definido como lógico no banco de dados de origem

Chave da API: `postgres-check-wallevel`

Essa avaliação de pré-migração verifica se `wal_level` está definida como lógica. Para que o DMS CDC funcione, esse parâmetro precisa estar habilitado no banco de dados de origem.

Iniciando e visualizando avaliações de tipo de dados (Legacy)

> **Note**
>
> Esta seção descreve o conteúdo legado. Recomendamos que você use execuções de avaliação de pré-migração, descritas anteriormente em Especificar, iniciar e visualizar as execuções de avaliação de pré-migração.
> As avaliações do tipo de dados não estão disponíveis no console. Você só pode executar avaliações de tipo de dados usando a API ou a CLI e só pode visualizar os resultados de uma avaliação de tipo de dados no bucket S3 da tarefa.

Uma avaliação do tipo de dados identifica os tipos de dados em um banco de dados de origem que podem não ser migrados corretamente porque o destino não os suporta. Durante essa avaliação, AWS DMS lê os esquemas do banco de dados de origem para uma tarefa de migração e cria uma lista dos tipos de dados da coluna. Em seguida, ele compara essa lista com uma lista predefinida de tipos de dados suportados pelo. AWS DMS Se sua tarefa de migração tiver tipos de dados incompatíveis, AWS DMS cria um relatório que você pode examinar para ver se sua tarefa de migração tem algum tipo de dados incompatível. AWS DMS não cria um relatório se sua tarefa de migração não tiver nenhum tipo de dados incompatível.

AWS DMS suporta a criação de relatórios de avaliação de tipo de dados para os seguintes bancos de dados relacionais:

- Oracle
- SQL Server
- PostgreSQL
- MySQL
- MariaDB
- Amazon Aurora

Você pode iniciar e visualizar um relatório de avaliação do tipo de dados usando a CLI e os SDKs para acessar a API: AWS DMS

- A CLI utiliza o comando `start-replication-task-assessment` para iniciar uma avaliação do tipo de dados e utiliza o comando `describe-replication-task-assessment-results` para visualizar o relatório de avaliação do tipo de dados mais recente em formato JSON.
- A AWS DMS API usa a `StartReplicationTaskAssessment`operação para iniciar uma avaliação do tipo de dados e usa a `DescribeReplicationTaskAssessmentResults`operação para visualizar o relatório de avaliação do tipo de dados mais recente no formato JSON.

O relatório de avaliação de tipo de dados é um arquivo JSON único que inclui um resumo que lista os tipos de dados incompatíveis e a contagem de colunas de cada um. Ele inclui uma lista de estruturas de dados para cada tipo de dados incompatível, incluindo os esquemas, tabelas e colunas que têm o tipo de dados incompatível. É possível utilizar o relatório para modificar os tipos de dados de origem e melhorar o sucesso da migração.

Há dois níveis de tipos de dados incompatíveis. Os tipos de dados que aparecem no relatório como incompatíveis não podem ser migrados. Os tipos de dados que aparecem no relatório como parcialmente compatíveis podem ser convertidos em outro tipo de dados, mas não serem migrados como o esperado.

O exemplo a seguir mostra um exemplo de relatório de avaliação de tipo de dados que pode ser visualizado.

```
{
    "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 armazena as avaliações de tipo de dados mais recentes e anteriores em um bucket do Amazon S3 criado AWS DMS por em sua conta. O nome do bucket do Amazon S3 tem o seguinte formato, em que *customerId* é seu ID de cliente e *customerDNS* é um identificador interno.

```
dms-customerId-customerDNS
```

> **Note**
>
> Por padrão, é possível criar até 100 buckets do Amazon S3 em cada uma das suas contas da AWS . Como AWS DMS cria um bucket em sua conta, certifique-se de que ele não exceda seu limite de bucket. Caso contrário, a avaliação de tipo de dados falhará.

Todos os relatórios de avaliação de tipo de dados de uma determinada tarefa de migração são armazenados em uma pasta do bucket nomeada com o identificador da tarefa. O nome do arquivo de cada relatório é a data da avaliação do tipo de dados no formato yyyy-mm-dd-hh-mm. Para visualizar e comparar relatórios de avaliação de tarefas anteriores, utilize o console de gerenciamento do Amazon S3.

AWS DMS também cria uma função AWS Identity and Access Management (IAM) para permitir o acesso ao bucket do S3 criado para esses relatórios. O nome do perfil é `dms-access-for-tasks`. O perfil utiliza a política `AmazonDMSRedshiftS3Role`. Se ocorrer um ResourceNotFoundFaulterro durante a execução`StartReplicationTaskAssessment`, consulte a ResourceNotFoundFault seção Solução de problemas para obter informações sobre como criar a `dms-access-for-tasks` função manualmente.

A avaliação da solução de problemas é

A seguir, você encontrará tópicos sobre a solução de problemas com a execução de relatórios de avaliação com AWS Database Migration Service. Esses tópicos podem ajudá-lo a resolver problemas comuns.

Tópicos

- ResourceNotFoundFault ao correr StartReplicationTaskAssessment

ResourceNotFoundFault ao correr StartReplicationTaskAssessment

Você pode encontrar a seguinte exceção ao executar a StartReplicationTaskAssessmentação.

```
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
```

Se você encontrar essa exceção, crie a dms-access-for-tasksfunção fazendo o seguinte:

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles.
3. Escolha Criar Perfil.
4. Na página Selecionar entidade confiável, em Tipo de entidade confiável, escolha Política de confiança personalizada.
5. Cole o seguinte JSON no editor, substituindo o texto existente.

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

A política anterior concede a `sts:AssumeRole` permissão para AWS DMS. Quando você adiciona a política AmazonDMSRedshifts3Role, o DMS pode criar o bucket do S3 na sua conta e colocar os resultados da avaliação do tipo de dados nesse bucket do S3.

6. Escolha Próximo.
7. Na página Adicionar permissões, pesquise e adicione a política AmazonDMSRedshifts3Role. Escolha Próximo.
8. Na página Nome, revisão e criação, dê um nome à função dms-access-for-tasks. Selecione Criar função.

Especificar dados complementares para configurações de tarefa

Quando você cria ou modifica uma tarefa de replicação para endpoints do AWS DMS, a tarefa pode exigir informações adicionais para executar a migração. Você pode especificar essas informações adicionais utilizando uma opção no console do DMS. Ou você pode especificá-la utilizando o parâmetro `TaskData` para a operação de API do DMS `CreateReplicationTask` ou `ModifyReplicationTask`.

Se o endpoint de destino for o Amazon Neptune, será necessário especificar dados de mapeamento suplementares ao mapeamento de tabela. Esses dados de mapeamento suplementares especificam como converter dados relacionais de origem em dados do grafo de destino que um banco de dados pode consumir. Nesse caso, você pode usar um dos dois formatos possíveis. Para obter mais informações, consulte Especificar regras de mapeamento de grafos utilizando Gremlin e R2RML para o Amazon Neptune como destino.

Monitoramento de tarefas do AWS DMS

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance do AWS DMS e de suas soluções da AWS. É necessário coletar dados de monitoramento de todas as partes de sua solução da AWS para depurar uma falha de vários pontos com mais facilidade, caso ocorra. A AWS fornece várias ferramentas para monitorar seus recursos e tarefas do AWS DMS e responder a possíveis incidentes:

Eventos e notificações do AWS DMS

O AWS DMS utiliza o Amazon Simple Notification Service (Amazon SNS) para fornecer notificações quando ocorre um evento do AWS DMS, por exemplo, a criação ou a exclusão de uma instância de replicação. O AWS DMS agrupa eventos em categorias que você pode assinar, para ser notificado quando ocorre um evento nessa categoria. Por exemplo, se você assinar a categoria Criação de uma instância de replicação específica, será notificado sempre que ocorrer um evento relacionado à criação que afete a instância de replicação. É possível trabalhar com essas notificações em qualquer formato compatível com o Amazon SNS em uma região da AWS, como uma mensagem de e-mail, uma mensagem de texto ou uma chamada para um endpoint HTTP. Para obter mais informações, consulte Como trabalhar com eventos e notificações do Amazon SNS no AWS Database Migration Service.

Status da tarefa

É possível monitorar o andamento de uma tarefa verificando o status da tarefa e monitorando a tabela de controle da tarefa. O status da tarefa indica a condição de uma tarefa do AWS DMS e seus recursos associados. Ele inclui indicações se a tarefa está sendo criada, iniciada, está em execução ou interrompida. Ele também inclui o estado atual das tabelas que a tarefa está migrando, por exemplo, se uma carga completa de uma tabela foi iniciada ou está em andamento e detalhes como o número de inserções, exclusões e atualizações que ocorreram para a tabela. Para obter mais informações sobre a condição de recurso de tarefa e tarefa de monitoramento, consulte Status da tarefa e Estado da tabela durante as tarefas. Para obter mais informações sobre tabelas de controle, consulte Configurações de tarefa de tabela de controle.

Alarmes e logs do Amazon CloudWatch

Ao utilizar os alarmes do Amazon CloudWatch, você observa uma ou mais métricas de tarefa em um período especificado. Se a métrica ultrapassar um controle de utilização especificado, uma notificação será enviada para um tópico do Amazon SNS. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. Em vez disso, o estado precisa ter

mudado e ser mantido por um determinado número de períodos. O AWS DMS também utiliza o CloudWatch para registrar em log as informações de tarefas durante o processo de migração. É possível utilizar a AWS CLI ou a API do AWS DMS para visualizar informações sobre os logs de tarefas. Para obter mais informações sobre como utilizar o CloudWatch com o AWS DMS, consulte Monitoramento de tarefas de replicação utilizando o Amazon CloudWatch. Para obter mais informações sobre as métricas de monitoramento do AWS DMS, consulte Métricas do AWS Database Migration Service. Para obter mais informações sobre como utilizar os logs de tarefas do AWS DMS, consulte Visualização e gerenciamento dos logs de tarefas do AWS.

Logs do Time Travel

Para registrar em log e depurar tarefas de replicação, é possível utilizar o Time Travel do AWS DMS. Nessa abordagem, você utiliza o Amazon S3 para armazenar logs e criptografá-los utilizando as chaves de criptografia. É possível recuperar os logs do S3 utilizando filtros de data e hora e visualizar, baixar e ofuscar os logs conforme necessário. Ao fazer isso, é possível “voltar no tempo” para investigar as atividades do banco de dados.

É possível utilizar o Time Travel com os endpoints de origem do PostgreSQL compatível com o DMS e com os endpoints de destino do PostgreSQL e do MySQL compatíveis com o DMS. É possível ativar o Time Travel somente para tarefas de carga máxima e de CDC e para tarefas somente de CDC. Para ativar o Time Travel ou modificar qualquer configuração existente do Time Travel, interrompa a tarefa.

Para obter mais informações sobre os logs do Time Travel, consulte Configurações de tarefa do Time Travel. Para obter as práticas recomendadas para a utilização dos logs do Time Travel, consulte Solução de problemas de tarefas de replicação com o Time Travel.

AWS CloudTrailLogs do

O AWS DMS é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, por um perfil do IAM ou por um serviço da AWS no AWS DMS. O CloudTrail captura todas as chamadas de API para o AWS DMS como eventos, incluindo as chamadas no console do AWS DMS e as chamadas de código para operações da API do AWS DMS. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o AWS DMS. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). utilizando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o AWS DMS, o endereço IP no qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, além de detalhes adicionais. Para obter mais informações, consulte Registrar em log chamadas de API do AWS DMS com o AWS CloudTrail.

Logs de banco de dados

É possível exibir, fazer download e observar logs de banco de dados para seus endpoints de tarefas utilizando o AWS Management Console, a AWS CLI, ou a API para seu serviço de banco de dados da AWS. Para obter mais informações, consulte a documentação do seu serviço de banco de dados na Documentação da AWS.

Para obter mais informações, consulte os tópicos a seguir.

Tópicos

- Status da tarefa
- Estado da tabela durante as tarefas
- Monitoramento de tarefas de replicação utilizando o Amazon CloudWatch
- Métricas do AWS Database Migration Service
- Visualização e gerenciamento dos logs de tarefas do AWS
- Registrar em log chamadas de API do AWS DMS com o AWS CloudTrail
- Registro em log de contexto do AWS DMS

Status da tarefa

O status da tarefa indica a condição da tarefa. A tabela a seguir mostra os possíveis status que uma tarefa pode ter:

Status da tarefa	Descrição
Criando	O AWS DMS está criando a tarefa.
Running	A tarefa está executando as ações de migração especific adas.
Stopped	A tarefa foi interrompida.
Stopping	A tarefa está sendo interrompida. Em geral, esta é uma indicação de intervenção do usuário na tarefa.

Status da tarefa	Descrição
Deleting	A tarefa está sendo excluída, normalmente a partir de uma solicitação de intervenção do usuário.
Failed	A tarefa falhou. Para obter mais informações, consulte os arquivos de log da tarefa.
Erro	A tarefa foi interrompida devido a um erro. Uma breve descrição do erro da tarefa é fornecida na última seção de mensagens de falha da guia Visão geral.
Em execução com erros	A tarefa está sendo executada com um status de erro. Isso geralmente indica que uma ou mais tabelas na tarefa não puderam ser migradas. A tarefa continua carregando outras tabelas de acordo com as regras de seleção.
Starting	A tarefa está se conectando à instância de replicação e aos endpoints de origem e de destino. Todos os filtros e as transformações estão sendo aplicados.
Ready	A tarefa está pronta para ser executada. Este status geralmente ocorre depois do status “Creating”.
Modifying	A tarefa está sendo alterada, normalmente devido a uma ação do usuário que modificou as definições da tarefa.
Movendo	A tarefa está em processo de transferência para outra instância de replicação. A replicação permanece nesse estado até que a transferência seja concluída. A exclusão da tarefa é a única operação permitida na tarefa de replicação enquanto ela está sendo movida.
Falha na movimentação	A movimentação da tarefa falhou por algum motivo, como não haver espaço de armazenamento suficiente na instância de replicação de destino. Quando uma tarefa de replicação está nesse estado, ela pode ser iniciada, modificada, movida ou excluída.

Status da tarefa	Descrição
no dispositivo	A migração do banco de dados especificada para essa tarefa está sendo testada em resposta à execução da operação StartReplicationTaskAssessmentRun ou StartReplicationTaskAssessment.

A barra de status de tarefa fornece uma estimativa do andamento da tarefa. A qualidade dessa estimativa depende da qualidade das estatísticas de tabela do banco de dados de origem: quanto melhores as estatísticas de tabela, mais precisa a estimativa. Para tarefas com apenas uma tabela que não tem estatísticas sobre linhas estimadas, não é possível fornecer qualquer estimativa sobre a porcentagem de conclusão. Nesse caso, o estado da tarefa e a indicação de linhas carregadas podem ser utilizados para confirmar que a tarefa realmente está sendo executada e progredindo.

Observe que a coluna "última atualização" do console do DMS indica somente a hora em que o AWS DMS atualizou o registro das estatísticas de uma tabela pela última vez. Ela não indica a hora da última atualização da tabela.

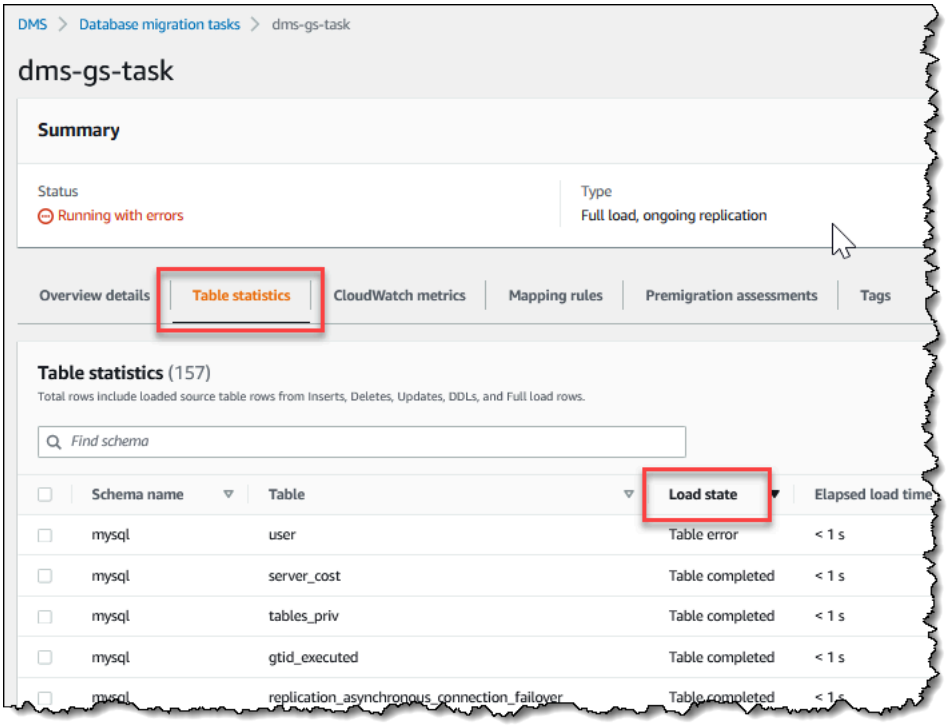
Além de utilizar o console do DMS, é possível gerar uma descrição das tarefas de replicação atuais, incluindo o status da tarefa, utilizando o comando `aws dms describe-replication-tasks` na AWS CLI, conforme mostrado no exemplo a seguir.

```
{
    "ReplicationTasks": [
        {
            "ReplicationTaskIdentifier": "moveit2",
            "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
            "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "MigrationType": "full-load",
            "TableMappings": ...output omitted... ,
            "ReplicationTaskSettings": ...output omitted... ,
            "Status": "stopped",
            "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
            "ReplicationTaskCreationDate": 1590524772.505,
            "ReplicationTaskStartDate": 1590619805.212,
```

```
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
            "ReplicationTaskStats": {
                "FullLoadProgressPercent": 100,
                "ElapsedTimeMillis": 0,
                "TablesLoaded": 0,
                "TablesLoading": 0,
                "TablesQueued": 0,
                "TablesErrored": 0,
                "FreshStartDate": 1590619811.528,
                "StartDate": 1590619811.528,
                "StopDate": 1590619842.068
            }
        }
    ]
}
```


Estado da tabela durante as tarefas

O console do AWS DMS atualiza as informações sobre o estado das tabelas durante a migração. A tabela a seguir mostra os possíveis valores para o estado:



Estado	Descrição
Table does not exist	O AWS DMS não encontrou a tabela no endpoint de origem.
Before load	O processamento de carga completa foi ativado, mas ainda não foi iniciado.
Full load	O processamento de carga completa está em andamento .
Table completed	O carregamento total foi concluído.
Table cancelled	O carregamento da tabela foi cancelado.
Erro de tabela	Ocorreu um erro durante o carregamento da tabela.

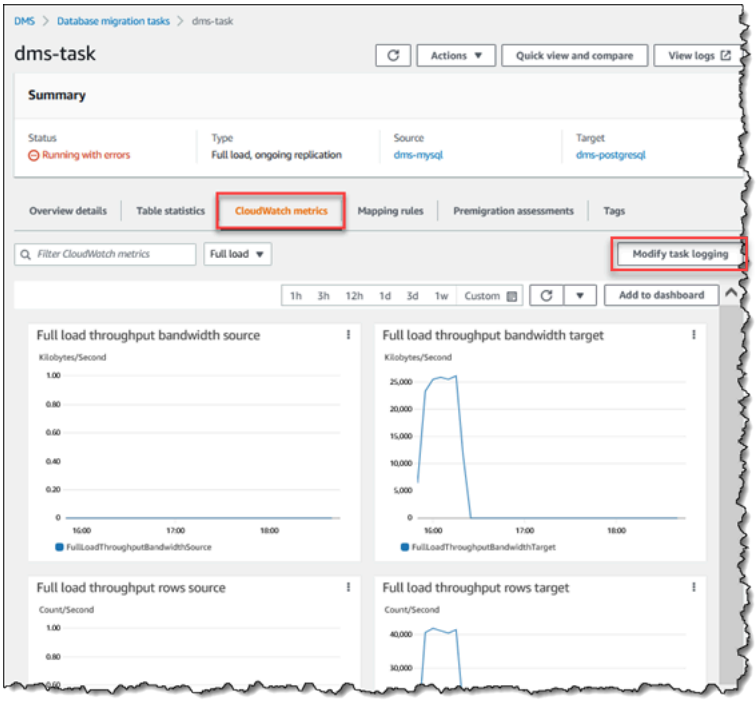
Monitoramento de tarefas de replicação utilizando o Amazon CloudWatch

É possível utilizar os alarmes ou eventos do Amazon CloudWatch para acompanhar mais de perto a migração. Para obter mais informações sobre o Amazon CloudWatch, consulte O que são o Amazon CloudWatch, o Amazon CloudWatch Events e o Amazon CloudWatch Logs? no Guia do usuário do Amazon CloudWatch. Observe que a utilização do Amazon CloudWatch pode ser cobrada.

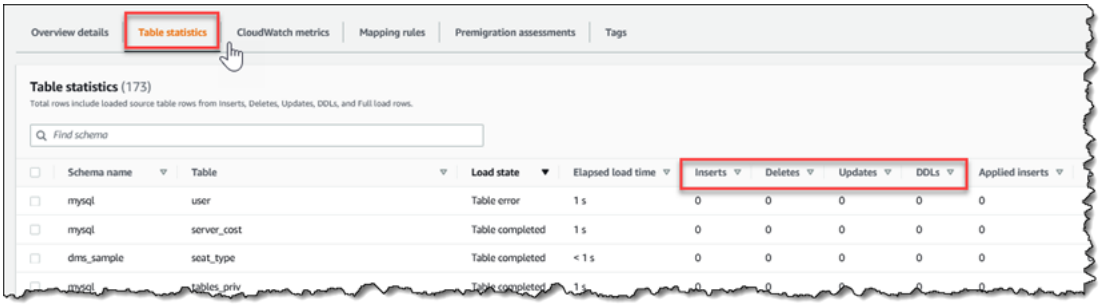
Se a tarefa de replicação não criar logs do CloudWatch, consulte AWS DMSnão cria CloudWatch registros no guia de solução de problemas.

O console do AWS DMS mostra as estatísticas básicas do CloudWatch de cada tarefa, incluindo o status das tarefas, porcentagens de conclusão, tempo decorrido e estatísticas de tabelas, como mostramos a seguir. Selecione a tarefa de replicação e selecione a guia Métricas do CloudWatch.

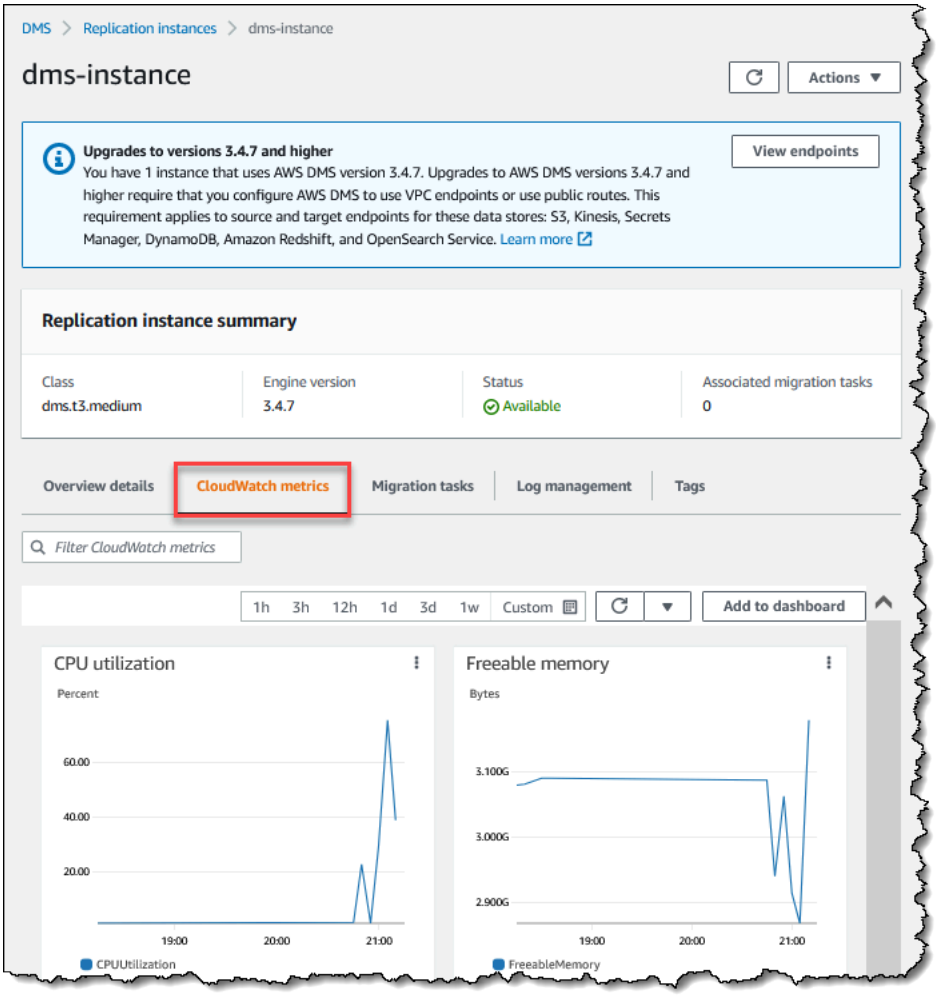
Para visualizar e modificar as configurações do log de tarefas do CloudWatch, escolha Modificar o registro em log de tarefas. Para obter mais informações, consulte Configurações de registro de tarefa.



O console do AWS DMS mostra as estatísticas de desempenho de cada tabela, incluindo o número de inserções, as exclusões e as atualizações, quando você seleciona a guia Estatísticas da tabela.



Além disso, se você selecionar uma instância de replicação na página Instância de replicação, poderá visualizar as métricas de desempenho da instância selecionando a guia Métricas do CloudWatch.



Métricas do AWS Database Migration Service

O AWS DMS fornece as seguintes estatísticas:

- Métricas do host: estatísticas de desempenho e de utilização do host de replicação, fornecidas pelo Amazon CloudWatch. Para obter uma lista completa das métricas disponíveis, consulte Métricas de instâncias de replicação.
- Métricas de tarefas de replicação: estatísticas das tarefas de replicação incluindo as alterações de entrada e confirmadas e a latência entre o host de replicação e os bancos de dados de origem e de destino. Para obter uma lista completa das métricas disponíveis, consulte Métricas de tarefas de replicação.
- Métricas de tabelas: estatísticas das tabelas que estão sendo migradas, incluindo o número de inserções, atualizações, exclusões e instruções DDL concluídas.

As métricas de tarefas são divididas em estatísticas entre o host de replicação e o endpoint de origem, e estatísticas entre o host de replicação e o endpoint de destino. É possível determinar a estatística total de uma tarefa somando as duas estatísticas relacionadas. Por exemplo, é possível determinar a latência total, ou o atraso de réplica, para uma tarefa combinando os valores CDCLatencySource e CDCLatencyTarget.

Os valores das métricas de uma tarefa podem ser influenciados pela atividade atual do banco de dados de origem. Por exemplo, se uma transação foi iniciada, mas não foi confirmada, a métrica CDCLatencySource continua aumentando até que a transação seja confirmada.

Para a instância de replicação, a métrica FreeableMemory requer esclarecimento. Memória passível de liberação não é indicação de memória real livre disponível. É a memória que está em uso no momento que pode ser liberada e utilizada para outros fins. É uma combinação de buffers e cache em uso na instância de replicação.

Embora a métrica FreeableMemory não reflita a memória real livre disponível, a combinação das métricas FreeableMemory e SwapUsage pode indicar se a instância de replicação está sobrecarregada.

Monitore as seguintes condições destas duas métricas:

- A métrica FreeableMemory se aproximando de zero.
- A métrica SwapUsage aumenta ou flutua.

Se você encontrar uma dessas duas condições, mude para uma instância de replicação maior. Você também deve reduzir o número e o tipo de tarefas em execução na instância de replicação. Tarefas de carregamento total exigem mais memória do que tarefas que apenas replicam alterações.

Para estimar aproximadamente os requisitos reais de memória para uma tarefa de migração do AWS DMS, é possível utilizar os parâmetros a seguir.

Colunas de LOB

O número médio de colunas de LOB em cada tabela no escopo de migração.

Número máximo de tabelas para carga em paralelo

O número máximo de tabelas que o AWS DMS carrega em paralelo em uma tarefa.

O valor padrão é 8.

Tamanho do bloco de LOB

O tamanho dos blocos de LOB, em kilobytes, que o AWS DMS utiliza para replicar dados no banco de dados de destino.

Taxa de confirmação durante carga máxima

O número máximo de registros que o AWS DMS pode transferir em paralelo.

O valor padrão é 10.000.

Tamanho do LOB

O tamanho máximo de um LOB individual, em kilobytes.

Tamanho da matriz em massa

O número máximo de linhas que são buscadas ou processadas pelo driver de endpoint. Esse valor depende das configurações do driver.

O valor padrão é 1,000.

Depois de determinar esses valores, é possível utilizar um dos métodos a seguir para estimar a quantidade de memória necessária para a tarefa de migração. Esses métodos dependem da opção escolhida para as Configurações da coluna LOB na tarefa de migração.

- Para Modo LOB completo, utilize a fórmula a seguir.

  ```
  Required memory = (LOB columns) * (Maximum number of tables to load in parallel) * (LOB chunk size) * (Commit rate during full load)
  ```

  Considere um exemplo em que as tabelas de origem incluem, em média, 2 colunas LOB, e o tamanho dos blocos de LOB é 64 KB. Se você utilizar os valores padrão para `Maximum number of tables to load in parallel` e `Commit rate during full load`, a quantidade de memória necessária para a tarefa será a seguinte.

  ```
  Required memory = 2 * 8 * 64 * 10,000 = 10,240,000 KB
  ```

  > **Note**
  >
  > Para reduzir o valor da Taxa de confirmação durante a carga máxima, abra o console do AWS DMS, escolha Tarefas de migração de banco de dados e crie ou modifique uma

tarefa. Expanda Configurações avançadas e insira seu valor para a Taxa de confirmação durante a carga máxima.

- Para Modo LOB limitado, utilize a fórmula a seguir.

  ```
  Required memory = (LOB columns) * (Maximum number of tables to load in parallel) * (LOB size) * (Bulk array size)
  ```

  Considere um exemplo em que as tabelas de origem incluem, em média, 2 colunas LOB, e o tamanho máximo de um LOB individual é 4.096 KB. Se você utilizar os valores padrão para `Maximum number of tables to load in parallel` e `Bulk array size`, a quantidade de memória necessária para a tarefa será a seguinte.

  ```
  Required memory = 2 * 8 * 4,096 * 1,000 = 65,536,000 KB
  ```

Para que o AWS DMS execute essas conversões de forma ideal, a CPU deve estar disponível quando as conversões ocorrerem. Sobrecarregar a CPU e não ter recursos de CPU suficientes pode resultar em migrações lentas. O AWS DMS pode consumir muita CPU, especialmente ao executar migrações e replicações heterogêneas, como migrações do Oracle para o PostgreSQL. A classe da instância de replicação C4 pode ser uma boa opção para essas situações. Para obter mais informações, consulte Escolhendo a instância de replicação AWS DMS certa para sua migração.

Métricas de instâncias de replicação

O monitoramento de instâncias de replicação inclui métricas do Amazon CloudWatch para as seguintes estatísticas:

Métrica	Descrição
AvailableMemory	Uma estimativa da memória disponível para iniciar novas aplicações, sem troca. Para obter mais informações, consulte o valor de `MemAvailable` na seção `/proc/memInfo` da página Linux man-pages. Unidade: bytes
CPUAllocated	A porcentagem de CPU alocada de forma máxima para a tarefa (0 significa que não há limite).

Métrica	Descrição
	O AWS DMS eleva essa métrica em relação às dimensões combinadas de `ReplicationInstanceIdentifer` e `ReplicationTaskIdentifier` no console do CloudWatch. Utilize a categoria `ReplicationInstanceIdentifier, ReplicationTaskIdentifier` para visualizar essa métrica. Unidades: percentual
CPUUtilization	O percentual de vCPU alocada (CPU virtual) alocada em utilização na instância no momento. Unidades: percentual
DiskQueueDepth	O número de solicitações de leitura/gravação (E/Ss) pendentes aguardando para acessar o disco. Unidade: contagem
FreeStorageSpace	A quantidade de espaço de armazenamento disponível. Unidade: bytes
FreeMemory	A quantidade de memória física disponível para utilização por aplicativos, cache de páginas e para as estruturas de dados de propriedade do kernel. Para obter mais informações, consulte o valor de `MemFree` na seção `/proc/memInfo` da página Linux man-pages. Unidade: bytes
FreeableMemory	A quantidade de memória de acesso aleatório disponível. Unidade: bytes

Métrica	Descrição
MemoryAllocated	A alocação máxima de memória para a tarefa (0 significa que não há limites). O AWS DMS eleva essa métrica em relação às dimensões combinadas de `ReplicationInstanceIdentifer` e `ReplicationTaskIdentifier` no console do CloudWatch. Utilize a categoria `ReplicationInstanceIdentifier, ReplicationTaskIdentifier` para visualizar essa métrica. Unidades: MiB
WriteIOPS	O número médio de operações de E/S de gravação de disco por segundo. Unidade: contagem/segundo
ReadIOPS	O número médio de operações E/S de leitura de disco por segundo. Unidade: contagem/segundo
WriteThroughput	O número médio de bytes gravados no disco por segundo. Unidade: bytes/segundo
ReadThroughput	O número médio de bytes lidos do disco por segundo. Unidade: bytes/segundo
WriteLatency	O tempo médio necessário por operação de I/O (saída) de disco. Unidade: milissegundos
ReadLatency	O tempo médio necessário por operação de I/O (entrada) de disco. Unidade: milissegundos
SwapUsage	A quantidade de espaço de troca utilizada na instância de replicação. Unidade: bytes

Métrica	Descrição
NetworkTransmitThroughput	O tráfego de rede de saída (transmissão) na instância de replicação, incluindo o tráfego do banco de dados cliente e o tráfego do AWS DMS utilizados para monitoramento e replicação. Unidade: bytes/segundo
NetworkReceiveThroughput	O tráfego de rede de entrada (recebimento) na instância de replicação, incluindo o tráfego do banco de dados cliente e o tráfego do AWS DMS utilizados para monitoramento e replicação. Unidade: bytes/segundo

Métricas de tarefas de replicação

O monitoramento de tarefas de replicação inclui métricas para as seguintes estatísticas.

Métrica	Descrição
FullLoadThroughputBandwidthTarget	Dados de saída transmitidos de uma carga máxima para o destino em KB por segundo.
FullLoadThroughputRowsTarget	As alterações de saída de uma carga máxima para o destino em linhas por segundo.
CDCIncomingChanges	O número total de eventos de alteração em um ponto no tempo aguardando para serem aplicados ao destino. Observe que isso não é o mesmo que uma medida da taxa de alteração de transação do endpoint de origem. Um número alto para essa métrica normalmente indica que o AWS DMS não consegue aplicar as alterações capturadas de maneira oportuna, provocando uma alta latência no destino.
CDCChangesMemorySource	O número de linhas que se acumulam na memória esperando para serem confirmadas a partir da origem. É possível visualizar essa métrica junto com CDCChangesDiskSource.

Métrica	Descrição
CDCChange sMemoryTarget	O número de linhas que se acumulam na memória esperando para serem confirmadas no destino. É possível visualizar essa métrica junto com CDCChangesDiskTarget.
CDCChangesDiskSour ce	O número de linhas que se acumulam no disco esperando para serem confirmadas a partir da origem. É possível visualizar essa métrica junto com CDCChangesMemorySource.
CDCChangesDiskTarg et	O número de linhas que se acumulam no disco esperando para serem confirmadas no destino. É possível visualizar essa métrica junto com CDCChangesMemoryTarget.
CDCThroughputBandw idthTarget	Dados de saída transmitidos para o destino em KB por segundo. CDCThroughputBandwidth registra dados de saída transmitidos em pontos de amostragem. Se nenhum tráfego de rede da tarefa for encontrado, o valor será zero. Como a CDC não emite transações prolongadas, o tráfego de rede pode não ser registrado.
CDCThroughputRowsS ource	As alterações das tarefas de entrada a partir de origem em linhas por segundo.
CDCThroughputRowsT arget	As alterações da tarefa de saída para o destino em linhas por segundo.

Métrica	Descrição
CDCLatencySource	O intervalo, em segundos, entre o último evento capturado no endpoint de origem e o timestamp atual do sistema da instância do AWS DMS. CDCLatencySource representa a latência entre a origem e a instância de replicação. CDCLatencySource alta significa que o processo de captura de alterações da origem está atrasado. Para identificar a latência em uma replicação contínua, é possível visualizar essa métrica junto com CDCLatencyTarget. Se CDCLatencySource e CDCLatencyTarget estiverem altas, investigue a CDCLatencySource primeiro. A CDCSourceLatency pode ser 0 quando não há atraso na replicaçã o entre a origem e a instância de replicação. A CDCSourceLatency também pode se tornar zero quando a tarefa de replicação tenta ler o próximo evento no log de transações da origem e não há novos eventos em comparação com a última leitura na origem. Quando isso ocorre, a tarefa redefine a CDCSourceLatency como 0.

Métrica	Descrição
CDCLatencyTarget	O intervalo, em segundos, entre o primeiro timestamp de evento em espera de confirmação no destino e o timestamp atual da instância do AWS DMS. A latência do destino é a diferença entre a hora do servidor da instância de replicação e o ID do evento não confirmad o mais antigo encaminhado para um componente de destino. Em outras palavras, a latência de destino é a diferença do timestamp entre a instância de replicação e o evento mais antigo aplicado, mas não confirmado pelo endpoint de TRG (99%). Quando a CDCLatenc yTarget está alta, isso indica que o processo de aplicação de eventos de alteração no destino está atrasado. Para identificar a latência em uma replicação contínua, é possível visualizar essa métrica junto com a CDCLatencySource. Se a CDCLatencyTarget estiver alta, mas a CDCLatencySource não estiver alta, investigue se: • Não existe nenhuma chave primária ou índice no destino • Os gargalos de recursos ocorrem no destino ou na instância de replicação • Os problemas de rede residem entre a instância de replicação e o destino
CPUUtilization	A porcentagem de CPU utilizada por uma tarefa em vários núcleos. A semântica da tarefa CPUUtilization é um pouco diferente da replicaçã o CPUUtilizaiton. Se uma vCPU for totalmente utilizada, isso indica 100%, mas se várias vCPUs estiverem em uso, o valor poderá estar acima de 100%. Unidades: percentual
SwapUsage	A quantidade de troca utilizada no host. Unidade: bytes

Métrica	Descrição
MemoryUsage	O grupo de controle (cgroup) memory.usage_in_bytes consumido por uma tarefa. O DMS utiliza cgroups para controlar o uso dos recursos do sistema, como memória e CPU. Essa métrica indica a utilização de memória de uma tarefa em megabytes dentro do cgroup alocado para essa tarefa. Os limites do cgroup são baseados nos recursos disponíveis para a classe da instância de replicação do DMS. memory.usage_in_bytes consiste no tamanho do conjunto residente (RSS), no cache e em componentes e troca da memória. O sistema operacional pode recuperar a memória cache, se necessário. É recomendável monitorar também a métrica da instância de replicaçã o, AvailableMemory. O AWS DMS eleva essa métrica em relação às dimensões combinada s de `ReplicationInstanceIdentifer` e `Replicati onTaskIdentifier` no console do CloudWatch. Utilize a categoria `ReplicationInstanceIdentifier, Replicati onTaskIdentifier` para visualizar essa métrica.

Visualização e gerenciamento dos logs de tarefas do AWS

É possível utilizar o Amazon CloudWatch para registrar em log as informações das tarefas durante um processo de migração do AWS DMS. O registro em log é ativado quando você seleciona as configurações de tarefa. Para obter mais informações, consulte Configurações de registro de tarefa.

Para visualizar logs de uma tarefa executada, siga estas etapas:

1. Abra o console do AWS DMS e selecione Tarefas de migração de banco de dados no painel de navegação. A caixa de diálogo "Tarefas de migração de banco de dados" é exibida.
2. Selecione o nome da tarefa. A caixa de diálogo "Detalhes da visão geral" é exibida.
3. Localize a seção Logs de tarefa de migração e escolha Visualizar CloudWatch Logs.

Além disso, é possível utilizar a AWS CLI ou a API do AWS DMS para visualizar informações sobre logs de tarefas. Para fazer isso, utilize o comando `describe-replication-instance-task-logs` da AWS CLI ou a ação `DescribeReplicationInstanceTaskLogs` da API do AWS DMS.

Por exemplo, o seguinte comando da AWS CLI mostra os metadados do log de tarefas no formato JSON.

```
$ aws dms describe-replication-instance-task-logs \
   --replication-instance-arn arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAY
```

O seguinte é um exemplo de resposta do comando.

```
{
    "ReplicationInstanceTaskLogs": [
        {
            "ReplicationTaskArn": "arn:aws:dms:us-
east-1:237565436:task:MY34U6Z4MSY52GRTIX3O4AY",
            "ReplicationTaskName": "mysql-to-ddb",
            "ReplicationInstanceTaskLogSize": 3726134
        }
    ],
    "ReplicationInstanceArn": "arn:aws:dms:us-east-1:237565436:rep:CDSFSFSFFFSSUFCAY"
}
```

Nessa resposta, há um único log de tarefas (`mysql-to-ddb`) associado à instância de replicação. O tamanho do log é 3.726.124 bytes.

É possível utilizar as informações retornadas por `describe-replication-instance-task-logs` para diagnosticar e solucionar problemas com logs de tarefas. Por exemplo, se você ativar o log de depuração detalhada de uma tarefa, o log de tarefas crescerá rapidamente, potencialmente consumindo todo o armazenamento disponível na instância de replicação, e fazendo com que o status da instância seja alterado para `storage-full`. Descrevendo os logs de tarefas, é possível determinar de quais você não precisa mais e excluí-los, liberando espaço de armazenamento.

Para excluir os logs de tarefas de uma tarefa, defina a tarefa com `DeleteTaskLogs` definido como true. Por exemplo, o JSON a seguir exclui os logs de tarefas ao modificar uma tarefa utilizando o comando `modify-replication-task` da AWS CLI ou a ação `ModifyReplicationTask` da API do AWS DMS.

```
{
   "Logging": {
       "DeleteTaskLogs":true
   }
}
```

Registrar em log chamadas de API do AWS DMS com o AWS CloudTrail

O AWS DMS é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, um perfil ou um serviço da AWS no AWS DMS. O CloudTrail captura todas as chamadas de API para o AWS DMS como eventos, incluindo as chamadas no console do AWS DMS e as chamadas de código para operações da API do AWS DMS. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o AWS DMS. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). utilizando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o AWS DMS, o endereço IP no qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, além de detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

Informações do AWS DMS no CloudTrail

O CloudTrail é ativado na sua conta da AWS quando ela é criada. Quando ocorre uma atividade no AWS DMS, essa atividade é registrada em um evento do CloudTrail com outros eventos de produtos da AWS em Event history (Histórico de eventos). É possível visualizar, pesquisar e baixar os eventos recentes na sua conta da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos do AWS DMS, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da AWS. A trilha registra em log eventos de todas as regiões da AWS na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- Serviços e integrações compatíveis com o CloudTrail
- Configurar notificações do Amazon SNS para o CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões da AWS e Receber arquivos de log do CloudTrail de várias contas

Todas as ações do AWS DMS são registradas pelo CloudTrail e são documentadas na Referência de API do AWS Database Migration Service. Por exemplo, as chamadas para as ações `CreateReplicationInstance`, `TestConnection` e `StartReplicationTask` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas sobre entradas de arquivos de log do AWS DMS

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `RebootReplicationInstance`.

```
{
"eventVersion": "1.05",
"userIdentity": {
    "type": "AssumedRole",
```

```
        "principalId": "AKIAIOSFODNN7EXAMPLE:johndoe",
        "arn": "arn:aws:sts::123456789012:assumed-role/admin/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "ASIAYFI33SINADOJJEZW",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-08-01T16:42:09Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/admin",
                "accountId": "123456789012",
                "userName": "admin"
            }
        }
    },
    "eventTime": "2018-08-02T00:11:44Z",
    "eventSource": "dms.amazonaws.com",
    "eventName": "RebootReplicationInstance",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.64",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "forceFailover": false,
        "replicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:EX4MBJ2NMRDL3BMAYJOXUGYPUE"
    },
    "responseElements": {
        "replicationInstance": {
            "replicationInstanceIdentifier": "replication-instance-1",
            "replicationInstanceStatus": "rebooting",
            "allocatedStorage": 50,
            "replicationInstancePrivateIpAddresses": [
                "172.31.20.204"
            ],
            "instanceCreateTime": "Aug 1, 2018 11:56:21 PM",
            "autoMinorVersionUpgrade": true,
            "engineVersion": "2.4.3",
            "publiclyAccessible": true,
            "replicationInstanceClass": "dms.t2.medium",
            "availabilityZone": "us-east-1b",
```

```
            "kmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
f7bc0f8e-1a3a-4ace-9faa-e8494fa3921a",
            "replicationSubnetGroup": {
                "vpcId": "vpc-1f6a9c6a",
                "subnetGroupStatus": "Complete",
                "replicationSubnetGroupArn": "arn:aws:dms:us-
east-1:123456789012:subgrp:EDHRVRBAAAPONQAIYWP4NUW22M",
                "subnets": [
                    {
                        "subnetIdentifier": "subnet-cbfff283",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1b"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-d7c825e8",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1e"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-6746046b",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1f"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-bac383e0",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1c"
                        },
                        "subnetStatus": "Active"
                    },
                    {
                        "subnetIdentifier": "subnet-42599426",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1d"
                        },
                        "subnetStatus": "Active"
                    },
                    {
```

```
                        "subnetIdentifier": "subnet-da327bf6",
                        "subnetAvailabilityZone": {
                            "name": "us-east-1a"
                        },
                        "subnetStatus": "Active"
                    }
                ],
                "replicationSubnetGroupIdentifier": "default-vpc-1f6a9c6a",
                "replicationSubnetGroupDescription": "default group created by console
 for vpc id vpc-1f6a9c6a"
            },
            "replicationInstanceEniId": "eni-0d6db8c7137cb9844",
            "vpcSecurityGroups": [
                {
                    "vpcSecurityGroupId": "sg-f839b688",
                    "status": "active"
                }
            ],
            "pendingModifiedValues": {},
            "replicationInstancePublicIpAddresses": [
                "18.211.48.119"
            ],
            "replicationInstancePublicIpAddress": "18.211.48.119",
            "preferredMaintenanceWindow": "fri:22:44-fri:23:14",
            "replicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:EX4MBJ2NMRDL3BMAYJOXUGYPUE",
            "replicationInstanceEniIds": [
                "eni-0d6db8c7137cb9844"
            ],
            "multiAZ": false,
            "replicationInstancePrivateIpAddress": "172.31.20.204",
            "patchingPrecedence": 0
        }
    },
    "requestID": "a3c83c11-95e8-11e8-9d08-4b8f2b45bfd5",
    "eventID": "b3c4adb1-e34b-4744-bdeb-35528062a541",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Registro em log de contexto do AWS DMS

O AWS DMS utiliza o registro em log de contexto para fornecer informações sobre uma migração em andamento. O registro em log de contexto grava informações, como as seguintes, no log de tarefas do CloudWatch:

- Informações sobre a conexão da tarefa com os bancos de dados de origem e de destino.
- Comportamento da tarefa de replicação. É possível utilizar os logs de tarefas para diagnosticar problemas de replicação.
- Instruções SQL sem dados que o AWS DMS executa nos bancos de dados de origem e de destino. É possível utilizar os logs do SQL para diagnosticar um comportamento de migração inesperado.
- Detalhes da posição no fluxo de cada evento da CDC.

O log de contexto só está disponível no AWS DMS versão 3.5.0 ou superior.

O AWS DMS ativa o log de contexto por padrão. Para controlar o log de contexto, defina a configuração da tarefa `EnableLogContext` como `true` ou `false` ou modificando a tarefa no console.

O AWS DMS grava as informações do log de contexto na tarefa de replicação do log do CloudWatch a cada três minutos. Verifique se a instância de replicação tem espaço suficiente para o log da aplicação. Para obter mais informações sobre como gerenciar logs de tarefas, consulte Visualização e gerenciamento dos logs de tarefas do AWS.

Tópicos

Tipos de objeto

O AWS DMS produz logs de contexto no CloudWatch para os seguintes tipos de objeto.

Tipo de objeto	Descrição
TABLE_NAME	Essas entradas do log contêm informações sobre as tabelas que estão em escopo com a regra atual de mapeamento de tarefas. É possível utilizar essas entradas para examinar os eventos de tabelas em um período específic o durante a migração.
SCHEMA_NAME	Essas entradas do log contêm informações sobre os esquemas utilizados pela regra atual de mapeamento de tarefas. É possível utilizar essas entradas para determinar qual esquema o AWS DMS está utilizado por um período específico durante a migração.
TRANSACTION_ID	Essas entradas contêm o ID da transação de cada alteração de DML/DDL capturada no banco de dados de origem. É possível utilizar essas entradas do log para determinar as alterações ocorridas durante uma determinada transação.
CONNECTION_ID	Essas entradas contêm o ID da conexão. É possível utilizar essas entradas do log para determinar qual conexão o AWS DMS utiliza para cada etapa da migração.
STATEMENT	Essas entradas contêm o código SQL utilizado para buscar, processar e aplicar cada alteração da migração.
STREAM_POSITION	Essas entradas contêm a posição no arquivo de log de transações de cada ação da migração no banco de dados de origem. O formato dessas entradas varia entre os tipos de mecanismo de banco de dados de origem. Também é possível utilizar essas informações

Tipo de objeto	Descrição
	para determinar a posição inicial de um ponto de verificação de recuperação ao configurar a replicação somente de CDC.

Exemplos de logs

Esta seção contém exemplos de registros de log que podem ser utilizados para monitorar a replicação e diagnosticar problemas de replicação.

Exemplos de logs de conexão

Esta seção contém exemplos de log que incluem IDs de conexão.

```
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)
```

Exemplos de logs de comportamento de tarefas

Esta seção contém exemplos de logs sobre o comportamento do log de tarefas de replicação. É possível utilizar essas informações para diagnosticar problemas de replicação, como uma tarefa no status `IDLE`.

Os logs de `SOURCE_CAPTURE` a seguir indicam que não há eventos disponíveis para leitura no arquivo de log do banco de dados de origem e que contêm registros de `TARGET_APPLY` que indicam que não há eventos recebidos dos componentes do AWS DMS CDC a serem aplicados ao banco de dados de destino. Esses eventos também contêm detalhes de contexto relacionados a eventos aplicados anteriormente.

```
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
 (?,?,?,?,?),
```

Exemplos de logs de instruções SQL

Esta seção contém exemplos de logs sobre instruções SQL executadas em bancos de dados de origem e de destino. As instruções SQL que você vê nos logs mostram somente a instrução SQL, não mostram os dados. O log de `TARGET_APPLY` a seguir mostra uma instrução `INSERT` executada no destino.

```
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
 (?,?,?,?,?),
```

Limitações

As limitações a seguir se aplicam ao log de contexto do AWS DMS:

- Embora o AWS DMS crie um log mínimo para todos os tipos de endpoint, o log de contexto extenso específico do mecanismo está disponível somente para os seguintes tipos de endpoint. É recomendável ativar o log de contexto ao utilizar esses tipos de endpoint.
  - MySQL
  - PostgreSQL
  - Oracle
  - Microsoft SQL Server
  - MongoDB/Amazon DocumentDB
  - Amazon S3

Como trabalhar com eventos e notificações do Amazon EventBridge no AWS Database Migration Service

É possível utilizar o Amazon EventBridge para fornecer notificação quando ocorre um evento no AWS DMS, por exemplo, a criação ou a exclusão de uma instância de replicação. O EventBridge recebe eventos e roteia a notificação de um evento conforme definido pelas regras dos eventos. É possível trabalhar com essas notificações em qualquer formato compatível com o Amazon EventBridge de uma região da AWS. Para obter mais informações sobre como utilizar o Amazon EventBridge, consulte O que é o Amazon EventBridge? no Guia do usuário do Amazon EventBridge.

> **Note**
>
> O trabalho com eventos do Amazon EventBridge é compatível com o AWS DMS versão 3.4.5 e superior.

O EventBridge recebe um evento, um indicador de uma alteração no ambiente do AWS DMS, e aplica uma regra para encaminhar o evento para um mecanismo de notificação. As regras fazem a correspondência entre os eventos e os mecanismos de notificação com base na estrutura do evento, chamada de padrão de evento.

O AWS DMS agrupa os eventos em categorias às quais é possível aplicar um regra de evento para que você seja notificado quando ocorrer um evento dessa categoria. Por exemplo, suponha que você aplique uma regra de evento do EventBridge à categoria Criação de uma determinada instância de replicação. Você será notificado sempre que ocorrer um evento relacionado à criação que afete a instância de replicação. Ao aplicar uma regra para uma categoria de Alteração na configuração de uma instância de replicação, você será notificado quando a configuração da instância de replicação for alterada. Para obter uma lista das categorias de eventos fornecidas pelo AWS DMS, consulte as seguintes categorias e mensagens de eventos do AWS DMS.

> **Note**
>
> Para permitir a publicação em events.amazonaws.com, atualize as políticas de acesso dos tópicos do Amazon SNS. Para obter mais informações, consulte Uso de políticas baseadas em recursos do Amazon EventBridge no Guia do usuário do Amazon EventBridge.

Para obter mais informações sobre como mover assinaturas de eventos para o Amazon EventBridge, consulte Migrar assinaturas de eventos ativas do DMS para o Amazon EventBridge.
Para obter mais informações sobre como utilizar mensagens de texto com o Amazon SNS, consulte Envio e recebimento de notificações por SMS utilizando o Amazon SNS.

Utilização das regras de eventos do Amazon EventBridge para o AWS DMS

O Amazon EventBridge envia notificações de eventos aos endereços fornecidos ao criar uma regra de eventos do EventBridge. Talvez você queira criar várias regras diferentes. Por exemplo, é possível criar uma regra que receba todas as notificações de eventos e outra que inclua somente eventos essenciais dos recursos de produção do DMS. Também é possível ativar ou desativar notificações de eventos no EventBridge.

Para criar regras do Amazon EventBridge que reajam a eventos do AWS DMS

- Execute as etapas descritas em Criação de regras do Amazon EventBridge que reagem a eventos no Guia do usuário do Amazon EventBridge e crie uma regra para os eventos do AWS DMS:

  a. Especifique uma ação de notificação a ser executada quando o EventBridge receber um evento que corresponda ao padrão de evento na regra. Quando encontra uma correspondência, o EventBridge envia o evento e invoca a ação definida na regra.

  b. Em Provedor de serviços, escolha AWS.

  c. Em Nome do serviço, selecione Database Migration Service (DMS).

Você começará a receber notificações de eventos.

O exemplo de JSON a seguir mostra um modelo de eventos do EventBridge para um serviço do AWS DMS.

```
{
   "version":"0",
   "id":"11a11b11-222b-333a-44d4-01234a5b67890",
   "detail-type":"DMS Replication Task State Change",
```

```
   "source":"aws.dms",
   "account":"0123456789012",
   "time":"1970-01-01T00:00:00Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:dms:us-east-1:012345678901:task:AAAABBBB0CCCCDDDDEEEEE1FFFF2GGG3FFFFFF3"
   ],
   "detail":{
      "type":"REPLICATION_TASK",
      "category":"StateChange",
      "eventType":"REPLICATION_TASK_STARTED",
      "eventId":"DMS-EVENT-0069",
      "resourceLink":"https://console.aws.amazon.com/dms/v2/home?region=us-
east-1#taskDetails/taskName",
      "detailMessage":"Replication task started, with flag = fresh start"
   }
}
```

Para obter uma lista das categorias e dos eventos sobre os quais você pode ser notificado, consulte a próxima seção.

Categorias e mensagens de eventos do AWS DMS

O AWS DMS gera um número significativo de eventos em categorias que podem ser identificadas. Cada categoria se aplica a um tipo de origem de instância de replicação ou de tarefa de replicação.

Tópicos

- Mensagens de evento de ReplicationInstance
- Mensagens de evento de ReplicationTask
- Mensagens de evento de replicação

Mensagens de evento de ReplicationInstance

A tabela a seguir mostra as categorias e os eventos possíveis para o tipo de origem ReplicationInstance.

Categoria	ID do evento	Descrição
Criação	DMS-EVENT-0067	A instância de replicação está sendo criada.
Exclusão	DMS-EVENT-0066	A instância de replicação está sendo excluída.
Alteração na configuração	DMS-EVENT-0012	A classe dessa instância de replicação está sendo alterada.
Alteração na configuração	DMS-EVENT-0018	O armazenamento da instância de replicação está sendo aumentado.
Alteração na configuração	DMS-EVENT-0024	A instância de replicação está fazendo a transição para uma configuração multi-AZ.
Alteração na configuração	DMS-EVENT-0030	A instância de replicação está fazendo a transição para uma configuração Single-AZ.
Manutenção	DMS-EVENT-0026	Está ocorrendo a manutenção off-line da instância de replicação. A instância de replicação não está disponível no momento.
Criação	DMS-EVENT-0005	A instância de replicação foi criada.
Exclusão	DMS-EVENT-0003	A instância de replicação foi excluída.
Alteração na configuração	DMS-EVENT-0014	A classe dessa instância de replicação foi alterada.

Categoria	ID do evento	Descrição
Alteração na configuração	DMS-EVENT-0017	O armazenamento da instância de replicação foi aumentado.
Alteração na configuração	DMS-EVENT-0025	A instância de replicação concluiu a transição para uma configuração multi-AZ.
Alteração na configuração	DMS-EVENT-0029	A instância de replicação concluiu a transição para uma configuração Single-AZ.
Manutenção	DMS-EVENT-0047	O software de gerenciamento na instância de replicação foi atualizado.
Manutenção	DMS-EVENT-0027	A manutenção off-line da instância de replicação está concluída. A instância de replicação agora está disponível.
Manutenção	DMS-EVENT-0068	A instância de replicação está em um estado que não pode ser atualizado.
Failover	DMS-EVENT-0034	Se você solicitar failover com muita frequência, ocorrerá esse evento em vez dos eventos normais de failover.
Falha	DMS-EVENT-0031	Instância de replicação colocada no estado %s.
Falha	DMS-EVENT-0036	Falha na instância de replicação devido a uma rede incompatível.
Falha	DMS-EVENT-0037	Quando o serviço não pode acessar a chave do KMS usada para criptografar o volume de dados.

Categoria	ID do evento	Descrição
Falha		A instância de replicação inseriu parâmetros incompatíveis.
Failover		Tempo limite de espera por um estado seguro esgotado para iniciar o failover solicitado pelo usuário
Failover	DMS-EVENT-0013	Failover iniciado para uma instância de replicação multi-AZ.
Failover	DMS-EVENT-0049	O failover foi concluído para uma instância de replicação multi-AZ.
Failover	DMS-EVENT-0050	A ativação multi-AZ foi iniciada.
Failover	DMS-EVENT-0051	Ativação multi-AZ concluída.
StateChange		Os logs de consultas gerais e lentas foram automaticamente alternados como %s
StateChange		O AWS DMS não pode acessar a chave de criptografia do KMS da instância de aplicação %s. Provavelmente, a chave está desativada ou o AWS DMS não pode acessá-la. Se isso continuar, a aplicação será colocada em um estado inacessível. Consulte a seção de solução de problemas na documentação do AWS DMS para obter mais detalhes.
StateChange		O AWS DMS não pode acessar a chave de criptografia do KMS da instância da aplicação %s.

Categoria	ID do evento	Descrição
StateChange		O Amazon DMS não pode acessar a chave de criptografia do KMS da instância da aplicação %s. A aplicação será colocada em um estado inacessível. Consulte a seção de solução de problemas na documentação do Amazon DMS para obter mais detalhes.
StateChange		Reinício da aplicação no HM como parte da criação da instância de replicação
StateChange		Desligamento da aplicação no HM como parte da exclusão da instância de replicação
Failover	DMS-EVENT-0015	Failover de multi-AZ para o modo de espera concluído.
LowStorage	DMS-EVENT-0007	O armazenamento gratuito da instância de replicação está baixo.
LowStorage		Os inodes alocados foram esgotados: escale o armazenamento para resolver

Mensagens de evento de ReplicationTask

A tabela a seguir mostra as categorias e os eventos possíveis para o tipo de origem ReplicationTask.

Categoria	ID do evento	Descrição
Falha	DMS-EVENT-0078	Falha na tarefa de replicação.
Falha	DMS-EVENT-0082	Falha em uma chamada para limpar os dados da tarefa.

Categoria	ID do evento	Descrição
Alteração de estado	DMS-EVENT-0081	Foi solicitado o recarregamento dos detalhes da tabela.
Alteração de estado		A tarefa de replicação foi copiada.
Alteração de estado		Falha na cópia da tarefa de replicação.
Alteração de estado		A tarefa de replicação foi movida.
Alteração de estado		Falha na movimentação da tarefa de replicação.
Alteração de estado		Falha na criação da tarefa de destino.
Alteração de estado		A execução da avaliação da tarefa de replicação foi iniciada.
Alteração de estado		A execução da avaliação da tarefa de replicação foi concluída com êxito.
Alteração de estado		A execução da avaliação da tarefa de replicação foi concluída com falha.
StateChange		A execução da avaliação da tarefa de replicação foi concluída com aviso.
StateChange		A execução da avaliação da tarefa de replicação foi concluída com erro.
StateChange		A execução da avaliação da tarefa de replicação %s foi cancelada.
StateChange		A execução da avaliação da tarefa de replicação %s foi excluída.

Categoria	ID do evento	Descrição
StateChange		A execução da avaliação da tarefa de replicação não provisionou recursos.
StateChange		Falha na tarefa de replicação.
Criação		A tarefa de replicação foi criada.
ConfigurationChange		Uma tarefa de replicação foi modificada.
Falha		Falha na tarefa de replicação.
StateChange	DMS-EVENT-0091	Leitura pausada, limite de arquivos de troca atingido.
StateChange	DMS-EVENT-0092	Leitura pausada, limite de uso de disco atingido.
StateChange	DMS-EVENT-0093	Leitura pausada, limite de uso de disco atingido.
StateChange	DMS-EVENT-0093	Leitura retomada.
StateChange	DMS-EVENT-0069	A tarefa de replicação foi iniciada com taskType: %s, startType: %s
StateChange	DMS-EVENT-0079	A replicação foi interrompida.
Exclusão	DMS-EVENT-0073	A tarefa de replicação foi excluída.

Mensagens de evento de replicação

A tabela a seguir mostra as categorias e os eventos possíveis para o tipo de origem Replicação.

Categoria	Descrição
Alteração de estado	Evento de aumentar a escala verticalmente da replicação do DMS.
Alteração de estado	Evento de reduzir a escala verticalmente da replicação do DMS.
Alteração de estado	Evento de ajuste da escala de replicação do DMS concluído.
Alteração de estado	A replicação do DMS foi criada.
Alteração de estado	A replicação do DMS está sendo inicializada.
Alteração de estado	A replicação do DMS está preparando os recursos para a coleta de metadados.
Alteração de estado	As conexões vinculadas à replicação do DMS estão sendo testadas.
Alteração de estado	A replicação do DMS está buscando metadados
Alteração de estado	A replicação do DMS está calculando a capacidade
Alteração de estado	A replicação do DMS está provisionando a capacidade
Alteração de estado	A replicação do DMS foi provisionada.
Alteração de estado	A replicação do DMS foi iniciada
Alteração de estado	A replicação do DMS está em execução.
Alteração de estado	A replicação do DMS está sendo interrompida.
Alteração de estado	A replicação do DMS foi interrompida.
Alteração de estado	A replicação do DMS está sendo modificada.
Alteração de estado	A replicação do DMS está sendo excluída.
Alteração de estado	A replicação do DMS está desprovisionando a capacidade
Alteração de estado	A replicação do DMS foi desprovisionada.
Falha	Falha na replicação do DMS.

Como trabalhar com eventos e notificações do Amazon SNS no AWS Database Migration Service

A partir do lançamento do AWS DMS 3.4.5 e com versões posteriores, é recomendável utilizar o Amazon EventBridge para fornecer notificações quando ocorre um evento no AWS DMS. Para obter mais informações sobre como utilizar eventos do EventBridge com o AWS DMS, consulte Como trabalhar com eventos e notificações do Amazon EventBridge no AWS Database Migration Service.

Movimentação de assinaturas de eventos para o Amazon EventBridge

É possível utilizar o seguinte comando da AWS CLI para migrar assinaturas de eventos ativos do DMS para o Amazon EventBridge, até 10 por vez.

```
update-subscriptions-to-event-bridge [--force-move | --no-force-move]
```

Por padrão, o AWS DMS só migra assinaturas de eventos ativos quando a instância de replicação está atualizada com o AWS DMS 3.4.5 ou superior. Para substituir esse comportamento padrão, utilize a opção `--force-move`. No entanto, alguns tipos de eventos podem não estar disponíveis ao utilizar o Amazon EventBridge se as instâncias de replicação não estiverem atualizadas.

Para executar o comando `update-subscriptions-to-event-bridge` da CLI, um usuário do AWS Identity and Access Management (IAM) deve ter as permissões de política a seguir.

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "SNS:GetTopicAttributes",
              "SNS:SetTopicAttributes",
              "events:PutTargets",
              "events:EnableRule",
              "events:PutRule"
          ],
          "Resource": "*"
      }
```

```
  ]
}
```

Para obter mais informações sobre como mover assinaturas para o EventBridge, consulte UpdateSubscriptionsToEventBridge na Referência da API do AWS Database Migration Service.

Como trabalhar com eventos e notificações do Amazon SNS

As versões 3.4.5 e anteriores do AWS DMS são compatíveis com o trabalho com eventos e notificações conforme descrito a seguir.

O AWS Database Migration Service (AWS DMS) pode utilizar o Amazon Simple Notification Service (Amazon SNS) para fornecer notificações quando ocorre um evento do AWS DMS, por exemplo, a criação ou a exclusão de uma instância de replicação. É possível trabalhar com essas notificações em qualquer formato compatível com o Amazon SNS em uma região da AWS, como uma mensagem de e-mail, uma mensagem de texto ou uma chamada para um endpoint HTTP.

O AWS DMS agrupa os eventos em categorias que você pode assinar para receber notificações quando ocorrer um evento dessa categoria. Por exemplo, se você assinar a categoria Criação de uma instância de replicação específica, será notificado sempre que ocorrer um evento relacionado à criação que afete a instância de replicação. Se fizer uma assinatura na categoria de alteração de configuração em uma instância de replicação, será notificado quando a configuração da instância de replicação for alterada. Você também recebe uma notificação quando uma assinatura de notificação de evento é alterada. Para ver uma lista das categorias de eventos fornecidos pelo AWS DMS, consulte Categorias de eventos do AWS DMS e mensagens de eventos para notificações do SNS.

O AWS DMS envia notificações de eventos para os endereços fornecidos quando a assinatura do evento é criada. Pode ser interessante para você criar várias assinaturas diferentes, como por exemplo uma assinatura para receber todas as notificações de eventos e outra que inclua somente eventos críticos para os seus recursos de produção do DMS. As notificações podem ser facilmente desativadas sem excluir uma assinatura ao cancelar a seleção da opção Ativada, no console do AWS DMS, ou definindo o parâmetro `Enabled` como falso utilizando a API do AWS DMS.

> **Note**
>
> As notificações de eventos do AWS DMS utilizando mensagens de texto SMS estão disponíveis no momento para recursos do AWS DMS em todas as regiões da AWS em que

o Amazon SNS é compatível. Para obter uma lista das regiões e países da AWS em que o Amazon SNS é compatível com mensagens SMS, consulte Regiões e países compatíveis. Para obter mais informações sobre como utilizar mensagens de texto com o SNS, consulte Envio e recebimento de notificações por SMS utilizando o Amazon SNS.
As notificações de eventos do AWS DMS diferem dos eventos do CloudTrail no CloudWatch ou no EventBridge. As notificações de eventos do CloudTrail podem ser geradas por qualquer invocação de API. O DMS envia uma notificação somente quando ocorre um evento do DMS.

O AWS DMS utiliza um identificador de assinatura para identificar cada assinatura. É possível ter várias assinaturas de eventos do AWS DMS publicadas no mesmo tópico do Amazon SNS. Taxas do Amazon SNS são aplicadas quando você utiliza a notificação de eventos. Para obter mais informações sobre o faturamento do Amazon SNS, consulte Preços do Amazon SNS.

Para assinar eventos do AWS DMS com o Amazon SNS utilize o seguinte processo:

1. Crie um tópico do Amazon SNS. Neste tópico, especifique o tipo de notificação que deseja receber e para qual endereço ou número a notificação deve ir.
2. Crie uma assinatura de notificação de evento do AWS DMS utilizando o AWS Management Console, a AWS CLI ou a API do AWS DMS.
3. O AWS DMS envia um e-mail ou uma mensagem SMS de aprovação para os endereços submetidos com a assinatura. Para confirmar a assinatura, clique no link no e-mail ou mensagem SMS de aprovação.
4. Quando a assinatura é confirmada, o status da assinatura é atualizado na seção Assinaturas de eventos no console do AWS DMS.
5. E você começará a receber notificações de eventos.

Para obter uma lista das categorias e dos eventos sobre os quais você pode ser notificado, consulte a próxima seção. Para obter mais detalhes sobre como assinar e trabalhar com assinaturas de eventos no AWS DMS, consulte Assinatura para notificação de eventos do AWS DMS utilizando o SNS.

Categorias de eventos do AWS DMS e mensagens de eventos para notificações do SNS

> **Important**
>
> A partir do lançamento do AWS DMS 3.4.5 e com versões posteriores, é recomendável utilizar o Amazon EventBridge para fornecer notificações quando ocorre um evento no AWS DMS. Para obter mais informações sobre como utilizar eventos do EventBridge com o AWS DMS, consulte Como trabalhar com eventos e notificações do Amazon EventBridge no AWS Database Migration Service.

O AWS DMS gera um número significativo de eventos em categorias que você pode assinar utilizando o console do AWS DMS ou a API do AWS DMS. Cada categoria se aplica a um tipo de origem. No momento, o AWS DMS é compatível com os tipos de origem de instância de replicação e de tarefa de replicação.

A tabela a seguir mostra as categorias e os eventos possíveis para o tipo de origem de instância de replicação.

Categoria	ID do evento no DMS	Descrição
Alteração na configuração	DMS-EVENT-0012	A classe dessa instância de replicação está sendo alterada.
Alteração na configuração	DMS-EVENT-0014	A classe dessa instância de replicação foi alterada.
Alteração na configuração	DMS-EVENT-0018	O armazenamento da instância de replicação está sendo aumentado.
Alteração na configuração	DMS-EVENT-0017	O armazenamento da instância de replicação foi aumentado.
Alteração na configuração	DMS-EVENT-0024	A instância de replicação está fazendo a transição para uma configuração multi-AZ.

Categoria	ID do evento no DMS	Descrição
Alteração na configuração	DMS-EVENT-0025	A instância de replicação concluiu a transição para uma configuração multi-AZ.
Alteração na configuração	DMS-EVENT-0030	A instância de replicação está fazendo a transição para uma configuração Single-AZ.
Alteração na configuração	DMS-EVENT-0029	A instância de replicação concluiu a transição para uma configuração Single-AZ.
Criação	DMS-EVENT-0067	A instância de replicação está sendo criada.
Criação	DMS-EVENT-0005	A instância de replicação está criada.
Exclusão	DMS-EVENT-0066	A instância de replicação está sendo excluída.
Exclusão	DMS-EVENT-0003	A instância de replicação está excluída.
Manutenção	DMS-EVENT-0047	O software de gerenciamento na instância de replicação foi atualizado.
Manutenção	DMS-EVENT-0026	Está ocorrendo a manutenção off-line da instância de replicação. A instância de replicação não está disponível no momento.
Manutenção	DMS-EVENT-0027	A manutenção off-line da instância de replicaçã o está concluída. A instância de replicação agora está disponível.
Manutenção	DMS-EVENT-0068	A instância de replicação está em um estado que não pode ser atualizado.
LowStorage	DMS-EVENT-0007	A instância de replicação consumiu mais de 90% do armazenamento alocado. É possível monitorar o espaço de armazenamento de uma instância de replicação utilizando a métrica Espaço de armazenamento gratuito.

Categoria	ID do evento no DMS	Descrição
Failover	DMS-EVENT-0013	Failover iniciado para uma instância de replicação multi-AZ.
Failover	DMS-EVENT-0049	O failover está concluído para uma instância de replicação multi-AZ.
Failover	DMS-EVENT-0015	Failover de multi-AZ para o modo de espera concluído.
Failover	DMS-EVENT-0050	A ativação multi-AZ foi iniciada.
Failover	DMS-EVENT-0051	A ativação multi-AZ foi concluída.
Failover	DMS-EVENT-0034	Se você solicitar failover com muita frequência, ocorrerá esse evento em vez dos eventos normais de failover.
Falha	DMS-EVENT-0031	A instância de replicação entrou em falha de armazenamento.
Falha	DMS-EVENT-0036	Falha na instância de replicação devido a uma rede incompatível.
Falha	DMS-EVENT-0037	O serviço não pode acessar a chave do AWS KMS usada para criptografar o volume de dados.

A tabela a seguir mostra as categorias e os eventos possíveis para o tipo de origem da tarefa de replicação.

Categoria	ID do evento no DMS	Descrição
Alteração de estado	DMS-EVENT-0069	A tarefa de replicação foi iniciada.
Alteração de estado	DMS-EVENT-0081	Foi solicitado o recarregamento dos detalhes da tabela.

Categoria	ID do evento no DMS	Descrição
Alteração de estado	DMS-EVENT-0079	A tarefa de replicação foi interrompida.
Alteração de estado	DMS-EVENT-0091	Leitura pausada, limite de arquivos de troca atingido.
Alteração de estado	DMS-EVENT-0092	Leitura pausada, limite de uso de disco atingido.
Alteração de estado	DMS-EVENT-0093	Leitura retomada.
Falha	DMS-EVENT-0078	Falha na tarefa de replicação.
Falha	DMS-EVENT-0082	Falha na limpeza dos dados da tarefa em uma chamada para excluir a tarefa.
Alteração na configuração	DMS-EVENT-0080	A tarefa de replicação está modificada.
Exclusão	DMS-EVENT-0073	A instância de replicação está excluída.
Criação	DMS-EVENT-0074	A instância de replicação está criada.

O exemplo a seguir mostra uma assinatura do AWS DMS evento com a categoria Alteração de estado.

```
        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
```

Assinatura para notificação de eventos do AWS DMS utilizando o SNS

> **Important**
>
> A partir do lançamento do AWS DMS 3.4.5 e com versões posteriores, é recomendável utilizar o Amazon EventBridge para fornecer notificações quando ocorre um evento no AWS DMS. Para obter mais informações sobre como utilizar eventos do EventBridge com o AWS DMS, consulte Como trabalhar com eventos e notificações do Amazon EventBridge no AWS Database Migration Service.

É possível criar uma assinatura de notificação de evento do AWS DMS a fim de ser notificado quando ocorrer um evento no AWS DMS. A forma mais fácil de criar uma assinatura é com o console do AWS DMS. Em uma assinatura de notificação, você escolhe como e para onde enviar notificações. Você especifica o tipo de origem sobre o qual deseja ser notificado. Atualmente, o AWS DMS é compatível com os tipos de origem de instância de replicação e de tarefa de replicação. Dependendo do tipo de origem selecionado, escolha as categorias e as origens de eventos e identifique a origem da qual deseja receber notificações de eventos.

Utilização do AWS Management Console

> **Important**
>
> A partir do lançamento do AWS DMS 3.4.5 e com versões posteriores, é recomendável utilizar o Amazon EventBridge para fornecer notificações quando ocorre um evento no AWS DMS. Para obter mais informações sobre como utilizar eventos do EventBridge com o AWS DMS, consulte Como trabalhar com eventos e notificações do Amazon EventBridge no AWS Database Migration Service.

Como assinar uma notificação de eventos do AWS DMS com o Amazon SNS utilizando o console

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.

Se estiver conectado como um usuário do IAM, verifique se você possui as permissões necessárias para acessar o AWS DMS.

2. No painel de navegação, escolha Assinaturas de eventos.
3. Na página Assinaturas de eventos, escolha Criar assinatura de evento.
4. Na página Criar assinatura de evento, faça o seguinte:

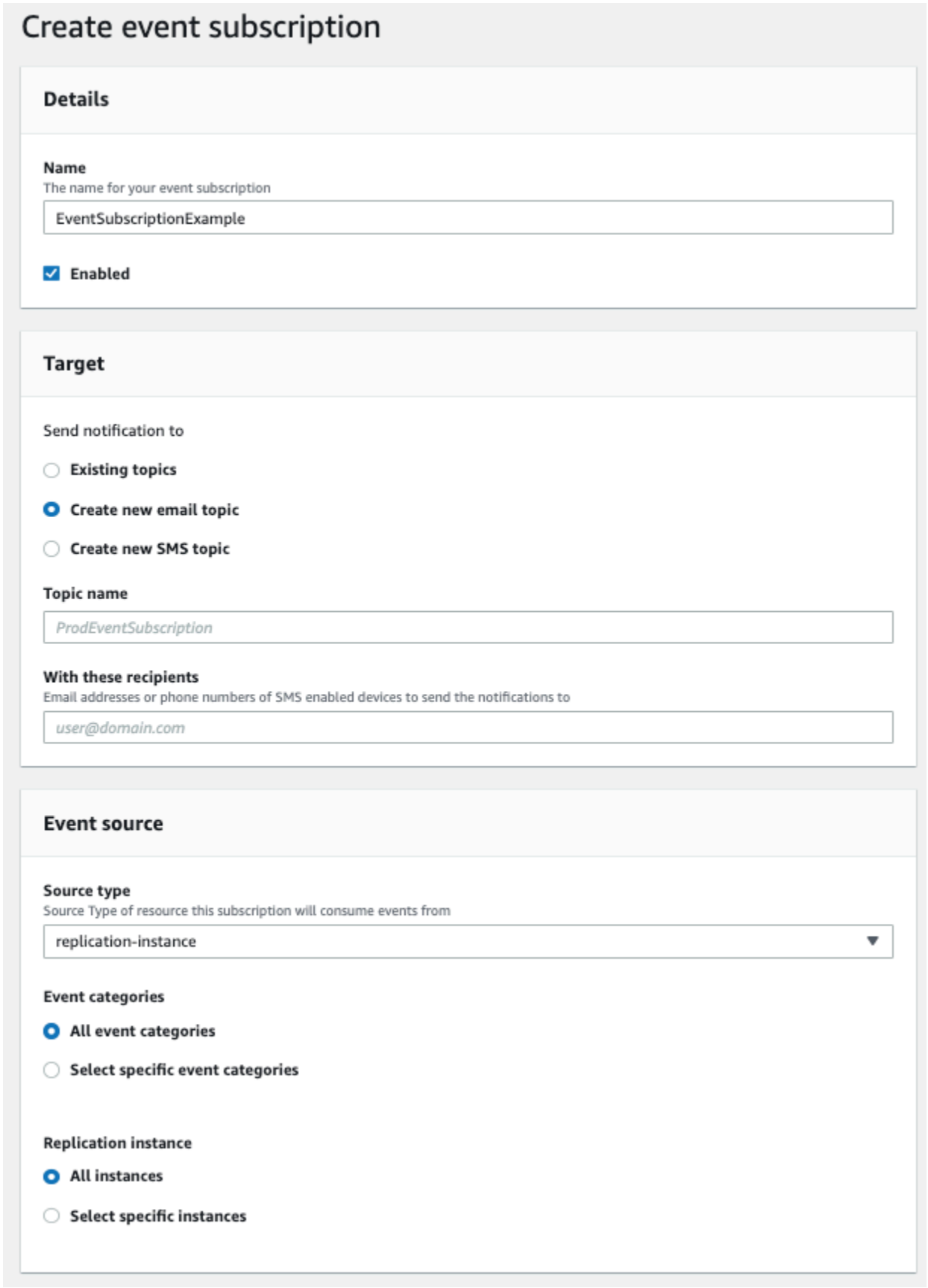
   a. Em Detalhes, em Nome, insira um nome para a assinatura de notificação de evento.

   b. Escolha Habilitada, para ativar a assinatura. Se quiser criar a assinatura, mas ainda não quiser receber notificações, não escolha Habilitada.

   c. Em Destino, escolha Tópicos existentes, Criar novo tópico de e-mail ou Criar novo tópico de SMS para enviar notificações. Verifique se já existe um tópico do Amazon SNS para o qual enviar notificações, ou crie o tópico. Se escolher um tópico, poderá inserir um endereço de e-mail para receber as notificações.

   d. Em Origem do evento, em Tipo de evento, escolha um tipo de origem. As únicas opções são replication-instance e replication-task.

   e. Dependendo do tipo de origem selecionado, escolha as categorias de eventos e as origens das quais deseja receber notificações.


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. Selecione Criar assinatura de evento.

O console do AWS DMS indica que a assinatura está sendo criada.

> **Note**
>
> Também é possível criar assinaturas de notificação de eventos do Amazon SNS utilizando a API do AWS DMS e a CLI. Para obter mais informações, consulte CreateEventSubscription na Referência da API do AWS DMS e create-event-subscription na documentação Referência da CLI do AWS DMS.

Validação da política de acesso do tópico do SNS

A política de acesso ao SNS exige permissões para que o AWS DMS publique eventos nó tópico do SNS. É possível validar e atualizar a política de acesso conforme descrito nos procedimentos a seguir.

Como validar a política de acesso

1. Abra o console do Amazon SNS.
2. No painel de navegação, escolha Tópicos e selecione o tópico sobre o qual você deseja receber notificações do DMS.
3. Selecione a guia Política de acesso.

Será possível atualizar a política se a política de acesso ao SNS não permitir que o AWS DMS publique eventos no tópico do SNS.

Como atualizar a política de acesso

1. Na seção Detalhes da página de tópicos, escolha Editar.
2. Expanda a seção Política de acesso e anexe a seguinte política ao editor JSON.

```
{
      "Sid": "dms-allow-publish",
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
```

```
      },
      "Action": "sns:Publish",
      "Resource": "your-SNS-topic-ARN"
    }
```

É recomendável restringir ainda mais o acesso ao tópico do SNS especificando a condição `aws:SourceArn`, que é o ARN de EventSubscription do DMS que publica eventos no tópico.

```
...
"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. Escolha Salvar alterações.

Validação de dados do AWS DMS

Tópicos

O AWS DMS é compatível com a validação de dados para garantir que os dados sejam migrados com precisão da origem para o destino. Se ativada, a validação começa imediatamente após a execução da carga máxima de uma tabela. A validação compara as alterações incrementais de uma tarefa ativada para a CDC à medida que ocorrem.

Durante a validação de dados, o AWS DMS compara cada linha na origem com a linha correspondente no destino e verifica se essas linhas contêm quaisquer incompatibilidades. Para fazer isso, o AWS DMS emite consultas apropriadas para recuperar os dados. Observe que essas consultas consomem recursos adicionais na origem e no destino, bem como em outros recursos de rede.

Para uma tarefa exclusiva de apenas CDC com a validação ativada, todos os dados preexistentes em uma tabela são validados antes de iniciar a validação de novos dados.

A validação de dados funciona com os seguintes bancos de dados de origem sempre que o AWS DMS for compatível com eles como endpoints de origem:

- Oracle
- Banco de dados compatível com o PostgreSQL (PostgreSQL, Aurora PostgreSQL ou Aurora Sem Servidor para PostgreSQL)
- Banco de dados compatível com o MySQL (MySQL, MariaDB, Aurora MySQL ou Aurora Sem Servidor para MySQL)

- Microsoft SQL Server
- IBM Db2 LUW

A validação de dados funciona com os seguintes bancos de dados de destino sempre que o AWS DMS for compatível com eles como endpoints de destino:

- Oracle
- Banco de dados compatível com o PostgreSQL (PostgreSQL, Aurora PostgreSQL ou Aurora Sem Servidor para PostgreSQL)
- Banco de dados compatível com o MySQL (MySQL, MariaDB, Aurora MySQL ou Aurora Sem Servidor para MySQL)
- Microsoft SQL Server
- IBM Db2 LUW
- Amazon Redshift
- Amazon S3. Para obter informações sobre como validar os dados de destino do Amazon S3, consulte Validação de dados de destino do Amazon S3.

Para obter mais informações sobre os endpoints com suporte, consulte Como trabalhar com endpoints do AWS DMS.

A validação de dados requer tempo adicional, além da quantidade necessária para a migração em si. O tempo extra necessário depende do volume de dados que foi migrado.

Para ter mais informações sobre essas configurações, consulte Configurações da tarefa de validação de dados.

Para obter um exemplo das configurações da tarefa `ValidationSettings` em um arquivo JSON, consulte Exemplo de configurações de tarefas.

Estatísticas da tarefa de replicação

Quando a validação de dados é ativada, o AWS DMS passa as seguintes estatísticas no nível da tabela:

- ValidationState: o estado de validação da tabela. O parâmetro pode ter os valores a seguir:

  - Não ativado: a validação não é ativada para a tabela na tarefa de migração.
  - Registros pendentes: alguns registros na tabela estão aguardando validação.
  - Registros incompatíveis: alguns registros na tabela não correspondem entre a origem e o destino. Uma incompatibilidade pode ocorrer por vários motivos. Para obter mais informações, consulte a tabela `awsdms_control.awsdms_validation_failures_v1` no endpoint de destino.
  - Registros suspensos: não foi possível validar alguns registros na tabela.
  - Nenhuma chave primária: não foi possível validar a tabela, pois ela não tinha uma chave primária.
  - Erro de tabela: não foi possível validar a tabela, porque ela estava em estado de erro, e alguns dados não foram migrados.
  - Validadas: todas as linhas na tabela estão validadas. Se a tabela for atualizada, o status poderá não ser mais Validado.
  - Erro: não é possível validar a tabela devido a um erro inesperado.
  - Validação pendente: a tabela está aguardando validação.
  - Preparando a tabela: preparando a tabela ativada na tarefa de migração para validação.
  - Revalidação pendente: todas as linhas na tabela estão pendentes de validação após a atualização da tabela.
- ValidationPending: o número de registros migrados que foram para o destino, mas que ainda não foram validados.
- ValidationSuspended: o número de registros que o AWS DMS não pode comparar. Por exemplo, se um registro na origem for atualizado constantemente, o AWS DMS não conseguirá comparar a origem e o destino.
- ValidationFailed: o número de registros que não foram aprovados na fase de validação de dados.

Para obter um exemplo das configurações da tarefa `ValidationSettings` em um arquivo JSON, consulte Exemplo de configurações de tarefas.

É possível visualizar as informações de validação de dados utilizando o console, a AWS CLI ou a API do AWS DMS.

- No console, é possível optar por validar uma tarefa ao criá-la ou modificá-la. Para visualizar o relatório de validação de dados utilizando o console, escolha a página Tarefas e escolha a guia Estatísticas da tabela, na seção de detalhes.

- Utilizando a CLI, defina o parâmetro `EnableValidation` como `true` ao criar ou modificar uma tarefa para começar a validação de dados. O exemplo a seguir cria uma tarefa e permite a validação de dados.

```
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"}]}'
```

Use o comando `describe-table-statistics` para receber o relatório de validação de dados no formato JSON. O comando a seguir mostra o relatório de validação de dados.

```
aws dms  describe-table-statistics --replication-task-arn arn:aws:dms:us-
east-1:5731014:
rep:36KWVMB7Q
```

O relatório seria semelhante ao seguinte.

```
{
    "ReplicationTaskArn": "arn:aws:dms:us-west-2:5731014:task:VFPFTYKK2RYSI",
    "TableStatistics": [
        {
            "ValidationPendingRecords": 2,
            "Inserts": 25,
            "ValidationState": "Pending records",
            "ValidationSuspendedRecords": 0,
            "LastUpdateTime": 1510181065.349,
            "FullLoadErrorRows": 0,
            "FullLoadCondtnlChkFailedRows": 0,
            "Ddls": 0,
            "TableName": "t_binary",
            "ValidationFailedRecords": 0,
```

```
            "Updates": 0,
            "FullLoadRows": 10,
            "TableState": "Table completed",
            "SchemaName": "d_types_s_sqlserver",
            "Deletes": 0
        }
}
```

- Utilizando a API do AWS DMS, crie uma tarefa que utiliza a ação CreateReplicationTask e defina o parâmetro `EnableValidation` como verdadeiro para validar os dados migrados pela tarefa. Use a ação DescribeTableStatistics para receber o relatório de validação de dados no formato JSON.

Estatísticas de tarefas de replicação com o Amazon CloudWatch

Quando o Amazon CloudWatch está ativado, o AWS DMS fornece as seguintes estatísticas de tarefas de replicação:

- ValidationSucceededRecordCount: número de linhas validadas pelo AWS DMS por minuto.
- ValidationAttemptedRecordCount: número de linhas com tentativa de validação por minuto.
- ValidationFailedOverallCount: número de linhas onde a validação falhou.
- ValidationSuspendedOverallCount: número de linhas onde a validação foi suspensa.
- ValidationPendingOverallCount: número de linhas onde a validação ainda está pendente.
- ValidationBulkQuerySourceLatency: o AWS DMS pode executar a validação de dados em massa, especialmente em certos cenários durante uma replicação de carga máxima ou contínua quando houver muitas alterações. Essa métrica indica a latência necessária para ler um conjunto de dados em massa no endpoint de origem.
- ValidationBulkQueryTargetLatency: o AWS DMS pode realizar a validação de dados em massa, especialmente em certos cenários durante uma replicação de carga máxima ou contínua quando houver muitas alterações. Essa métrica indica a latência necessária para ler um conjunto de dados em massa no endpoint de destino.
- ValidationItemQuerySourceLatency: durante a replicação contínua, a validação de dados pode identificar alterações contínuas e validar essas alterações. Essa métrica indica a latência de leitura das alterações a partir da origem. A validação pode executar mais consultas do que o necessário, com base no número de alterações, se houver erros durante a validação.
- ValidationItemQueryTargetLatency: durante a replicação contínua, a validação de dados pode identificar alterações contínuas e validar as alterações linha por linha. Essa métrica fornece a

latência de leitura das alterações a partir do destino. A validação pode executar mais consultas do que o necessário, com base no número de alterações, se houver erros durante a validação.

Para coletar informações sobre a validação de dados nas estatísticas ativadas pelo CloudWatch, selecione Ativar logs do CloudWatch ao criar ou modificar uma tarefa utilizando o console. Para visualizar as informações sobre a validação de dados e garantir que os dados foram migrados de forma precisa da origem para o destino, faça o seguinte.

1. Escolha a tarefa pai na lista na página Tarefas de migração de banco de dados.
2. Escolha a guia Métricas do CloudWatch.
3. Selecione Validação no menu suspenso.

Revalidar tabelas durante uma tarefa

Enquanto uma tarefa estiver em execução, é possível solicitar que o AWS DMS execute a validação de dados.

AWS Management Console

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.

   Se estiver conectado como usuário do AWS Identity and Access Management (IAM), verifique se você tem as permissões necessárias para acessar o AWS DMS. Para obter as permissões necessárias, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, selecione Tasks.
3. Escolha a tarefa em execução cuja tabela você deseja revalidar.
4. Escolha a guia Table Statistics (Estatísticas da tabela).
5. Escolha a tabela que deseja revalidar (é possível escolher até 10 tabelas por vez). Se a tarefa não estiver mais em execução, não será possível revalidar a(s) tabela(s).
6. Selecione Revalidate (Revalidar).

Utilizar o editor JSON para modificar regras de validação

Para adicionar uma regra de validação a uma tarefa utilizando o editor JSON no console do AWS DMS, faça o seguinte:

1. Selecione Tarefas de migração de banco de dados.
2. Selecione a tarefa na lista de tarefas de migração.
3. Se a tarefa estiver em execução, selecione Interromper no menu suspenso Ações.
4. Depois que a tarefa for interrompida, para modificá-la, selecione Modificar no menu suspenso Ações.
5. Na seção Mapeamentos de tabela, selecione o editor JSON e adicione sua regra de validação aos mapeamentos de tabela.

Por exemplo, é possível adicionar a regra de validação a seguir para executar um perfil de substituição na origem. Nesse caso, se a regra de validação encontrar um byte nulo, ela o validará como um espaço.

```
{
 "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}"
}
```

Tarefas somente de validação

É possível criar tarefas somente de validação para visualizar e validar os dados sem executar nenhuma migração ou replicação de dados. Para criar uma tarefa somente de validação, defina as

configurações `EnableValidation` e `ValidationOnly` como `true`. Ao ativar `ValidationOnly`, requisitos adicionais se aplicam. Para obter mais informações, consulte Configurações da tarefa de validação de dados.

Para um tipo de migração de carga máxima somente, uma tarefa somente de validação é concluída muito mais rapidamente do que a CDC equivalente quando muitas falhas são relatadas. Mas as alterações no endpoint de origem ou de destino são relatadas como falhas no modo de carga máxima, uma possível desvantagem.

Uma tarefa somente de validação da CDC atrasa a validação com base na latência média e repete as falhas várias vezes antes de relatá-las. Se a maioria das comparações de dados resultar em falhas, uma tarefa somente de validação do modo CDC será muito lenta, uma desvantagem potencial.

Uma tarefa somente de validação deve ser configurada na mesma direção da tarefa de replicação, especialmente para a CDC. Isso ocorre porque uma tarefa somente de validação de CDC detecta quais linhas foram alteradas e precisam ser revalidadas com base no log de alterações na origem. Se o destino for especificado como a origem, ele só saberá sobre as alterações enviadas ao destino pelo DMS e não terá a garantia de detectar erros de replicação.

Validação somente de carga máxima

A partir do AWS DMS versão 3.4.6 e superior, uma tarefa somente de validação de carga máxima compara rapidamente todas as linhas das tabelas de origem e de destino em uma única passagem, relata imediatamente quaisquer falhas e é encerrada. A validação nunca é suspensa devido a falhas nesse modo, ela é otimizada para velocidade. Mas as alterações no endpoint de origem ou de destino são relatadas como falhas.

> **Note**
>
> A partir do AWS DMS versão 3.4.6 e superior, esse comportamento da validação também se aplica à tarefa de migração de carga máxima com a validação ativada.

Validação somente de CDC

Uma tarefa somente de validação de CDC valida todas as linhas existentes entre as tabelas de origem e de destino em um novo início. Além disso, uma tarefa somente de validação de CDC é

executada continuamente, revalida as alterações de replicação em andamento, limita o número de falhas relatadas em cada passagem e repete as linhas incompatíveis antes de declará-las como falhas. Ela é otimizada para evitar falsos positivos.

A validação de uma tabela (ou de toda a tarefa) será suspensa se os limites de `FailureMaxCount` ou de `TableFailureMaxCount` forem violados. Isso também se aplica a uma tarefa de migração de CDC ou de carga máxima+CDC com a validação ativada. E uma tarefa de CDC com validação ativada atrasa a revalidação de cada linha alterada com base na latência média de origem e de destino.

Mas tarefa de somente validação de CDC não migra dados e não tem latência. Ela define `ValidationQueryCdcDelaySeconds` como 180 por padrão. É possível aumentar a quantidade para considerar ambientes de alta latência e ajudar a evitar falsos positivos.

Casos de uso de somente de validação

Os casos de uso para a divisão da parte de validação de dados de uma tarefa de migração ou replicação em uma tarefa somente de validação separada incluem, mas não estão limitados, ao seguinte:

- Controle exatamente quando a validação ocorre: as consultas de validação adicionam uma carga extra aos endpoints de origem e de destino. Portanto, migrar ou replicar dados em uma primeira tarefa e validar os resultados em outra tarefa pode ser benéfico.
- Reduza a carga na instância de replicação: a divisão da validação de dados para execução em sua própria instância pode ser vantajosa.
- Obtenha rapidamente quantas linhas não correspondem em um determinado momento: por exemplo, imediatamente antes ou durante uma transição para a produção da janela de manutenção para um endpoint de destino, é possível criar uma tarefa de somente validação de carga máxima para obter uma resposta à sua pergunta.
- Quando falhas de validação são esperadas para uma tarefa de migração com um componente da CDC: por exemplo, ao migrar o Oracle para o `varchar2` PostgreSQL`jsonb`, a validação da CDC continua repetindo essas linhas com falha e limita o número de falhas relatadas a cada vez. Porém, é possível criar uma tarefa somente de validação de carga máxima e obter uma resposta mais rápida.
- Você desenvolveu um script/utilitário de reparo de dados que lê a tabela com falha de validação: (consulte também Solução de problemas). Uma tarefa somente de validação de carga máxima relata rapidamente as falhas nas quais o script de reparo de dados deve atuar.

Para obter um exemplo das configurações da tarefa `ValidationSettings` em um arquivo JSON, consulte Exemplo de configurações de tarefas).

Solução de problemas

Durante a validação, o AWS DMS cria uma nova tabela no endpoint de destino: `awsdms_control.awsdms_validation_failures_v1`. Se qualquer registro entrar no estado ValidationSuspended ou ValidationFailed, o AWS DMS gravará as informações de diagnóstico em `awsdms_control.awsdms_validation_failures_v1`. É possível consultar essa tabela para ajudar a solucionar erros de validação.

Para obter informações sobre como alterar o esquema padrão em que a tabela é criada no destino, consulte Configurações da tarefa da tabela de controle.

Veja a seguir uma descrição da tabela `awsdms_control.awsdms_validation_failures_v1`:

Nome da coluna	Tipo de dados	Descrição
TASK_NAME	VARCHAR(128) NOT NULL	Identificador de tarefa do AWS DMS.
TABLE_OWNER	VARCHAR(128) NOT NULL	Schema (proprietário) da tabela.
TABLE_NAME	VARCHAR(128) NOT NULL	Nome da tabela.
FAILURE_TIME	DATETIME(3) NOT NULL	Hora em que a falha ocorreu.
KEY_TYPE	VARCHAR(128) NOT NULL	Reservado para uso futuro (o valor é sempre 'Linha')
KEY	TEXT NOT NULL	Esta é a chave primária para o tipo de registro de linha.
FAILURE_TYPE	VARCHAR(128) NOT NULL	Severidade do erro de validação. Pode ser `RECORD_DIFF` , `MISSING_SOURCE` ou `MISSING_TARGET` .

Nome da coluna	Tipo de dados	Descrição
DETAILS	VARCHAR(8000) NOT NULL	String formatada em JSON de todos os valores de colunas de origem/destino que não correspondem à chave fornecida.

A consulta a seguir mostra todas as falhas de uma tarefa pela tabela `awsdms_control.awsdms_validation_failures_v1`. O nome da tarefa deve ser seu ID de recurso externo. O ID de recurso externo da tarefa é o último valor em seu ARN. Por exemplo, para uma tarefa com um valor de ARN arn:aws:dms:us-west-2:5599:task: VFPFKH4FJR3FTYKK2RYSI, o ID de recurso externo será 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'}],]
```

É possível examinar o campo `DETAILS` para determinar quais colunas não correspondem. Como você tem a chave primária do registro que falhou, poderá consultar os endpoints de origem e de destino para ver qual parte do registro não corresponde.

Desempenho da validação do Redshift

O Amazon Redshift difere dos bancos de dados relacionais em vários sentidos, como armazenamento colunar, MPP, compactação de dados e outros fatores. Essas diferenças conferem ao Redshift um perfil de desempenho distinto em relação aos bancos de dados relacionais.

Durante a fase de replicação de carga máxima, a validação usa consultas de intervalo e o tamanho dos dados é controlado pela configuração `PartitionSize`. Essas consultas baseadas em intervalos selecionam todos os registros da tabela de origem.

Na replicação contínua, as consultas alternam entre buscas de registros individuais e baseadas em intervalos. O tipo de consulta é determinado dinamicamente com base em vários fatores, como os seguintes:

- Volume de consultas
- Tipos de consulta DML na tabela de origem
- Latência de tarefa
- Número total de registros
- Configurações de validação, como `PartitionSize`

Você pode ver uma carga adicional no cluster do Amazon Redshift devido a consultas de validação. Como os fatores acima variam entre os casos de uso, você deve analisar o desempenho da consulta de validação e ajustar o cluster e a tabela adequadamente. Algumas opções para mitigar problemas de desempenho incluem o seguinte:

- Reduza as configurações `ThreadCount` e `PartitionSize` para ajudar a diminuir a workload durante a validação da carga máxima. Observe que isso retardará a validação de dados.
- Embora o Redshift não imponha chaves primárias, o AWS DMS depende delas para identificar os registros de maneira exclusiva no destino para validação de dados. Se possível, defina a chave primária para espelhar a chave de classificação de forma que as consultas de validação de carga máxima sejam executadas mais rapidamente.

Limitações

- A validação de dados requer que a tabela tenha uma chave primária ou índice exclusivo.
  - As colunas de chave primária não podem ser do tipo `CLOB`, `BLOB` ou `BYTE`.
  - Para colunas de chave primária do tipo `VARCHAR` ou `CHAR`, o comprimento deve ser inferior a 1024. Você deve especificar o tamanho no tipo de dados. Não é possível usar tipos de dados ilimitados como chave primária para validação de dados.
  - Uma chave do Oracle criada com a cláusula `NOVALIDATE` não é considerada uma chave primária ou um índice exclusivo.

  - Para uma tabela do Oracle sem uma chave primária e somente com uma chave exclusiva, as colunas com a restrição exclusiva também devem ter uma restrição `NOT NULL`.
- A validação de valores NULL PK/UK não é compatível.
- Se o agrupamento da coluna de chave primária na instância PostgreSQL de destino não for definido como "C", a ordem de classificação da chave primária será diferente da ordem de classificação em Oracle. Se a ordem de classificação for diferente entre PostgreSQL e Oracle, a validação de dados não poderá validar os registros.
- A validação de dados gera consultas adicionais em relação a bancos de dados de origem e destino. Os dois bancos de dados devem ter recursos suficientes para lidar com essa carga adicional. Isso se aplica principalmente aos destinos do Redshift. Para obter mais informações, consulte Desempenho da validação do Redshift a seguir.
- A validação de dados não é compatível ao consolidar vários bancos de dados em um banco de dados.
- Para um endpoint Oracle de origem ou destino, o AWS DMS utiliza DBMS_CRYPTO para validar LOBs. Se o endpoint da Oracle usar LOBs, você deverá conceder a permissão de execução em dbms_crypto para a conta de usuário usada para acessar esse endpoint. Para isso, execute a seguinte instrução:

```
grant execute on sys.dbms_crypto to dms_endpoint_user;
```

- Se o banco de dados de destino for modificado fora do AWS DMS durante a validação, as discrepâncias poderão não ser relatadas com precisão. Esse resultado pode ocorrer caso um de seus aplicativos grave dados na tabela de destino enquanto o AWS DMS está executando a validação na mesma tabela.
- Se uma ou mais linhas forem modificadas continuamente durante a validação, o AWS DMS não poderá validar essas linhas.
- Se o AWS DMS detectar mais de 10.000 registros com falha ou suspensos, ele interromperá a validação. Antes de continuar, solucione os problemas subjacentes com os dados.
- O AWS DMS não é compatível com a validação de dados de visualizações.
- O AWS DMS não é compatível com a validação de dados quando as configurações da tarefa de substituição de caracteres são utilizadas.
- O AWS DMS não é compatível com a validação do tipo Oracle LONG.
- O AWS DMS não é compatível com a validação do tipo Oracle Spatial durante a migração heterogênea.

Para obter as limitações ao utilizar a validação de destino do S3, consulte Limitações da utilização da validação de destino do S3..

Validação de dados de destino do Amazon S3

O AWS DMS é compatível com a validação de dados replicados em destinos do Amazon S3. Como o AWS DMS armazena dados replicados como arquivos sem formatação no Amazon S3, usamos consultas `CREATE TABLE AS SELECT` (CTAS) do Amazon Athena para validar os dados.

As consultas de dados armazenados no Amazon S3 exigem uso intensivo de comutação. Portanto, o AWS DMS executa a validação nos dados do Amazon S3 durante a captura de dados de alteração (CDC) somente uma vez por dia, à meia-noite (0h) UTC. Cada validação diária que o AWS DMS executa é chamada de validação de intervalo. Durante uma validação de intervalo, o AWS DMS valida todos os registros de alteração que foram migrados para o bucket de destino do Amazon S3 nas últimas 24 horas. Para obter mais informações sobre as limitações de validação de intervalo, consulte Limitações da utilização da validação de destino do S3..

A validação de destino do Amazon S3 utiliza o Amazon Athena, portanto, custos adicionais são aplicados. Para obter mais informações, consulte Preços do Amazon Athena.

> **Note**
>
> A validação de destino do S3 requer o AWS DMS versão 3.5.0 ou posterior.

Tópicos

- Pré-requisitos da validação do S3 de destino
- Permissões para utilizar a validação de destino do S3
- Limitações da utilização da validação de destino do S3.
- Utilizar tarefas de somente validação com a validação do S3 de destino

Pré-requisitos da validação do S3 de destino

Antes de utilizar a validação de destino do S3, verifique as seguintes configurações e permissões:

- Defina o valor de `DataFormat` das S3Settings do endpoint como `parquet`. Para ter mais informações, consulte Configurações de parquet para S3.

- Verifique se o perfil atribuído à conta do usuário utilizada para criar a tarefa de migração tem o conjunto de permissões correto. Consulte Permissões a seguir.

Para tarefas que utilizam a replicação contínua (CDC), verifique as seguintes configurações:

- Ative o registro em log suplementar para ter registros completos nos dados da CDC. Para obter informações sobre como ativar o registro em log complementar, consulte Adição automática de registro em log complementar a um endpoint de origem Oracle na seção Compatibilidade com a solução de problemas e diagnóstico deste guia.
- Defina o parâmetro `TimestampColumnName` para o endpoint de destino. Não há limitações no nome da coluna de timestamp. Para obter mais informações, consulte S3Settings.
- Configure o particionamento de pastas baseadas em data do destino. Para ter mais informações, consulte Utilizar o particionamento de pastas com base em data.

Permissões para utilizar a validação de destino do S3

Para configurar o acesso para utilizar a validação do destino do S3, verifique se o perfil atribuído à conta de usuário utilizada para criar a tarefa de migração tem o seguinte conjunto de permissões. Substitua os valores de amostra por seus próprios valores.

```
{
    "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>/*"
            ]
        }
    ]
}
```

Limitações da utilização da validação de destino do S3.

Visualize as seguintes limitações adicionais que se aplicam à utilização da validação de destino do S3. Para obter as limitações que se aplicam a todas as validações, consulte Limitações.

- O valor de `DatePartitionSequence` precisa de um componente Day. A validação de destino do S3 não é compatível com o formato `YYYYMM`.

- Quando a validação do intervalo está em execução durante a CDC, é possível ver erros de validação falsos na tabela `awsdms_validation_failures_v1`. Esses erros ocorrem porque o AWS DMS migra as alterações que chegaram durante a validação do intervalo para a pasta de partições do dia seguinte. Normalmente, essas alterações são gravadas na pasta de partições do dia atual. Esses erros falsos são uma limitação da validação da replicação de um banco de dados de origem dinâmico para um destino estático, como o Amazon S3. Para investigar esses erros falsos, verifique os registros perto do final da janela de validação (0h UTC), que é quando esses erros normalmente aparecem.

  Para minimizar o número de erros falsos, verifique se o valor de `CDCLatencySource` da tarefa é baixo. Para obter informações sobre como monitorar a latência, consulte Métricas de tarefas de replicação.
- As tarefas no estado `failed` ou `stopped` não validam as alterações do dia anterior. Para minimizar os erros de validação causados por falhas inesperadas, crie tarefas separadas somente de validação com os mesmos mapeamentos de tabela e endpoints de origem e de destino. Para obter mais informações sobre as tarefas de somente validação, consulte Utilizar tarefas de somente validação com a validação do S3 de destino.
- A coluna Status de validação nas estatísticas da tabela reflete o estado da validação do intervalo mais recente. Como resultado, uma tabela com incompatibilidades pode ser mostrada como validada após a validação do intervalo do dia seguinte. Verifique `s3_validation_failures folder` no bucket do Amazon S3 de destino para obter as incompatibilidades que ocorreram há mais de um dia.
- A validação do S3 usa o recurso de tabela agrupada do Amazon Athena. Isso permite que a validação do S3 faça uma cópia agrupada dos dados da tabela de destino. Isso significa que a cópia dos dados da tabela é dividida em subconjuntos que correspondem ao particionamento interno da validação do DMS. As mesas com compartimentos Athena têm um limite de 100.000 baldes. Todas as tabelas que a validação do S3 tentar validar que excedam esse limite falharão na validação. O número de buckets que o S3 Validation tenta criar é igual ao seguinte:

  ```
  (#records in the table) / (validation partition size setting)
  ```

  Para contornar essa limitação, aumente a configuração do tamanho da partição de validação para que o número de buckets criados pelo S3 Validation seja inferior a 100.000. Para obter mais informações sobre o armazenamento em intervalos, consulte Particionamento e armazenamento em intervalos no Athena no Guia do usuário do Amazon Athena.

Utilizar tarefas de somente validação com a validação do S3 de destino

Uma tarefa somente de validação executa a validação nos dados que devem ser migrados sem executar a migração.

As tarefas somente de validação continuam sendo executadas, mesmo que a tarefa de migração seja interrompida, o que garante que o AWS DMS não perca a janela de validação do intervalo de 0h UTC.

A utilização de tarefas de somente validação com os endpoints de destino do Amazon S3 apresenta as seguintes limitações:

- A validação de tarefas de carga máxima do Amazon S3 com a configuração Somente validação ativada é compatível, mas opera de forma diferente das tarefas de carga máxima e de somente validação de outros endpoints. Para o S3 como destino, uma tarefa desse tipo valida somente com base nos dados de carga máxima no destino do S3 e não validará em relação a nenhum dado migrado como parte de uma migração de CDC. Utilize esse recurso somente para validar os dados criados por uma tarefa somente de carga máxima. A utilização desse modo para validar os dados em um destino que tenha uma tarefa de CDC ativa em execução não produzirá uma validação eficaz.
- As tarefas somente de validação validam somente as alterações desde a última janela de validação do intervalo (0h UTC). As tarefas de somente validação não validam dados de carga máxima ou de dados de CDC de dias anteriores.

Marcar recursos no AWS Database Migration Service

É possível utilizar tags no AWS Database Migration Service (AWS DMS) para adicionar metadados aos recursos. Além disso, é possível utilizar essas tags com políticas do AWS Identity and Access Management (IAM) para gerenciar o acesso aos recursos do AWS DMS e controlar as ações que podem ser aplicadas aos recursos do AWS DMS. Por fim, utilize essas tags para monitorar os custos agrupando as despesas de recursos marcados com tags semelhantes.

Todos os recursos do AWS DMS podem ser marcados com tags:

- Certificados
- Provedores de dados
- Migrações de dados
- Endpoints
- Assinaturas de eventos
- Perfis de instância
- Projetos de migração
- Instâncias de replicação
- Grupos de sub-rede de replicação
- Tarefas de replicação

Uma tag do AWS DMS é um par de nome-valor que você define e associa a um recurso do AWS DMS. O nome é referido como a chave. Fornecer um valor para a chave é opcional. É possível utilizar tags para atribuir informações arbitrárias a um domínio do AWS DMS. Uma chave de tags pode ser usada, por exemplo, para definir uma categoria, e o valor da tag pode ser um item nessa categoria. Por exemplo, é possível definir uma chave de tag de "projeto" e um valor de tag de "Salix", indicando que o recurso do AWS DMS é atribuído ao projeto Salix. As tags também podem ser utilizadas para designar recursos do AWS DMS como sendo utilizados para testes ou produção, utilizando uma tag como environment=teste ou environment=produção. É recomendável utilizar um conjunto consistente de chaves de tags para facilitar o monitoramento de metadados associados aos recursos do AWS DMS.

Também é possível utilizar tags para organizar sua fatura da AWS para refletir sua própria estrutura de custo. Para fazer isso, inscreva-se para obter a fatura da sua Conta da AWS com os valores de chave de tag incluídos. Então, para ver o custo de recursos combinados, organize suas informações

de faturamento de acordo com recursos com os mesmos valores de chave de tags. Por exemplo, é possível marcar vários recursos com um nome de aplicação específico, e depois organizar suas informações de faturamento para ver o custo total daquela aplicação em vários serviços. Para obter mais informações, consulte Usar tags de alocação de custos no Guia do usuário do AWS Billing.

Cada recurso do AWS DMS tem um conjunto de tags que contém todas as tags atribuídas àquele recurso do AWS DMS. Um conjunto de tags pode conter até dez tags ou estar vazio. Se você adicionar uma tag a um recurso do AWS DMS que tenha a mesma chave que uma tag existente no recurso, o novo valor substituirá o antigo.

A AWS não aplica nenhum significado semântico às tags. Elas são interpretadas estritamente como cadeias de caracteres da AWS. O DMS pode definir tags em recurso do AWS DMS, dependendo das configurações utilizadas ao criar o recurso.

A lista a seguir descreve as características de uma tag do AWS DMS.

- A chave de tags é o nome obrigatório da tag. O valor da string pode ter de 1 a 128 caracteres Unicode e não pode ter os prefixos "aws:" ou "dms:". A string pode conter apenas o conjunto de letras Unicode, dígitos, espaço em branco, “_”, “.”, “/”, “=”, “+”, “-” (Java regex: `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`).

- O valor da tag é um valor de string opcional da tag. O valor da string pode ter de 1 a 256 caracteres Unicode e não pode ter os prefixos "aws:" ou "dms:". A string pode conter apenas o conjunto de letras Unicode, dígitos, espaço em branco, “_”, “.”, “/”, “=”, “+”, “-” (Java regex: `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`).

  Os valores não têm que ser exclusivos em um conjunto de tags e podem ser nulos. Por exemplo, é possível ter um par de chave-valor em um conjunto de tags definido como projeto/Trinity e centro-custos/Trinity.

É possível utilizar a AWS CLI ou a API do AWS DMS para adicionar, listar e excluir tags dos recursos do AWS DMS. Ao utilizar a AWS CLI ou a API do AWS DMS, forneça o nome de recurso da Amazon (ARN) para o recurso do AWS DMS com o qual deseja trabalhar. Para obter mais informações sobre a criação de um ARN, consulte Construindo um nome de recurso da Amazon (ARN) para AWS DMS.

Observe que as tags ficam armazenadas em cache para fins de autorização. Por isso, as adições e atualizações em tags de recursos do AWS DMS podem demorar vários minutos para serem disponibilizadas.

API

É possível adicionar, listar ou remover tags de um recurso do AWS DMS utilizando a API do AWS DMS.

- Para adicionar uma tag a um recurso do AWS DMS, utilize a operação `AddTagsToResource`.
- Para listar as tags atribuídas a um recurso do AWS DMS, utilize a operação `ListTagsForResource`.
- Para remover tags de um recurso do AWS DMS, utilize a operação `RemoveTagsFromResource`.

Para saber mais sobre como criar o ARN necessário, consulte Construindo um nome de recurso da Amazon (ARN) para AWS DMS.

Ao trabalhar com o XML utilizando a API do AWS DML, as tags utilizam o seguinte esquema:

```
<Tagging>
  <TagSet>
   <Tag>
    <Key>Project</Key>
    <Value>Trinity</Value>
   </Tag>
   <Tag>
    <Key>User</Key>
    <Value>Jones</Value>
   </Tag>
  </TagSet>
</Tagging>
```

A tabela a seguir fornece uma lista das tags XML permitidas e suas características. Observe que os valores de Chave e Valor diferenciam maiúsculas e minúsculas. Por exemplo, projeto=Trinity e PROJETO=Trinity são duas tags distintas.

Elemento de marcação por tag	Descrição
TagSet	Um conjunto de tags é um contêiner de todas as tags atribuídas a um recurso do Amazon RDS. Só pode haver um conjunto de tags por

Elemento de marcação por tag	Descrição
	recurso. Você trabalha com um TagSet somente por meio da API do AWS DMS.
Tag	Uma tag é um par de chave-valor definido pelo usuário. Pode haver de 1 a 10 tags em um conjunto de tags.
Chave	Uma chave é o nome obrigatório da tag. O valor da string pode ter de 1 a 128 caracteres Unicode e não pode ter os prefixos "dms:" or "aws:". A string pode conter apenas o conjunto de letras Unicode, dígitos, espaço em branco, “_”, “.”, “/”, “=”, “+”, “-” (Java regex: `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`). As chaves devem ser exclusivas a um conjunto de tags. Por exemplo, não pode haver um par de chaves em um conjunto de tags com a mesma chave com valores diferentes, como projeto/Trinity e projeto/X anadu.
Valor	Um valor é o valor opcional da tag. O valor da string pode ter de 1 a 256 caracteres Unicode e não pode ter os prefixos "dms:" or "aws:". A string pode conter apenas o conjunto de letras Unicode, dígitos, espaço em branco, “_”, “.”, “/”, “=”, “+”, “-” (Java regex: `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`). Os valores não têm que ser exclusivos em um conjunto de tags e podem ser nulos. Por exemplo, é possível ter um par de chave-valor em um conjunto de tags definido como projeto/Trinity e centro-custos/Trinity.

Segurança em AWS Database Migration Service

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de um data center e de uma arquitetura de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem.

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos compliance programs AWS. Para saber mais sobre os programas de conformidade que se aplicam AWS DMS, consulte AWS serviços no escopo por programa de conformidade.
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, inclusive a confidencialidade dos dados, os requisitos da organização, as leis e as regulamentações vigentes.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar AWS DMS. Os tópicos a seguir mostram como configurar para atender AWS DMS aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus AWS DMS recursos.

Você pode gerenciar o acesso aos seus AWS DMS recursos e bancos de dados (DBs). O método usado para gerenciar o acesso depende da tarefa de replicação que você precisa executar com AWS DMS:

- Use políticas AWS Identity and Access Management (IAM) para atribuir permissões que determinam quem tem permissão para gerenciar AWS DMS recursos. AWS DMS exige que você tenha as permissões apropriadas se fizer login como usuário do IAM. Por exemplo, é possível utilizar o IAM para determinar quem tem permissão para criar, descrever, modificar e excluir instâncias e clusters de banco de dados, marcar recursos ou modificar grupos de segurança. Para obter mais informações sobre o IAM e como usá-lo com AWS DMS, consulteGerenciamento de identidade e acesso para AWS Database Migration Service.
- AWS DMS usa Secure Sockets Layer (SSL) para suas conexões de endpoint com Transport Layer Security (TLS). Para obter mais informações sobre como usar SSL/TLS com AWS DMS, consulte. Usando SSL com AWS Database Migration Service

- AWS DMS usa chaves de criptografia AWS Key Management Service (AWS KMS) para criptografar o armazenamento usado pela sua instância de replicação e suas informações de conexão de endpoint. AWS DMS também usa chaves de AWS KMS criptografia para proteger seus dados de destino em repouso para endpoints de destino do Amazon S3 e do Amazon Redshift. Para ter mais informações, consulte Configurando uma chave de criptografia e especificando permissões AWS KMS.
- AWS DMS sempre cria sua instância de replicação em uma nuvem privada virtual (VPC) com base no serviço Amazon VPC para o maior controle de acesso à rede possível. Para as instâncias de banco de dados e clusters de instância, utilize a mesma VPC que a instância de replicação ou VPCs adicionais para corresponder a esse nível de controle de acesso. Cada Amazon VPC utilizada deve estar associada a um grupo de segurança que tenha regras que permitem que todo o tráfego saia em todas as portas da VPC. Essa abordagem permite a comunicação da instância de replicação com os endpoints dos bancos de dados de origem e de destino, desde que a entrada correta esteja ativada nesses endpoints.

  Para obter mais informações sobre as configurações de rede disponíveis para AWS DMS, consulteConfigurar uma rede para uma instância de replicação. Para obter mais informações sobre como criar uma instância de banco de dados ou um cluster de instância em uma VPC, consulte a documentação de gerenciamento de cluster e segurança nos bancos de dados da Amazon na Documentação da AWS. Para obter mais informações sobre configurações de rede compatíveis com o AWS DMS , consulte Configurar uma rede para uma instância de replicação.
- Para visualizar os registros de migração do banco de dados, você precisa das permissões apropriadas do Amazon CloudWatch Logs para a função do IAM que você está usando. Para obter mais informações sobre registro em log do AWS DMS, consulte Monitoramento de tarefas de replicação utilizando o Amazon CloudWatch.

Tópicos

- Segurança de rede para AWS Database Migration Service
- Usando SSL com AWS Database Migration Service
- Alterar a senha do banco de dados

Proteção de dados em AWS Database Migration Service

Criptografia de dados

Você pode ativar a criptografia para recursos de dados de endpoints de AWS DMS destino compatíveis. AWS DMS também criptografa conexões de AWS DMS AWS DMS e entre todos os seus endpoints de origem e destino. Além disso, você pode gerenciar as chaves que AWS DMS e seus endpoints de destino compatíveis usam para habilitar essa criptografia.

Tópicos

Criptografia inativa

AWS DMS suporta criptografia em repouso, permitindo que você especifique o modo de criptografia do lado do servidor que você deseja usar para enviar seus dados replicados para o Amazon S3 antes de serem copiados para endpoints de destino compatíveis. AWS DMS É possível especificar esse modo de criptografia definindo o atributo de conexão extra `encryptionMode` para o endpoint. Se essa `encryptionMode` configuração especificar o modo de criptografia da chave KMS, você também poderá criar AWS KMS chaves personalizadas especificamente para criptografar os dados de destino dos seguintes AWS DMS endpoints de destino:

- Amazon Redshift: para obter mais informações sobre a configuração do `encryptionMode`, consulte Configurações de endpoint ao utilizar o Amazon Redshift como destino do AWS DMS. Para obter mais informações sobre a criação de uma chave AWS KMS de criptografia personalizada, consulteCriar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift.
- Amazon S3: para obter mais informações sobre a configuração do `encryptionMode`, consulte Configurações de endpoint ao utilizar o Amazon S3 como destino para o AWS DMS. Para obter mais informações sobre a criação de uma chave AWS KMS de criptografia personalizada, consulteCriar chaves do AWS KMS para criptografar objetos de destino do Amazon S3.

Criptografia em trânsito

AWS DMS oferece suporte à criptografia em trânsito, garantindo que os dados que ele replica sejam movidos com segurança do endpoint de origem para o endpoint de destino. Isso inclui criptografar um bucket do S3 na instância de replicação usada por sua tarefa de replicação para armazenamento intermediário à medida que os dados são movidos pelo pipeline de replicação. Para criptografar conexões de tarefas com endpoints de origem e destino, AWS DMS use Secure Socket Layer (SSL) ou Transport Layer Security (TLS). Ao criptografar as conexões com os dois endpoints, AWS DMS garante que seus dados estejam seguros à medida que se movem do endpoint de origem para a tarefa de replicação e da tarefa para o endpoint de destino. Para obter mais informações sobre como usar SSL/TLS com, consulte AWS DMSUsando SSL com AWS Database Migration Service

AWS DMS suporta chaves padrão e personalizadas para criptografar o armazenamento de replicação intermediária e as informações de conexão. Essas chaves são gerenciadas com o AWS KMS. Para ter mais informações, consulte Configurando uma chave de criptografia e especificando permissões AWS KMS.

Gerenciamento de chaves

AWS DMS suporta chaves padrão ou personalizadas para criptografar o armazenamento de replicação, as informações de conexão e o armazenamento de dados de destino para determinados endpoints de destino. Você gerencia essas chaves usando AWS KMS o. Para ter mais informações, consulte Configurando uma chave de criptografia e especificando permissões AWS KMS.

Privacidade do tráfego entre redes

As conexões são fornecidas com proteção entre os endpoints de origem AWS DMS e de destino na mesma AWS região, seja em execução local ou como parte de um AWS serviço na nuvem. (Pelo menos um endpoint, origem ou destino, deve ser executado como parte de um AWS serviço na nuvem.) Essa proteção se aplica se esses componentes compartilharem a mesma nuvem privada virtual (VPC) ou existirem em VPCs separadas, se as VPCs estiverem todas na mesma região. AWS Para obter mais informações sobre as configurações de rede suportadas para AWS DMS, consulteConfigurar uma rede para uma instância de replicação. Para obter mais informações sobre as considerações de segurança ao usar essas configurações de rede, consulte Segurança de rede para AWS Database Migration Service.

Proteção de dados no DMS Fleet Advisor

O DMS Fleet Advisor coleta e analisa os metadados do banco de dados para determinar o tamanho certo do destino da migração. O DMS Fleet Advisor não acessa dados nas tabelas e não os transfere. Além disso, o DMS Fleet Advisor não rastreia a utilização de recursos do banco de dados e não acessa as estatísticas de utilização.

Você controla o acesso aos bancos de dados ao criar usuários de banco de dados que o DMS Fleet Advisor utiliza para trabalhar com os bancos de dados. Conceda os privilégios necessários a esses usuários. Para utilizar o DMS Fleet Advisor, conceda permissões de leitura aos usuários do banco de dados. O DMS Fleet Advisor não modifica os bancos de dados e não requer permissões de gravação. Para ter mais informações, consulte Criar usuários do banco de dados para o AWS DMS Fleet Advisor.

Você pode usar a criptografia de dados em seus bancos de dados. AWS DMS também criptografa conexões no DMS Fleet Advisor e em seus coletores de dados.

O coletor de dados do DMS utiliza a interface de programação de aplicações de proteção de dados (DPAPI) para criptografar, proteger e armazenar informações sobre o ambiente do cliente e as credenciais do banco de dados. O DMS Fleet Advisor armazena esses dados criptografados em um arquivo no servidor em que o coletor de dados do DMS funciona. O DMS Fleet Advisor não transfere esses dados desse servidor. Para obter mais informações sobre o DPAPI, consulte Como utilizar a proteção de dados.

Depois de instalar o coletor de dados do DMS, é possível visualizar todas as consultas que essa aplicação executa para coletar métricas. É possível executar o coletor de dados do DMS em modo off-line e revisar os dados coletados no servidor. Além disso, é possível revisar esses dados coletados no bucket do Amazon S3. Para ter mais informações, consulte Como funciona o coletor de dados do DMS?.

Gerenciamento de identidade e acesso para AWS Database Migration Service

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS DMS os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz AWS DMS.

Usuário do serviço — Se você usar o AWS DMS serviço para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais AWS DMS recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se

não for possível acessar um atributo no AWS DMS, consulte Solução de problemas AWS Database Migration Service de identidade e acesso.

Administrador de serviços — Se você é responsável pelos AWS DMS recursos da sua empresa, provavelmente tem acesso total AWS DMS a. É seu trabalho determinar quais AWS DMS recursos e recursos seus usuários do serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como sua empresa pode usar o IAM com AWS DMS, consulteComo AWS Database Migration Service funciona com o IAM.

Administrador do IAM: Se você for um administrador do IAM, talvez queira saber detalhes sobre como pode gravar políticas para gerenciar acesso ao AWS DMS. Para ver exemplos de políticas AWS DMS baseadas em identidade que você pode usar no IAM, consulte. AWS Database Migration Service exemplos de políticas baseadas em identidade

Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como usuário do Usuário raiz da conta da AWS IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia AWS IAM Identity Center do usuário e Utilizar a autenticação multifator (MFA) na AWS no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do Usuário do IAM.

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de um perfil) no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma função do IAM no AWS Management Console trocando de funções. Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para o uso de perfis, consulte Utilizar perfis do IAM no Guia do usuário do IAM.

Funções do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidades de terceiros no Guia do Usuário do IAM. Se você usar o Centro de identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões no Guia do usuário AWS IAM Identity Center .
- Permissões temporárias para usuários do IAM –- um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas –- é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service

(Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

- Função de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

- Aplicativos em execução no Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo AWS CLI solicitações de API. AWS É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Utilizar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar perfis do IAM, consulte Quando criar um perfil do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do Usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissões para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem presumir os perfis.

As políticas do IAM definem permissões para uma ação independente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criando políticas do IAM no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do Usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um atributo avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do Usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em. AWS Organizations AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte How SCPs work (Como os SCPs funcionam) no Guia do usuário do AWS Organizations .
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do

usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do Usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como AWS Database Migration Service funciona com o IAM

Antes de usar o IAM para gerenciar o acesso AWS DMS, você deve entender quais recursos do IAM estão disponíveis para uso AWS DMS. Para ter uma visão de alto nível de como AWS DMS e outros AWS serviços funcionam com o IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

Tópicos

- Políticas baseadas em identidade do AWS DMS
- Políticas baseadas em recursos do AWS DMS
- Autorização baseada em tags do AWS DMS
- Funções do IAM para AWS DMS
- Gerenciamento de identidade e acesso do DMS Fleet Advisor

Políticas baseadas em identidade do AWS DMS

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados e as condições sob as quais as ações são permitidas ou negadas. O AWS DMS é compatível com ações, recursos e chaves de condição específicos. Para saber mais sobre todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

As ações políticas AWS DMS usam o seguinte prefixo antes da ação:`dms:`. Por exemplo, para conceder permissão a alguém para criar uma tarefa de replicação com a operação da AWS DMS `CreateReplicationTask` API, você inclui a `dms:CreateReplicationTask` ação na política dessa pessoa. As declarações de política devem incluir um `NotAction` elemento `Action` ou. AWS DMS define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme a seguir.

```
"Action": [
      "dms:action1",
      "dms:action2"
```

Você também pode especificar várias ações utilizando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a ação a seguir:

```
"Action": "dms:Describe*"
```

Para ver uma lista de AWS DMS ações, consulte Ações definidas por AWS Database Migration Service no Guia do usuário do IAM.

Recursos

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

AWS DMS funciona com os seguintes recursos:

- Certificados
- Endpoints
- Assinaturas de eventos
- Instâncias de replicação
- Grupos de sub-rede (segurança) de replicação
- Tarefas de replicação

O recurso ou recursos AWS DMS necessários dependem da ação ou ações que você invoca. É necessária uma política que permita essas ações no recurso ou nos recursos associados especificados pelos ARNs de recurso.

Por exemplo, um recurso de AWS DMS endpoint tem o seguinte ARN:

```
arn:${Partition}:dms:${Region}:${Account}:endpoint/${InstanceId}
```

Para obter mais informações sobre o formato dos ARNs, consulte Amazon Resource Names (ARNs) e namespaces AWS de serviços.

Por exemplo, para especificar a instância de endpoint `1A2B3C4D5E6F7G8H9I0J1K2L3M` para a região `us-east-2` em sua instrução, utilize o ARN a seguir.

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/1A2B3C4D5E6F7G8H9I0J1K2L3M"
```

Para especificar todos os endpoints que pertencem a uma conta específica, utilize o caractere curinga (*):

```
"Resource": "arn:aws:dms:us-east-2:987654321098:endpoint/*"
```

Algumas AWS DMS ações, como as de criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve utilizar o caractere curinga (*).

```
"Resource": "*"
```

Algumas ações AWS DMS da API envolvem vários recursos. Por exemplo, o `StartReplicationTask` inicia e conecta uma tarefa de replicação para dois recursos de endpoint de banco de dados, uma origem e um destino, portanto, um usuário do IAM deve ter permissões para ler o endpoint de origem e gravar no endpoint de destino. Para especificar vários recursos em uma única instrução, separe os ARNs com vírgulas.

```
"Resource": [
      "resource1",
      "resource2" ]
```

Para obter mais informações sobre como controlar o acesso aos AWS DMS recursos usando políticas, consulteUsar nomes de recursos para controle de acesso. Para ver uma lista de tipos de recurso do AWS DMS e seus ARNs, consulte Tipos de recursos definidos pelo AWS Database Migration Service do Guia do usuário do IAM. Para saber com quais ações é possível especificar o ARN de cada recurso, consulte Ações definidas pelo AWS Database Migration Service.

Chaves de condição

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar

vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

AWS DMS define seu próprio conjunto de chaves de condição e também suporta o uso de algumas chaves de condição globais. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

AWS DMS define um conjunto de tags padrão que você pode usar em suas chaves de condição e também permite que você defina suas próprias tags personalizadas. Para ter mais informações, consulte Uso de tags para controlar o acesso.

Para ver uma lista de chaves de AWS DMS condição, consulte Chaves de condição AWS Database Migration Service no Guia do usuário do IAM. Para saber com quais ações e recursos é possível utilizar uma chave de condição, consulte Ações definidas pelo AWS Database Migration Service e Recursos definidos pelo AWS Database Migration Service.

Exemplos

Para ver exemplos de políticas AWS DMS baseadas em identidade, consulte. AWS Database Migration Service exemplos de políticas baseadas em identidade

Políticas baseadas em recursos do AWS DMS

Políticas baseadas em recursos são documentos de política JSON que especificam quais ações um diretor específico pode realizar em um determinado AWS DMS recurso e sob quais condições. AWS DMS oferece suporte a políticas de permissões baseadas em recursos para chaves de AWS KMS criptografia que você cria para criptografar dados migrados para endpoints de destino compatíveis.

Os endpoints de destino compatíveis incluem o Amazon Redshift e o Amazon S3. Usando políticas baseadas em recursos, é possível conceder a permissão para utilizar essas chaves de criptografia para outras contas para cada endpoint de destino.

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recurso. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso estão em AWS contas diferentes, você também deve conceder permissão à entidade principal para acessar o recurso. Conceda permissão anexando uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do Usuário do IAM.

O AWS DMS serviço oferece suporte a apenas um tipo de política baseada em recursos chamada política de chaves, que é anexada a uma chave de AWS KMS criptografia. Essa política define quais entidades principais (contas, usuários, perfis e usuários federados) podem criptografar dados migrados no endpoint de destino compatível.

Para saber como associar uma política baseada em recursos a uma chave de criptografia criada para os endpoints de destino compatíveis, consulte Criar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift e Criar chaves do AWS KMS para criptografar objetos de destino do Amazon S3.

Exemplos

Para obter exemplos de políticas AWS DMS baseadas em recursos, consulte. Exemplos de políticas baseadas em recursos para AWS KMS

Autorização baseada em tags do AWS DMS

Você pode anexar tags a AWS DMS recursos ou passar tags em uma solicitação para AWS DMS. Para controlar o acesso com base em tags, você fornece as informações da tag no elemento condicional de uma política usando a chave de `aws:TagKeys` condição `dms:ResourceTag/`*`key-name`*`aws:RequestTag/`*`key-name`*, ou. AWS DMS define um conjunto de tags padrão que você pode usar em suas chaves de condição e também permite que você defina suas próprias tags personalizadas. Para ter mais informações, consulte Uso de tags para controlar o acesso.

Para obter um exemplo de política baseada em identidade que limita o acesso a um recurso com base em tags, consulte Acessar recursos com base em tags do AWS DMS.

Funções do IAM para AWS DMS

Uma função do IAM é uma entidade dentro da sua AWS conta que tem permissões específicas.

Usando credenciais temporárias com AWS DMS

É possível utilizar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando operações de AWS STS API, como AssumeRoleou GetFederationToken.

AWS DMS suporta o uso de credenciais temporárias.

Funções vinculadas a serviço

As funções vinculadas ao serviço permitem que AWS os serviços acessem recursos em outros serviços para concluir uma ação em seu nome. Os perfis vinculados a serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados a serviço.

Para obter detalhes sobre como criar ou gerenciar funções AWS DMS vinculadas a serviços, consulte. Usar perfis vinculados a serviço

Perfis de serviço

Esse atributo permite que um serviço assuma um perfil de serviço em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do IAM pode alterar as permissões para esse perfil. Porém, fazer isso pode alterar a funcionalidade do serviço.

AWS DMS oferece suporte a dois tipos de funções de serviço que você deve criar para usar determinados endpoints de origem ou destino:

- Funções com permissões para permitir que o AWS DMS acesse os seguintes endpoints de origem e destino (ou seus recursos):
  - Amazon DynamoDB como destino: para obter mais informações, consulte Pré-requisitos para a utilização do DynamoDB como destino do AWS Database Migration Service.

- OpenSearch como alvo — Para obter mais informações, consultePré-requisitos para utilizar o Amazon OpenSearch Service como destino do AWS Database Migration Service.
- Amazon Kinesis como destino: para obter mais informações, consulte Pré-requisitos para usar um stream de dados do Kinesis como destino para AWS Database Migration Service.
- Amazon Redshift como destino: é necessário criar o perfil especificado somente para criar uma chave de criptografia do KMS personalizada a fim de criptografar os dados de destino ou especificar um bucket do S3 personalizado para conter o armazenamento de tarefas intermediárias. Para obter mais informações, consulte Criar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift ou Configurações do bucket do Amazon S3.
- Amazon S3 como origem ou como destino: para obter mais informações, consulte Pré-requisitos ao usar o Amazon S3 como fonte para AWS DMS ou Pré-requisitos da utilização do Amazon S3 como destino.

  Por exemplo, para ler dados de um endpoint de origem do S3 ou enviar dados para um endpoint de destino do S3, é necessário criar um perfil de serviço como um pré-requisito para acessar o S3 para cada uma dessas operações de endpoint.

- Funções com permissões necessárias para usar a API AWS CLI e AWS DMS — Duas funções do IAM que você precisa criar são `dms-vpc-role` e. `dms-cloudwatch-logs-role` Se você usa o Amazon Redshift como banco de dados de destino, você também deve criar e adicionar a função do IAM `dms-access-for-endpoint` à sua AWS conta. Para ter mais informações, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

Escolha de uma função do IAM em AWS DMS

Se você usa a API AWS CLI ou a API do AWS DMS para a migração do banco de dados, deve adicionar determinadas funções do IAM à sua AWS conta antes de poder usar os recursos do AWS DMS. Duas delas são `dms-vpc-role` e `dms-cloudwatch-logs-role`. Se você usa o Amazon Redshift como banco de dados de destino, você também deve adicionar a função do IAM `dms-access-for-endpoint` à sua AWS conta. Para ter mais informações, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

Gerenciamento de identidade e acesso do DMS Fleet Advisor

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, além das condições sob as quais as ações são permitidas ou negadas. O DMS Fleet Advisor é compatível com ações, recursos e chaves de condição específicos. Para saber mais sobre

todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

O DMS Fleet Advisor utiliza perfis do IAM para acessar o Amazon Simple Storage Service. Uma função do IAM é uma entidade dentro da sua AWS conta que tem permissões específicas. Para ter mais informações, consulte Criar recursos do IAM.

AWS Database Migration Service exemplos de políticas baseadas em identidade

Por padrão, os usuários e os perfis do IAM não têm permissão para criar ou modificar recursos do AWS DMS . Eles também não podem realizar tarefas usando a AWS API AWS Management Console AWS CLI, ou. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e perfis permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM utilizando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

- Melhores práticas de política
- Usar o console do AWS DMS
- Permitir que usuários visualizem suas próprias permissões
- Acessar um bucket do Amazon S3
- Acessar recursos com base em tags do AWS DMS

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir AWS DMS recursos em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões

definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do Usuário do IAM.

- Aplique permissões de privilégio mínimo –- ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do Usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso –- você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode gravar uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: Condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais –- o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de 100 verificações de política e recomendações acionáveis para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas Recomendadas de Segurança no IAM no Guia do Usuário do IAM.

Usar o console do AWS DMS

A política a seguir fornece acesso ao AWS DMS, incluindo o console do AWS DMS, e também especifica permissões para determinadas ações necessárias de outros serviços da Amazon, como o Amazon EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
    ]
}
```

Um detalhamento dessas permissões pode ajudá-lo a entender melhor por que cada uma é necessária para utilizar o console.

A seção a seguir é necessária para permitir que o usuário liste suas chaves do AWS KMS disponíveis e o alias para exibição no console. Essa entrada não será necessária se você souber o Nome de recurso da Amazon (ARN) da chave do KMS e estiver usando apenas a AWS Command Line Interface (AWS CLI).

```
{
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

A seção a seguir é necessária para determinados tipos de endpoints que exigem que um ARN de perfil seja passado com o endpoint. Além disso, se as AWS DMS funções necessárias não forem criadas com antecedência, o AWS DMS console poderá criar a função. Se todos os perfis forem configurados com antecedência, tudo isso será necessário em `iam:GetRole` e `iam:PassRole`.

Para obter mais informações sobre funções, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

```
{
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

A seção a seguir é obrigatória porque AWS DMS precisa criar a instância do Amazon EC2 e configurar a rede para a instância de replicação criada. Esses recursos existem na conta do cliente, por isso, a capacidade de executar essas ações em nome do cliente é necessária.

```
{
            "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"
        }
```

A seção a seguir é necessária para permitir que o usuário possa visualizar as métricas da instância de replicação.

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
```

```
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

Esta seção é necessária para permitir que o usuário veja os logs de replicação.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

O console do AWS DMS cria várias funções que são automaticamente anexadas à sua AWS conta quando você usa o console do AWS DMS. Se você usar o AWS Command Line Interface (AWS CLI) ou a API do AWS DMS para sua migração, precisará adicionar essas funções à sua conta. Para obter mais informações sobre essas funções, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

Para obter mais informações sobre os requisitos para usar essa política para acessar o AWS DMS, consultePermissões do IAM necessárias para utilizar o AWS DMS.

Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
```

```
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Acessar um bucket do Amazon S3

AWS O DMS usa buckets Amazon S3 como armazenamento intermediário para migração de banco de dados. Normalmente, o AWS DMS gerencia buckets S3 padrão para essa finalidade. No entanto, em certos casos, especialmente quando você usa a AWS CLI ou a API do AWS DMS, o AWS DMS permite que você especifique seu próprio bucket do S3. Por exemplo, é possível especificar seu próprio bucket do S3 para migrar dados para um endpoint de destino do Amazon Redshift. Nesse caso, você precisa criar uma função com permissões com base na `AmazonDMSRedshiftS3Role` política AWS gerenciada.

O exemplo a seguir mostra uma versão da política `AmazonDMSRedshiftS3Role`. Ele permite que o AWS DMS conceda a um usuário do IAM em sua AWS conta acesso a um dos seus buckets do Amazon S3. Ela também permite que o usuário adicione, atualize e exclua objetos.

Além de conceder as permissões `s3:PutObject`, `s3:GetObject` e `s3:DeleteObject` ao usuário, a política também concede as permissões `s3:ListAllMyBuckets`, `s3:GetBucketLocation` e `s3:ListBucket`. Estas são permissões adicionais, exigidas pelo

console. Outras permissões permitem que o AWS DMS gerencie o ciclo de vida do bucket. Além disso, a ação `s3:GetObjectAcl` é necessária para poder copiar objetos.

```
{
    "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-*"
        }
    ]
}
```

Para obter mais informações sobre como criar um perfil baseado nessa política, consulte Configurações do bucket do Amazon S3.

Acessar recursos com base em tags do AWS DMS

É possível utilizar condições na política baseada em identidade para controlar o acesso aos recursos do AWS DMS com base em tags. Este exemplo mostra como você pode criar uma política que permita acesso a todos os endpoints do AWS DMS. No entanto, a permissão será concedida somente se a tag de banco de dados de endpoint `Owner` tiver o valor do nome desse usuário.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:*:*:endpoint/*",
            "Condition": {
                "StringEquals": {"dms:endpoint-tag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

É possível anexar essa política aos usuários do IAM na sua conta. Se um usuário chamado `richard-roe` tentar acessar um AWS DMS endpoint, o banco de dados do endpoint deverá ser marcado como `Owner=richard-roe` ou. `owner=richard-roe` Caso contrário, esse usuário terá o acesso negado. A chave da tag de condição `Owner` corresponde a `Owner` e a `owner` porque os nomes das chaves de condição não fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulte Elementos de política JSON do IAM: condição no Guia do usuário do IAM.

Exemplos de políticas baseadas em recursos para AWS KMS

AWS O DMS permite que você crie chaves de AWS KMS criptografia personalizadas para criptografar dados de endpoint de destino compatíveis. Para saber como criar e associar uma política de chave à chave de criptografia criada para a criptografia compatível de dados de destino, consulte Criar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift e Criar chaves do AWS KMS para criptografar objetos de destino do Amazon S3.

Tópicos

- Uma política para uma chave de AWS KMS criptografia personalizada para criptografar dados de destino do Amazon Redshift
- Uma política para uma chave de AWS KMS criptografia personalizada para criptografar dados de destino do Amazon S3

Uma política para uma chave de AWS KMS criptografia personalizada para criptografar dados de destino do Amazon Redshift

O exemplo a seguir mostra o JSON para a política de chave criada para uma chave de criptografia do AWS KMS criada para criptografar dados de destino do Amazon Redshift.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/Admin"
        ]
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key",
```

```
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-Redshift-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-Redshift-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": true
        }
      }
    }
  ]
}
```

Aqui, é possível ver onde a política de chave faz referência ao perfil para acessar dados do endpoint de destino do Amazon Redshift criado antes de criar a chave. No exemplo, é `DMS-Redshift-endpoint-access-role`. Também é possível ver as diferentes ações chave permitidas para os diferentes principais (usuários e funções). Por exemplo, qualquer usuário com `DMS-Redshift-endpoint-access-role` pode criptografar, descriptografar e criptografar novamente os dados

de destino. Esse usuário também pode gerar chaves de dados para exportação para criptografar os dados externos. AWS KMS Eles também podem retornar informações detalhadas sobre uma AWS KMS chave, como a chave que você acabou de criar. Além disso, esse usuário pode gerenciar anexos aos recursos da AWS , como o endpoint de destino.

Uma política para uma chave de AWS KMS criptografia personalizada para criptografar dados de destino do Amazon S3

O exemplo a seguir mostra o JSON da política de chave criada para uma chave de criptografia do AWS KMS que você cria para criptografar dados de destino do Amazon S3.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/Admin"
        ]
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
```

```
        "kms:Disable*",
        "kms:Get*",
        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-S3-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::987654321098:role/DMS-S3-endpoint-access-role"
        ]
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": true
```

```
        }
      }
    }
  ]
```

Aqui, é possível ver onde a política de chave faz referência ao perfil para acessar dados de endpoint de destino do Amazon S3 criado antes de criar a chave. No exemplo, é `DMS-S3-endpoint-access-role`. Também é possível ver as diferentes ações chave permitidas para os diferentes principais (usuários e funções). Por exemplo, qualquer usuário com `DMS-S3-endpoint-access-role` pode criptografar, descriptografar e criptografar novamente os dados de destino. Esse usuário também pode gerar chaves de dados para exportação para criptografar os dados externos. AWS KMS Eles também podem retornar informações detalhadas sobre uma AWS KMS chave, como a chave que você acabou de criar. Além disso, esse usuário pode gerenciar anexos para recursos da AWS , como o endpoint de destino.

Utilizar segredos para acessar endpoints do AWS Database Migration Service

Pois AWS DMS, um segredo é uma chave criptografada que você pode usar para representar um conjunto de credenciais de usuário para autenticar, por meio de autenticação secreta, a conexão de banco de dados para uma AWS DMS fonte compatível ou um endpoint de destino. Para um endpoint Oracle que também usa o Oracle Automatic Storage Management (ASM), é AWS DMS necessário um segredo adicional que represente as credenciais do usuário para acessar o Oracle ASM.

Você pode criar o segredo ou segredos AWS DMS necessários para a autenticação secreta usando AWS Secrets Manager um serviço para criar, armazenar e recuperar credenciais com segurança para acessar aplicativos, serviços e recursos de TI na nuvem e no local. Isso inclui compatibilidade com a alternância periódica automática do valor do segredo criptografado sem sua intervenção, fornecendo um nível adicional de segurança às credenciais. Ativar a rotação de valores secretos AWS Secrets Manager também garante que essa rotação de valores secretos ocorra sem qualquer efeito em qualquer migração de banco de dados que dependa do segredo. Para autenticar uma conexão de banco de dados do endpoint secretamente, crie um segredo cuja identidade ou ARN você atribui ao `SecretsManagerSecretId` e inclui nas configurações do endpoint. Para autenticar um Oracle ASM como parte de um endpoint do Oracle secretamente, crie um segredo cuja identidade ou ARN você atribui ao `SecretsManagerOracleAsmSecretId` e inclui nas configurações do endpoint.

> **Note**
>
> Não é possível utilizar credenciais mestras gerenciadas pelo Amazon RDS Aurora. Essas credenciais não incluem informações de host ou porta, que AWS DMS precisam estabelecer conexões. Em vez disso, crie um novo usuário e segredo. Para obter informações sobre como criar um usuário e um segredo, consulte Usando o AWS Management Console para criar uma função de acesso secreta e secreta a seguir.

Para obter mais informações sobre AWS Secrets Manager, consulte O que é o AWS Secrets Manager? no Guia do AWS Secrets Manager usuário.

AWS DMS oferece suporte à autenticação secreta para os seguintes bancos de dados locais ou AWS gerenciados em endpoints de origem e destino compatíveis:

- Amazon DocumentDB
- IBM Db2 LUW
- Microsoft SQL Server
- MongoDB
- MySQL
- Oracle
- PostgreSQL
- Amazon Redshift
- SAP ASE

Para se conectar a qualquer um desses bancos de dados, você tem a opção de inserir um dos seguintes conjuntos de valores, mas não ambos, como parte das configurações do endpoint:

- Valores de texto não criptografado para autenticar a conexão do banco de dados utilizando as configurações `UserName`, `Password`, `ServerName` e `Port`. Para um endpoint do Oracle que também utiliza o Oracle ASM, inclua valores adicionais de texto não criptografado para autenticar o ASM utilizando as configurações `AsmUserName`, `AsmPassword` e `AsmServerName`.
- Autenticação de segredo utilizando valores para as configurações `SecretsManagerSecretId` e `SecretsManagerAccessRoleArn`. Para um endpoint do Oracle que utiliza o Oracle ASM, inclua valores adicionais nas configurações `SecretsManagerOracleAsmSecretId` e

`SecretsManagerOracleAsmAccessRoleArn`. Os valores de segredos dessas configurações podem incluir o seguinte para:

- `SecretsManagerSecretId`: o nome do recurso da Amazon (ARN) completo, o ARN parcial ou o nome amigável de um segredo que você criou para acesso ao banco de dados do endpoint no AWS Secrets Manager.
- `SecretsManagerAccessRoleArn`— O ARN de uma função de acesso secreto que você criou no IAM para fornecer AWS DMS acesso a esse `SecretsManagerSecretId` segredo em seu nome.
- `SecretsManagerOracleAsmSecretId`: o nome do recurso da Amazon (ARN) completo, o ARN parcial ou o nome amigável de um segredo que você criou para acesso ao Oracle ASM no AWS Secrets Manager.
- `SecretsManagerOracleAsmAccessRoleArn`: o ARN de um perfil de acesso secreto que você criou no IAM para fornecer ao AWS DMS acesso a esse segredo do `SecretsManagerOracleAsmSecretId` em seu nome.

> **Note**
>
> Você também pode usar uma única função de acesso secreto para fornecer AWS DMS acesso tanto ao `SecretsManagerSecretId` segredo quanto ao `SecretsManagerOracleAsmSecretId` segredo. Se você criar essa único acesso secreto para ambos os segredos, atribua o mesmo ARN para esse perfil de acesso a `SecretsManagerAccessRoleArn` e a `SecretsManagerOracleAsmAccessRoleArn`. Por exemplo, se o perfil de acesso secreto para ambos os segredos tiver seu ARN atribuído à variável `ARN2xsecrets`, você poderá definir essas configurações de ARN da seguinte forma:
>
> ```
> SecretsManagerAccessRoleArn = ARN2xsecrets;
> SecretsManagerOracleAsmAccessRoleArn = ARN2xsecrets;
> ```

Para obter mais informações sobre como criar esses valores, consulte Usando o AWS Management Console para criar uma função de acesso secreta e secreta.

Depois de criar e especificar as configurações do segredo necessário e do endpoint do perfil de acesso secreto para seus endpoints, atualize as permissões nas contas de usuário que executarão

a solicitação de API `CreateEndpoint` ou `ModifyEndpoint` com esses detalhes do segredo. Certifique-se de que essas permissões de conta incluam a `IAM:GetRole` permissão na função de acesso secreto e a `SecretsManager:DescribeSecret` permissão na chave secreta. AWS DMS requer essas permissões para validar a função de acesso e seu segredo.

Como fornecer e verificar as permissões de usuário necessárias

1. Faça login no AWS Management Console e abra o AWS Identity and Access Management console emhttps://console.aws.amazon.com/iam/.
2. Escolha Usuários e escolha o ID de usuário utilizado para fazer chamadas de API `CreateEndpoint` e `ModifyEndpoint`.
3. Na guia Permissões, escolha {} JSON.
4. Verifique se o usuário tem as permissões a seguir:

```
{
 "Statement": [{
   "Effect": "Allow",
   "Action": [
    "iam:GetRole",
    "iam:PassRole"
   ],
   "Resource": "SECRET_ACCESS_ROLE_ARN"
  },
  {
   "Effect": "Allow",
   "Action": "secretsmanager:DescribeSecret",
   "Resource": "SECRET_ARN"
  }
 ]
}
```

5. Se o usuário não tiver essas permissões, adicione-as.
6. Se estiver utilizando um perfil do IAM para fazer chamadas de API do DMS, repita as etapas acima para o respectivo perfil.
7. Abra um terminal e use o AWS CLI para validar se as permissões foram dadas corretamente assumindo a função ou o usuário usado acima.

   a. Valide a permissão do usuário ao SecretAccessRole usar o `get-role` comando IAM.

```
aws iam get-role --role-name ROLE_NAME
```

Substitua *ROLE_NAME* pelo nome de `SecretsManagerAccessRole`.

Se o comando retornar uma mensagem de erro, verifique se as permissões foram concedidas corretamente.

b. Valide a permissão do usuário no segredo utilizando o comando `describe-secret` do Secrets Manager.

```
aws secretsmanager describe-secret --secret-id SECRET_NAME OR SECRET_ARN --
region=REGION_NAME
```

O usuário pode ser o nome amigável, o ARN parcial ou o ARN completo. Para obter mais informações, consulte describe-secret.

Se o comando retornar uma mensagem de erro, verifique se as permissões foram concedidas corretamente.

Usando o AWS Management Console para criar uma função de acesso secreta e secreta

Você pode usar o AWS Management Console para criar um segredo para autenticação de endpoint e criar a política e a função para permitir o acesso AWS DMS ao segredo em seu nome.

Para criar um segredo usando o AWS Management Console que AWS DMS pode ser usado para autenticar um banco de dados para conexões de endpoint de origem e destino

1. Faça login no AWS Management Console e abra o AWS Secrets Manager console emhttps://console.aws.amazon.com/secretsmanager/.
2. Selecione Armazenar um novo segredo.
3. Em Selecionar tipo de segredo na página Armazenar um novo segredo, escolha Outro tipo de segredo e escolha Texto sem formatação.

> **Note**
>
> Esse é o único local em que você precisa inserir credenciais de texto não criptografado para conectar-se ao banco de dados do endpoint daqui em diante.

4. No campo Texto sem formatação:

   - Para um segredo cuja identidade você atribui a `SecretsManagerSecretId`, insira a seguinte estrutura JSON.

```
{
  "username": db_username,
  "password": db_user_password,
  "port": db_port_number,
  "host": db_server_name
}
```

> **Note**
>
> Essa é a lista mínima de membros JSON necessária para autenticar o banco de dados do endpoint. É possível adicionar qualquer configuração adicional de endpoint JSON como membros JSON, tudo em letras minúsculas que desejar. No entanto, o AWS DMS ignora quaisquer membros JSON adicionais para autenticação de endpoint.

Aqui, o *db_username* é o nome do usuário que está acessando o banco de dados, *db_user_password* é a senha do usuário do banco de dados, *db_port_number* é o número da porta para acessar o banco de dados e *db_server_name* é o nome do servidor de banco de dados (endereço) na web, como no exemplo a seguir.

```
{
  "username": "admin",
  "password": "some_password",
  "port": "8190",
  "host": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com"
}
```

- Para um segredo cuja identidade você atribui a `SecretsManagerOracleAsmSecretId`, insira a seguinte estrutura JSON.

```
{
  "asm_user": asm_username,
  "asm_password": asm_user_password,
  "asm_server": asm_server_name
}
```

> **Note**
>
> Essa é a lista mínima de membros JSON necessária para autenticar o Oracle ASM para um endpoint do Oracle. Também é a lista completa que pode ser especificada com base nas configurações de endpoint do Oracle ASM disponíveis.

Aqui, *asm_username* é o nome do usuário que está acessando o Oracle ASM, *asm_user_password* é a senha do usuário do Oracle ASM e *asm_server_name* é o nome do servidor do Oracle ASM (endereço) na web, incluindo a porta, como no exemplo a seguir.

```
{
  "asm_user": "oracle_asm_user",
  "asm_password": "oracle_asm_password",
  "asm_server": "oracle101.abcdefghij.us-east-1.rds.amazonaws.com:8190/+ASM"
}
```

5. Selecione uma chave AWS KMS de criptografia para criptografar o segredo. Você pode aceitar a chave de criptografia padrão criada para seu serviço AWS Secrets Manager ou selecionar uma AWS KMS chave criada por você.
6. Especifique um nome para referenciar esse segredo e uma descrição opcional.
   Esse é o nome amigável utilizado como o valor de `SecretsManagerSecretId` ou `SecretsManagerOracleAsmSecretId`.
7. Se você quiser ativar a rotação automática no segredo, precisará selecionar ou criar uma AWS Lambda função com permissão para alternar as credenciais do segredo conforme descrito. No entanto, antes de definir a alternância automática para utilizar o função do Lambda, verifique as configurações do perfil adicionam os quatro caracteres a seguir ao valor da variável de ambiente `EXCLUDE_CHARACTERS`.

```
;.:+{}
```

AWS DMS não permite esses caracteres em senhas usadas para credenciais de endpoint. Configurar o função do Lambda para excluí-los impede que o AWS Secrets Manager gere esses caracteres como parte de seus valores de senha alternados. Depois de definir a rotação automática para usar sua função Lambda, gira AWS Secrets Manager imediatamente o segredo para validar sua configuração secreta.

> **Note**
>
> Dependendo da configuração do mecanismo de banco de dados, o banco de dados talvez não busque as credenciais alternadas. Nesse caso, é necessário reiniciar a tarefa manualmente para atualizar as credenciais.

8. Revise e guarde seu segredo em AWS Secrets Manager. Em seguida, você pode pesquisar cada segredo pelo nome amigável e recuperar o ARN secreto como o valor `SecretsManagerOracleAsmSecretId` ou conforme apropriado `SecretsManagerSecretId` para autenticar o acesso à sua conexão de banco de dados de endpoint e ao Oracle ASM (se usado). AWS Secrets Manager

Para criar a política de acesso secreto e a função para definir seu **`SecretsManagerAccessRoleArn`** ou**`SecretsManagerOracleAsmAccessRoleArn`**, AWS DMS o que AWS Secrets Manager permite acessar seu segredo apropriado

1. Faça login AWS Management Console e abra o console AWS Identity and Access Management (IAM) em https://console.aws.amazon.com/iam/.
2. Escolha Políticas e Criar política.
3. Escolha JSON e insira a política a seguir para ativar o acesso e a descriptografia do segredo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": secret_arn,
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                        "kms:Decrypt",
                        "kms:DescribeKey"
                    ],
            "Resource": kms_key_arn,
        }
    ]
}
```

Aqui, *secret_arn* é o ARN do segredo, que é possível obter de `SecretsManagerSecretId` ou de `SecretsManagerOracleAsmSecretId` conforme apropriado, e *kms_key_arn* é o ARN da chave do AWS KMS que você está utilizando para criptografar o segredo, como no exemplo a seguir.

```
{
    "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**
>
> Se você usar a chave de criptografia padrão criada por AWS Secrets Manager, não precisará especificar as AWS KMS permissões para*kms_key_arn*.

Se quiser que a política forneça acesso aos dois segredos, basta especificar um objeto de recurso JSON adicional para o outro *secret_arn*.
Se o segredo estiver em uma conta diferente, o perfil `SecretsManagerAccessRoleArn` precisará de uma política adicional para verificar o segredo entre contas. Para esses casos de uso, adicione a ação `secretsmanager:DescribeSecret` à política. Para obter mais detalhes sobre como configurar um segredo entre contas, consulte Permissões para AWS segredos do Secrets Manager para usuários em uma conta diferente.

4. Revise e crie a política com um nome amigável e uma descrição opcional.
5. Escolha Funções e Criar função.
6. Escolha Serviço da AWS como o tipo de entidade confiável.
7. Escolha DMS na lista de serviços como o serviço confiável e escolha Próximo: Permissões.
8. Pesquise e anexe a política que criada na etapa 4 e continue para adicionar quaisquer tags e revisar o perfil. Nesse ponto, edite as relações de confiança da função para usar seu diretor de serviço AWS DMS regional como entidade confiável. Essa entidade principal tem o seguinte formato.

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

Aqui, *region-name* é nome da sua região, como `us-east-1`. Assim, segue um diretor de serviço AWS DMS regional para essa região.

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

9. Depois de editar a entidade confiável do perfil, crie o perfil com um nome amigável e uma descrição opcional. Agora é possível pesquisar o novo perfil pelo nome amigável no IAM e recuperar o ARN do perfil como o valor `SecretsManagerAccessRoleArn` ou `SecretsManagerOracleAsmAccessRoleArn` para autenticar sua conexão com o banco de dados do endpoint.

Como utilizar o gerenciador de segredos com uma instância de replicação em uma sub-rede privada

1. Crie um endpoint da VPC do gerenciador secreto e anote o DNS do endpoint. Para obter mais informações sobre como criar um endpoint da VPC do gerenciador de segredos, consulte

Conectar-se ao Secrets Manager por meio de um endpoint da VPC no Guia do usuário do AWS Secrets Manager.

2. Anexe o grupo de segurança da instância de replicação ao endpoint da VPC do gerenciador de segredos.
3. Para as regras de saída do grupo de segurança da instância de replicação, permita todo o tráfego para o destino `0.0.0.0/0`.
4. Defina o atributo de conexão adicional `secretsManagerEndpointOverride=`*`secretsManager endpoint DNS`* do endpoint para fornecer o DNS do endpoint da VPC ao gerenciador de segredo, conforme mostrado no exemplo a seguir.

```
secretsManagerEndpointOverride=vpce-1234a5678b9012c-12345678.secretsmanager.eu-
west-1.vpce.amazonaws.com
```

Usar perfis vinculados a serviço do AWS DMS

O AWS Database Migration Service utiliza perfis vinculados a serviço do AWS Identity and Access Management (IAM). O perfil vinculado a serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao AWS DMS. Os perfis vinculados a serviços são predefinidos pelo AWS DMS e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Um perfil vinculado a serviço facilita a configuração do AWS DMS porque você não precisa adicionar as permissões necessárias manualmente. O AWS DMS define as permissões de seus perfis vinculados a serviço e, a menos que definido em contrário, somente o AWS DMS pode assumir seus perfis. As permissões definidas incluem a política de confiança e a política de permissões, e essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado a serviço poderá ser excluído somente após a exclusão dos recursos relacionados. Isso protege seus recursos do AWS DMS, pois você não pode remover por engano as permissões para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com perfis vinculados aos serviços, consulte Serviços da AWS que funcionam com o IAM e procure os serviços que apresentam Sim na coluna Funções vinculadas aos serviços. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Perfis vinculados a serviço para recursos do AWS DMS

Tópicos

Perfis vinculados a serviço do AWS DMS Fleet Advisor

O AWS DMS Fleet Advisor utiliza o perfil vinculado a serviço chamado AWSServiceRoleForDMSFleetAdvisor: o DMS Fleet Advisor utiliza esse perfil vinculado a serviço para gerenciar as métricas do Amazon CloudWatch. Esse perfil vinculado a serviço é anexado à seguinte política gerenciada: `AWSDMSFleetAdvisorServiceRolePolicy`. Para obter atualizações nessa política, consulte AWS políticas gerenciadas para AWS Database Migration Service.

O perfil vinculado a serviço AWSServiceRoleForDMSFleetAdvisor confia nos seguintes serviços para assumir o perfil:

- `dms-fleet-advisor.amazonaws.com`

A política de permissões chamada AWSDMSFleetAdvisorServiceRolePolicy permite que o AWS DMS Fleet Advisor conclua as seguintes ações nos recursos especificados:

- Ação: `cloudwatch:PutMetricData` em `all AWS resources`

  Esse perfil permite que as entidades principais publiquem pontos de dados de métricas no Amazon CloudWatch do AWS DMS. O Fleet Advisor exige essa permissão para exibir gráficos com métricas de banco de dados do CloudWatch.

O exemplo de código a seguir mostra a política AWSDMSFleetAdvisorServiceRolePolicy utilizada para criar o perfil AWSDMSFleetAdvisorServiceRolePolicy.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
```

```
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
            }
        }
    }
}
```

É necessário configurar permissões para permitir que uma entidade do IAM, como um usuário, grupo ou perfil, crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões vinculadas a serviço no Guia do usuário do IAM.

Criar um perfil vinculado a serviço para o AWS DMS Fleet Advisor

É possível utilizar o console do IAM para criar um perfil vinculado a serviço com o caso de uso do DMS: Fleet Advisor. Na AWS CLI ou na API do AWS, crie um perfil vinculado a serviço com o nome de serviço `dms-fleet-advisor.amazonaws.com`. Para obter mais informações, consulte Criar um perfil vinculado a serviço no Guia do usuário do IAM. Se você excluir esse perfil vinculado a serviço, será possível utilizar esse mesmo processo para criar o perfil novamente.

Crie esse perfil antes de criar um coletor de dados. O DMS Fleet Advisor utiliza esse perfil para exibir gráficos com métricas de banco de dados no AWS Management Console. Para obter mais informações, consulte Criar um coletor de dados.

Editar um perfil vinculado a serviço para o AWS DMS Fleet Advisor

O AWS DMS não permite que você edite o perfil vinculado a serviço AWSServiceRoleForDMSFleetAdvisor. Depois de criar um perfil vinculado a serviço, não será possível alterar o nome do perfil, pois várias entidades podem fazer referência ao perfil. No entanto, é possível editar a descrição do perfil utilizando o IAM. Para obter mais informações, consulte Editar um perfil vinculado a serviço no Guia do usuário do IAM.

Excluir um perfil vinculado a serviço do AWS DMS Fleet Advisor

Se não for mais necessário utilizar um recurso ou serviço que requer um perfil vinculado a serviço, é recomendável excluí-lo. Dessa forma, não haverá uma entidade não utilizada que não seja monitorada ou mantida ativamente. No entanto, você deve limpar os recursos do perfil vinculado a serviço antes de exclui-lo manualmente.

> **Note**
>
> Se o serviço AWS DMS estiver utilizando o perfil quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir os recursos do AWS DMS utilizados por AWSServiceRoleForDMSFleetAdvisor

1. Faça login no AWS Management Console e abra o console do AWS DMS em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, escolha Coletores de dados em Descobrir. A página Coletores de dados é aberta.
3. Escolha o coletor de dados e escolha Excluir.
4. Para confirmar a exclusão, insira o nome do conector no campo de entrada de texto. Escolha Excluir.

> **Important**
>
> Quando você exclui um coletor de dados do DMS, o DMS Fleet Advisor exclui todos os bancos de dados do Inventário que você descobriu utilizando esse coletor.

Depois de excluir todos os coletores de dados, é possível excluir o perfil vinculado a serviço.

Como excluir manualmente o perfil vinculado a serviço utilizando o IAM

Utilize o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado a serviço AWSServiceRoleForDMSFleetAdvisor. Para obter mais informações, consulte Excluir um perfil vinculado a serviço no Guia do usuário do IAM.

Regiões compatíveis com os perfis vinculados a serviço do AWS DMS Fleet Advisor

O AWS DMS Fleet Advisor é compatível com a utilização de perfis vinculados a serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Com suporte Regiões da AWS.

Perfil vinculado a serviço do AWS DMS com Tecnologia Sem Servidor

AWS DMS O Serverless usa a função vinculada ao serviço chamada. AWSServiceRoleForDMSServerless AWS DMS usa essa função vinculada ao serviço para criar e gerenciar AWS DMS recursos em seu nome, como métricas da Amazon CloudWatch . AWS DMS usa essa função para que você só precise se preocupar com as replicações. Essa função vinculada ao serviço é anexada à seguinte política gerenciada: `AWSDMSServerlessServiceRolePolicy`. Para obter atualizações dessa política, consulte AWS políticas gerenciadas para AWS Database Migration Service.

A função AWSServiceRoleForDMSServerless vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `dms.amazonaws.com`

O exemplo de código a seguir mostra a AWSDMSServerlessServiceRolePolicy política que você usa para criar a AWSServiceRoleForDMSServerless função.

```
{
    "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:*"
            ]
        }
    ]
}
```

É necessário configurar permissões para permitir que uma entidade do IAM, como um usuário, grupo ou perfil, crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte Permissões de perfil vinculado a serviços no Guia do usuário do IAM.

Criar um perfil vinculado a serviço para o AWS DMS com Tecnologia Sem Servidor

Quando você cria uma replicação, a tecnologia AWS DMS sem servidor cria programaticamente uma AWS DMS função vinculada ao serviço sem servidor. É possível visualizar esse perfil no console do IAM. É possível optar por criar esse perfil manualmente. Para criar a função manualmente, use o console do IAM para criar uma função vinculada ao serviço com o caso de uso do DMS. Na AWS CLI ou na AWS API, crie uma função vinculada ao serviço usando `dms.amazonaws.com` o nome do serviço. Para obter mais informações, consulte Criar um perfil vinculado a serviço no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

> **Note**
>
> Se você excluir um perfil enquanto tiver replicações na conta, a replicação resultará em uma falha.

Editar um perfil vinculado a serviço para o AWS DMS com Tecnologia Sem Servidor

AWS DMS não permite que você edite a função AWSServiceRoleForDMSServerless vinculada ao serviço. Depois que você criar um perfil vinculado ao serviço, não poderá alterar o nome do perfil, pois várias entidades podem fazer referência ao perfil. No entanto, você poderá editar a descrição do perfil usando o IAM. Para ter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir um perfil vinculado a serviço do AWS DMS com Tecnologia Sem Servidor

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, não haverá uma entidade não utilizada que não seja monitorada ou mantida ativamente. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

> **Note**
>
> Se o AWS DMS serviço estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir AWS DMS recursos usados pelo AWSServiceRoleForDMSServerless

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. No painel de navegação, escolha Sem servidor sob Descobrir. A página Sem servidor é aberta.
3. Escolha a replicação com tecnologia sem servidor e escolha Excluir.
4. Para confirmar a exclusão, insira o nome da replicação com tecnologia sem servidor no campo de entrada de texto. Escolha Excluir.

Depois de excluir todas as replicações com tecnologia sem servidor, é possível excluir o perfil vinculado a serviço.

Como excluir manualmente a função vinculada a serviço usando o IAM

Use o console do IAM AWS CLI, o ou a AWS API para excluir a função AWSServiceRoleForDMSServerless vinculada ao serviço. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com perfis vinculados a serviço do AWS DMS com Tecnologia Sem Servidor

AWS DMS O Serverless oferece suporte ao uso de funções vinculadas ao serviço em todas as regiões em que o serviço está disponível.

Solução de problemas AWS Database Migration Service de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS DMS um IAM.

Tópicos

- Não estou autorizado a realizar uma ação em AWS DMS
- Não estou autorizado a realizar iam: PassRole
- Sou administrador e quero permitir que outras pessoas acessem AWS DMS
- Quero permitir que pessoas fora da minha AWS conta acessem meus AWS DMS recursos

Não estou autorizado a realizar uma ação em AWS DMS

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. O administrador é a pessoa que forneceu o seu nome de usuário e senha.

O exemplo de erro a seguir ocorre quando o usuário do `mateojackson` IAM tenta usar o console para visualizar detalhes sobre um endpoint do AWS DMS, mas não tem `dms: DescribeEndpoint` permissões.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 dms:DescribeEndpoint on resource: my-postgresql-target
```

Nesse caso, Mateo pede ao administrador para atualizar suas políticas para permitir que ele acesse endpoint `my-postgresql-target` utilizando a ação `dms:DescribeEndpoint`.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS DMS.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no AWS DMS. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Sou administrador e quero permitir que outras pessoas acessem AWS DMS

Para permitir que outras pessoas acessem AWS DMS, você deve criar uma entidade do IAM (usuário ou função) para a pessoa ou o aplicativo que precisa de acesso. Elas usarão as credenciais dessa entidade para acessar a AWS. Você deve anexar uma política à entidade que concede a eles as permissões corretas no AWS DMS.

Para começar a usar imediatamente, consulte Criar os primeiros usuário e grupo delegados pelo IAM no Guia do usuário do IAM.

Quero permitir que pessoas fora da minha AWS conta acessem meus AWS DMS recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é AWS DMS compatível com esses recursos, consulteComo AWS Database Migration Service funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Permissões do IAM necessárias para utilizar o AWS DMS

Você utiliza determinadas permissões e perfis do IAM para utilizar o AWS DMS. Se você estiver conectado como usuário do IAM e quiser usar AWS DMS, o administrador da sua conta deverá anexar a política discutida nesta seção ao usuário, grupo ou função do IAM que você usa para executar AWS DMS. Para obter mais informações sobre permissões do IAM, consulte o Guia do usuário do IAM.

A política a seguir fornece acesso e também permissões para determinadas ações necessárias de outros serviços da Amazon AWS KMS, como IAM, Amazon EC2 e Amazon. AWS DMS CloudWatch CloudWatchmonitora sua AWS DMS migração em tempo real e coleta e rastreia métricas que indicam o progresso de sua migração. Você pode usar o CloudWatch Logs para depurar problemas com uma tarefa.

> **Note**
>
> Você pode restringir ainda mais o acesso aos AWS DMS recursos usando a marcação. Para obter mais informações sobre como restringir o acesso a AWS DMS recursos usando marcação, consulte. Controle de acesso minucioso com o uso de nomes de recursos e tags

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
    ]
}
```

A discriminação das permissões a seguir pode ajudar a compreender melhor a necessidade de cada uma.

A seção a seguir é necessária para permitir que o usuário chame operações de AWS DMS API.

```
{
            "Effect": "Allow",
            "Action": "dms:*",
            "Resource": "arn:aws:dms:region:account:resourcetype/id"
}
```

A seção a seguir é necessária para permitir que o usuário liste suas AWS KMS chaves e alias disponíveis para exibição no console. Essa entrada não é necessária se você souber o Amazon Resource Name (ARN) da chave KMS e estiver usando somente o AWS Command Line Interface ().AWS CLI

```
{
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

A seção a seguir é necessária para determinados tipos de endpoints que exigem que um ARN de perfil do IAM seja transmitido com o endpoint. Além disso, se as AWS DMS funções necessárias não forem criadas com antecedência, o AWS DMS console poderá criar a função. Se todos os perfis forem configurados com antecedência, bastará ter `iam:GetRole` e `iam:PassRole`. Para obter mais informações sobre funções, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

```
{
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole",
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
```

```
        }
```

A seção a seguir é obrigatória porque AWS DMS precisa criar a instância do Amazon EC2 e configurar a rede para a instância de replicação criada. Esses recursos existem na conta do cliente, por isso, a capacidade de executar essas ações em nome do cliente é necessária.

```
{
            "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"
        }
```

A seção a seguir é necessária para permitir que o usuário possa visualizar as métricas da instância de replicação.

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Get*",
                "cloudwatch:List*"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

Esta seção é necessária para permitir que o usuário veja os logs de replicação.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:FilterLogEvents",
```

```
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:service:region:account:resourcetype/id"
        }
```

O AWS DMS console cria várias funções que são automaticamente anexadas à sua AWS conta quando você usa o AWS DMS console. Se você usa o AWS Command Line Interface (AWS CLI) ou a AWS DMS API para sua migração, você precisa adicionar essas funções à sua conta. Para obter mais informações sobre essas funções, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

Criação de funções do IAM para usar com a AWS DMS API AWS CLI e

Se você usar a API AWS CLI ou a AWS DMS API para a migração do banco de dados, deverá adicionar três funções do IAM à sua AWS conta antes de poder usar os recursos do AWS DMS. Duas delas são `dms-vpc-role` e `dms-cloudwatch-logs-role`. Se você usa o Amazon Redshift como banco de dados de destino, você também deve adicionar a função do IAM `dms-access-for-endpoint` à sua AWS conta.

As políticas gerenciadas são atualizadas automaticamente. Se estiver utilizando uma política personalizada com os perfis do IAM, verifique periodicamente se há atualizações para a política gerenciada nesta documentação. Veja os detalhes da política gerenciada utilizando uma combinação dos comandos `get-policy` e `get-policy-version`.

Por exemplo, o comando `get-policy` a seguir recupera informações sobre o perfil do IAM especificado.

```
aws iam get-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole
```

As informações retornadas do comando são as seguintes.

```
{
    "Policy": {
        "PolicyName": "AmazonDMSVPCManagementRole",
        "Description": "Provides access to manage VPC settings for AWS managed customer
 configurations",
        "CreateDate": "2015-11-18T16:33:19Z",
        "AttachmentCount": 1,
        "IsAttachable": true,
        "PolicyId": "ANPAJHKIGMBQI4AEFFSYO",
        "DefaultVersionId": "v3",
        "Path": "/service-role/",
        "Arn": "arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole",
        "UpdateDate": "2016-05-23T16:29:57Z"
    }
}
```

O comando `get-policy-version` a seguir recupera informações de políticas do IAM.

```
aws iam get-policy-version --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonDMSVPCManagementRole --version-id v3
```

As informações retornadas do comando são as seguintes.

```
{
    "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
    }
}
```

É possível utilizar os mesmos comandos para obter informações sobre `AmazonDMSCloudWatchLogsRole` e a política gerenciada `AmazonDMSRedshiftS3Role`.

> **Note**
>
> Se você usar o AWS DMS console para a migração do banco de dados, essas funções serão adicionadas à sua AWS conta automaticamente.

Os procedimentos a seguir criam os perfis do IAM `dms-vpc-role`, `dms-cloudwatch-logs-role` e `dms-access-for-endpoint`.

Para criar a função dms-vpc-role do IAM para uso com a AWS DMS API AWS CLI ou

1. Crie um arquivo JSON com a seguinte política do IAM. Nomeie o arquivo como `dmsAssumeRolePolicyDocument.json`.json.

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

   Crie a função AWS CLI usando o comando a seguir.

```
aws iam create-role --role-name dms-vpc-role --assume-role-policy-document file://
dmsAssumeRolePolicyDocument.json
```

2. Anexe a política `AmazonDMSVPCManagementRole` a `dms-vpc-role` usando o seguinte comando.

```
aws iam attach-role-policy --role-name dms-vpc-role --policy-arn
 arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole
```

Para criar a função dms-cloudwatch-logs-role do IAM para uso com a AWS DMS API AWS CLI ou

1. Crie um arquivo JSON com a seguinte política do IAM. Nomeie o arquivo como `dmsAssumeRolePolicyDocument2.json`.json.

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

   Crie a função AWS CLI usando o comando a seguir.

   ```
   aws iam create-role --role-name dms-cloudwatch-logs-role --assume-role-policy-
   document file://dmsAssumeRolePolicyDocument2.json
   ```

2. Anexe a política `AmazonDMSCloudWatchLogsRole` a `dms-cloudwatch-logs-role` usando o seguinte comando.

   ```
   aws iam attach-role-policy --role-name dms-cloudwatch-logs-role --policy-arn
    arn:aws:iam::aws:policy/service-role/AmazonDMSCloudWatchLogsRole
   ```

Se você utilizar o Amazon Redshift como o banco de dados de destino, precisará criar ao perfil do IAM `dms-access-for-endpoint` para fornecer acesso ao Amazon S3.

Para criar a função dms-access-for-endpoint do IAM para uso com o Amazon Redshift como banco de dados de destino

1. Crie um arquivo JSON com a seguinte política do IAM. Nomeie o arquivo como `dmsAssumeRolePolicyDocument3.json`.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. Crie a função AWS CLI usando o comando a seguir.

```
  aws iam create-role --role-name dms-access-for-endpoint --assume-role-policy-
document file://dmsAssumeRolePolicyDocument3.json
```

3. Anexe a política `AmazonDMSRedshiftS3Role` ao perfil `dms-access-for-endpoint` utilizando o seguinte comando.

```
aws iam attach-role-policy --role-name dms-access-for-endpoint \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSRedshiftS3Role
```

Agora você deve ter as políticas do IAM em vigor para usar a AWS DMS API AWS CLI ou.

Prevenção contra o ataque “Confused deputy” entre serviços

O problema “confused deputy” é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A imitação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado para utilizar as suas permissões para atuar nos recursos de outro cliente em que, de outra forma, ele não teria permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição `aws:SourceAccount`global `aws:SourceArn`e as chaves de contexto nas políticas de recursos para limitar as permissões que AWS Database Migration Service concedem outro serviço ao recurso. Se o valor de `aws:SourceArn` não contiver o ID da conta, como um nome de instância de replicação (ARN) do AWS DMS , utilize ambas as chaves de contexto de condição global para limitar as permissões. Se você utilizar ambas as chaves de contexto de condição global e o valor de `aws:SourceArn` contiver o ID da conta, o valor de `aws:SourceAccount` e a conta no valor de `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na mesma declaração da política. Utilize `aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

AWS DMS suporta opções confusas de delegados a partir da versão 3.4.7 e superior. Para ter mais informações, consulte AWS Notas de versão do Database Migration Service 3.4.7. Se a instância de replicação utilizar o AWS DMS versão 3.4.6 ou inferior, atualize para a versão mais recente antes de definir as opções de “confused deputy”.

A maneira mais eficiente de se proteger contra o problema “confused deputy” é utilizar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou estiver especificando vários recursos, utilize a chave de condição de contexto global `aws:SourceArn` com caracteres curingas (`*`) para as partes desconhecidas do ARN. Por exemplo, `.arn:aws:dms:*:123456789012:rep:*`

Tópicos

- Funções do IAM a serem usadas com a AWS DMS API para prevenção auxiliar confusa entre serviços

- Política do IAM para armazenar avaliações de pré-processamento no Amazon S3 para prevenção de ataques “confused deputy” entre serviços
- Usando o Amazon DynamoDB como um endpoint de destino AWS DMS com a prevenção confusa de delegações entre serviços

Funções do IAM a serem usadas com a AWS DMS API para prevenção auxiliar confusa entre serviços

Para usar a API AWS CLI ou a AWS DMS API para sua migração de banco de dados, você deve adicionar as funções `dms-vpc-role` e `dms-cloudwatch-logs-role` IAM à sua AWS conta antes de poder usar os recursos do AWS DMS. Para ter mais informações, consulte Criação de funções do IAM para usar com a AWS DMS API AWS CLI e.

O exemplo a seguir mostra as políticas para utilizar o perfil `dms-vpc-role` com a instância de replicação `my-replication-instance`. Utilize essas políticas para evitar o problema “deputy confused”.

```
{
    "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"
                }
            }
        }
    ]
}
```

Política do IAM para armazenar avaliações de pré-processamento no Amazon S3 para prevenção de ataques “confused deputy” entre serviços

Para armazenar os resultados da pré-avaliação no bucket do S3, crie uma política do IAM que permita que o AWS DMS gerencie objetos no Amazon S3. Para ter mais informações, consulte Criar recursos do IAM .

O exemplo a seguir mostra uma política de confiança com condições substitutas confusas definidas em uma função do IAM que permite AWS DMS acessar todas as tarefas e execuções de avaliação em uma conta de usuário especificada.

```
{
    "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:*"
                    ]
                }
            }
        }
    ]
}
```

Usando o Amazon DynamoDB como um endpoint de destino AWS DMS com a prevenção confusa de delegações entre serviços

Para usar o Amazon DynamoDB como um endpoint de destino para sua migração de banco de dados, você deve criar a função do IAM que AWS DMS permita assumir e conceder acesso às

tabelas do DynamoDB. Utilize este perfil ao criar o endpoint de destino do DynamoDB no AWS DMS. Para ter mais informações, consulte Utilizar o Amazon DynamoDB como destino.

O exemplo a seguir mostra uma política de confiança com condições adjuntas confusas definidas em uma função do IAM que permite que todos os AWS DMS endpoints acessem as tabelas do DynamoDB.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "dms.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "your_account_id"
                },
                "ArnLike": {
                    "AWS:SourceArn":
 "arn:aws:dms:your_region:your_account_id:endpoint:*"
                }
            }
        }
    ]
}
```

AWS políticas gerenciadas para AWS Database Migration Service

Tópicos

- AWS política gerenciada: AmazonDMSVPC ManagementRole
- AWS política gerenciada: AWSDMSServerlessServiceRolePolicy
- AWS política gerenciada: AmazonDMS CloudWatch LogsRole
- AWS política gerenciada: AWSDMSFleetAdvisorServiceRolePolicy
- AWS DMS atualizações nas políticas AWS gerenciadas

AWS política gerenciada: AmazonDMSVPC ManagementRole

Essa política está anexada à `dms-vpc-role` função, que permite AWS DMS realizar ações em seu nome.

Essa política concede permissões ao colaborador que permitem AWS DMS gerenciar recursos de rede.

Detalhes da permissão

Essa política inclui as seguintes operações:

- `ec2:CreateNetworkInterface`— AWS DMS precisa dessa permissão para criar interfaces de rede. Essas interfaces são essenciais para que a instância de AWS DMS replicação se conecte aos bancos de dados de origem e destino.
- `ec2:DescribeAvailabilityZones`— Essa permissão permite AWS DMS recuperar informações sobre as zonas de disponibilidade em uma região. AWS DMS usa essas informações para garantir que provisione recursos nas zonas corretas para redundância e disponibilidade.
- `ec2:DescribeInternetGateways`— AWS DMS pode exigir essa permissão para entender os gateways da Internet configurados na VPC. Essas informações são cruciais se a instância de replicação ou os bancos de dados precisarem de acesso à Internet.
- `ec2:DescribeSecurityGroups`— Grupos de segurança controlam o tráfego de entrada e saída para instâncias e recursos. AWS DMS precisa descrever grupos de segurança para configurar corretamente as interfaces de rede e garantir a comunicação adequada entre a instância de replicação e os bancos de dados.
- `ec2:DescribeSubnets`— Essa permissão permite AWS DMS listar as sub-redes em uma VPC. AWS DMS usa essas informações para iniciar instâncias de replicação nas sub-redes apropriadas, garantindo que elas tenham a conectividade de rede necessária.
- `ec2:DescribeVpcs`— Descrever as VPCs é essencial AWS DMS para entender o ambiente de rede em que a instância de replicação e os bancos de dados residem. Isso inclui conhecer os blocos CIDR e outras configurações específicas do VPC.
- `ec2:DeleteNetworkInterface`— AWS DMS precisa dessa permissão para limpar as interfaces de rede que criou quando elas não são mais necessárias. Isso ajuda no gerenciamento de recursos e evita custos desnecessários.
- `ec2:ModifyNetworkInterfaceAttribute`— Essa permissão é necessária AWS DMS para modificar os atributos das interfaces de rede que ela gerencia. Isso pode incluir o ajuste das configurações para garantir a conectividade e a segurança.

- `ec2:DescribeDhcpOptions`— AWS DMS recupera os detalhes do conjunto de opções DHCP para a VPC especificada. Essas informações são necessárias para configurar a rede corretamente para as instâncias de replicação.
- `ec2:DescribeNetworkInterfaces`— AWS DMS recupera informações sobre as interfaces de rede existentes na VPC. Essas informações são necessárias AWS DMS para configurar as interfaces de rede corretamente e garantir a conectividade de rede adequada para o processo de migração.

```
{
    "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 política gerenciada: AWSDMSServerlessServiceRolePolicy

Essa política está anexada à `AWSServiceRoleForDMSServerless` função, que permite AWS DMS realizar ações em seu nome. Para ter mais informações, consulte Perfil vinculado a serviço do AWS DMS com Tecnologia Sem Servidor.

Essa política concede permissões ao colaborador que permitem AWS DMS gerenciar recursos de replicação.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- dms— Permite que os diretores interajam com AWS DMS os recursos.

```
{
    "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 política gerenciada: AmazonDMS CloudWatch LogsRole

Essa política está anexada à `dms-cloudwatch-logs-role` função, que permite AWS DMS realizar ações em seu nome. Para ter mais informações, consulte Usar perfis vinculados a serviço do AWS DMS.

Essa política concede ao colaborador permissões que permitem AWS DMS publicar registros de replicação em registros. CloudWatch

Detalhes das permissões

Esta política inclui as seguintes permissões:

- `logs`— Permite que os diretores publiquem registros no CloudWatch Logs. Essa permissão é necessária para que AWS DMS possa ser usada CloudWatch para exibir registros de replicação.

```
{
    "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 política gerenciada: AWSDMSFleetAdvisorServiceRolePolicy

Você não pode se vincular AWSDMSFleetAdvisorServiceRolePolicy às suas entidades do IAM. Essa política está vinculada a uma função vinculada ao serviço que permite que o AWS DMS Fleet Advisor execute ações em seu nome. Para ter mais informações, consulte Usar perfis vinculados a serviço do AWS DMS.

Essa política concede aos colaboradores permissões que permitem que o AWS DMS Fleet Advisor publique CloudWatch métricas da Amazon.

Detalhes das permissões

Esta política inclui as seguintes permissões:

- `cloudwatch`— Permite que os diretores publiquem pontos de dados métricos na Amazon CloudWatch. Essa permissão é necessária para que o AWS DMS Fleet Advisor possa usar CloudWatch para exibir gráficos com métricas de banco de dados.

```
{
    "Version": "2012-10-17",
    "Statement": {
```

```
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "AWS/DMS/FleetAdvisor"
            }
        }
    }
}
```

AWS DMS atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas AWS DMS desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico do AWS DMS documento.

Alteração	Descrição	Data
ManagementRoleAmazonDMSVPC — Alteração	AWS DMS adicionadas `ec2:DescribeDhcpOptions` e `ec2:DescribeNetworkInterfaces` operações para AWS DMS permitir o gerenciamento das configurações de rede em seu nome.	17 de junho de 2024
AWSDMSServerlessServiceRolePolicy – Nova política	AWS DMS adicionou a `AWSDMSServerlessServiceRolePolicy` função para permitir AWS DMS a criação e o gerenciamento de serviços em seu nome, como a publicação de CloudWatch métricas da Amazon.	22 de maio de 2023

Alteração	Descrição	Data
AmazonDMS CloudWatch LogsRole — Alteração	AWS DMS adicionou o ARN para recursos sem servidor a cada uma das permissões concedidas, para permitir o upload de registros de replicação de configurações de AWS DMS replicação sem servidor para Logs. CloudWatch	22 de maio de 2023
AWSDMSFleetAdvisor ServiceRolePolicy – Nova política	AWS DMS O Fleet Advisor adicionou uma nova política para permitir a publicação de pontos de dados métricos na Amazon CloudWatch.	6 de março de 2023
AWS DMS começou a rastrear alterações	AWS DMS começou a rastrear as mudanças em suas políticas AWS gerenciadas.	6 de março de 2023

Validação de conformidade do AWS Database Migration Service

Auditores de terceiros avaliam a segurança e a conformidade do AWS Database Migration Service como parte de vários programas de conformidade da AWS. Isso inclui os seguintes programas:

- SOC
- PCI
- ISO
- FedRAMP
- DoD CC SRG
- HIPAA BAA
- MTCS
- CS
- K-ISMS
- ENS High
- OSPAR
- HITRUST CSF

Para obter uma lista de serviços da AWS no escopo de programas de conformidade específicos, consulte Serviços da AWS em escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

É possível baixar relatórios de auditoria de terceiros utilizando o AWS Artifact. Para obter mais informações, consulte Download de relatórios no AWS artifact.

A sua responsabilidade em relação à conformidade ao utilizar o AWS DMS é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias de início rápido de segurança e conformidade : esses guias de implantação apresentam considerações de arquitetura e etapas para a implantação de ambientes básicos focados na segurança e na conformidade na AWS.

- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services: este whitepaper descreve como as empresas podem utilizar a AWS para criar aplicações em conformidade com a HIPAA.
- Recursos de conformidade da AWS: essa coleção de manuais e guias pode ser válida para seu setor e local.
- AWS Config: esse serviço da AWS avalia até que ponto suas configurações de recursos atendem adequadamente às práticas internas e às diretrizes e regulamentações do setor.
- AWS Security Hub: esse serviço da AWS fornece uma visão abrangente do estado da segurança na AWS que ajuda verificar a conformidade com os padrões e as práticas recomendadas de segurança do setor.

Resiliência no AWS Database Migration Service

A infraestrutura global da AWS é criada com base em regiões da AWS e zonas de disponibilidade da AWS. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, as quais são conectadas com baixa latência, alto throughput e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Além da infraestrutura global da AWS, o AWS DMS oferece suporte de alta disponibilidade e failover para uma instância de replicação utilizando uma implantação multi-AZ quando você escolhe a opção Multi-AZ.

Em uma implantação multi-AZ, o AWS DMS provisiona e mantém automaticamente uma réplica em espera da instância de replicação em outra zona de disponibilidade. A instância de replicação primária é replicada em sincronia para a réplica em espera. Se a instância de replicação primária falhar ou parar de responder, a instância em espera retomará qualquer tarefa em execução com o mínimo de interrupção. Como a primária está replicando constantemente seu estado para a de espera, a implantação multi-AZ implica alguma sobrecarga no desempenho.

Para obter mais informações sobre como trabalhar com implantações multi-AZ, consulte Trabalhando com uma instância de AWS DMS replicação.

Segurança da infraestrutura no AWS Database Migration Service

Por ser um serviço gerenciado, o AWS Database Migration Service é protegido pela segurança da rede global da AWS. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte Segurança na Nuvem AWS. Para projetar o ambiente da AWS utilizando as práticas recomendadas de segurança de infraestrutura, consulte Proteção de infraestrutura em Pilar segurança: AWS Well-Architected Framework.

Você usa chamadas de API publicadas pela AWS para acessar o AWS DMS por meio da rede. Os clientes devem oferecer suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível utilizar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

É possível chamar essas operações de API de qualquer local da rede. O AWS DMS também é compatível com as políticas de acesso baseadas em recursos, que podem especificar restrições sobre ações e recursos, por exemplo, com base no endereço IP de origem. Além disso, é possível utilizar políticas do AWS DMS para controlar o acesso de endpoints específicos da Amazon VPC ou específicos de nuvens privadas virtuais (VPCs). Efetivamente, isso isola o acesso à rede para um determinado recurso do AWS DMS somente da VPC específica dentro da rede da AWS. Para obter mais informações sobre como utilizar políticas de acesso baseadas em recursos com o AWS DMS, incluindo exemplos, consulte Controle de acesso minucioso com o uso de nomes de recursos e tags.

Para confinar a comunicação com o AWS DMS em uma única VPC, é possível criar um endpoint de interface da VPC que permita a conexão com o AWS DMS por meio do AWS PrivateLink. O AWS PrivateLink ajuda a garantir que qualquer chamada para o AWS DMS e os resultados associados permaneçam confinados à VPC específica para a qual o endpoint de interface foi criado. É possível especificar o URL desse endpoint de interface como uma opção em que cada comando do AWS DMS executado utiliza a AWS CLI ou um SDK. Isso ajuda a garantir que toda a comunicação com o AWS DMS permaneça confinada à VPC e, de outra forma, seja invisível para a internet pública.

Como criar um endpoint de interface para acessar o DMS em uma única VPC

1. Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, escolha Endpoints. Isso abre a página Criar endpoints, na qual é possível criar o endpoint da interface de uma VPC para o AWS DMS.
3. Escolha Serviços daAWS e pesquise e escolha um valor para Nome do serviço, nesse caso, AWS DMS, no formulário a seguir.

   ```
   com.amazonaws.region.dms
   ```

   Aqui, *region* especifica a região da AWS em que o AWS DMS é executado, por exemplo, `com.amazonaws.us-west-2.dms`.
4. Em VPC, escolha a VPC na qual criar o endpoint de interface, por exemplo, `vpc-12abcd34`.
5. Escolha um valor para a Zona de disponibilidade e para o ID de sub-rede. Esses valores devem indicar um local em que o endpoint do AWS DMS escolhido pode ser executado, por exemplo, `us-west-2a (usw2-az1)` e `subnet-ab123cd4`.
6. Escolha Habilitar nome DNS para criar o endpoint com um nome DNS. Esse nome DNS consiste no ID do endpoint (`vpce-12abcd34efg567hij`) hifenizado com uma string aleatória (`ab12dc34`). Eles são separados do nome do serviço por um ponto na ordem inversa, separados por pontos, com `vpce` adicionado (`dms.us-west-2.vpce.amazonaws.com`).

   Um exemplo é `vpce-12abcd34efg567hij-ab12dc34.dms.us-west-2.vpce.amazonaws.com`.
7. Em Grupo de segurança, escolha um grupo a ser utilizado para o endpoint.

   Ao configurar o grupo de segurança, permita chamadas HTTPS de saída neles. Para obter mais informações, consulte Criar grupos de segurança no Guia do usuário da Amazon VPC.
8. Escolha Acesso total ou um valor personalizado para Política. Por exemplo, é possível escolher uma política personalizada semelhante à seguinte que restringe o acesso do endpoint a determinadas ações e recursos.

   ```
   {
     "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": "*"
    }
  ]
}
```

Aqui, o exemplo de política permite qualquer chamada da API do AWS DMS, exceto para excluir ou modificar uma instância de replicação específica.

Agora é possível especificar um URL formado utilizando o nome DNS criado na etapa 6 como uma opção. Especifique isso para cada comando da CLI do AWS DMS ou operação da API para acessar a instância de serviço utilizando o endpoint de interface criado. Por exemplo, é possível executar o comando `DescribeEndpoints` da CLI do DMS nessa VPC, conforme mostrado a seguir.

```
$ aws dms describe-endpoints --endpoint-url https://vpce-12abcd34efg567hij-
ab12dc34.dms.us-west-2.vpce.amazonaws.com
```

Se você ativar a opção de DNS privado, não será necessário especificar o URL do endpoint na solicitação.

Para obter mais informações sobre como criar e utilizar endpoints de interface da VPC (incluindo ativar a opção DNS privado), consulte Endpoints de interface da VPC (PrivateLink da AWS) no Guia do usuário da Amazon VPC.

Controle de acesso minucioso com o uso de nomes de recursos e tags

Você pode usar nomes e tags de recursos com base nos Amazon Resource Names (ARNs) para gerenciar o acesso aos AWS DMS recursos. Para fazer isso, defina a ação permitida ou inclua instruções condicionais em políticas do IAM.

Usar nomes de recursos para controle de acesso

É possível criar uma conta de usuário do IAM e atribuir uma política com base no ARN do recurso do AWS DMS .

A política a seguir nega acesso à instância de AWS DMS replicação com o ARN arn:aws:dms:us-east- 1:152683116:rep:doh67ztoxglixmihkiTV:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:dms:us-east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV"
        }
    ]
}
```

Por exemplo, os comandos a seguir poderão falhar quando a política estiver em vigor:

```
$ aws dms delete-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV"

A client error (AccessDeniedException) occurred when calling the
 DeleteReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
```

```
dms:DeleteReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV

$ aws dms modify-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV"

A client error (AccessDeniedException) occurred when calling the
 ModifyReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:ModifyReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:DOH67ZTOXGLIXMIHKITV
```

Você também pode especificar políticas do IAM que limitam o acesso aos AWS DMS endpoints e às tarefas de replicação.

A política a seguir limita o acesso a um AWS DMS endpoint usando o ARN do endpoint.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:dms:us-
east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX"
        }
    ]
}
```

Por exemplo, os comandos a seguir poderão falhar quando a política que utiliza o ARN do endpoint estiver em vigor:

```
$ aws dms delete-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX"
```

```
A client error (AccessDeniedException) occurred when calling the DeleteEndpoint
 operation:
User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
 dms:DeleteEndpoint
on resource: arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX

$ aws dms modify-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX"


A client error (AccessDeniedException) occurred when calling the ModifyEndpoint
 operation:
User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
 dms:ModifyEndpoint
on resource: arn:aws:dms:us-east-1:152683116:endpoint:D6E37YBXTNHOA6XRQSZCUGX
```

A política a seguir limita o acesso a uma AWS DMS tarefa usando o ARN da tarefa.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:dms:us-east-1:152683116:task:UO3YR4N47DXH3ATT4YMWOIT"
        }
    ]
}
```

Por exemplo, os comandos a seguir poderão falhar quando a política que utiliza o ARN da tarefa estiver em vigor.

```
$ aws dms delete-replication-task
   --replication-task-arn "arn:aws:dms:us-
east-1:152683116:task:UO3YR4N47DXH3ATT4YMWOIT"
```

```
A client error (AccessDeniedException) occurred when calling the DeleteReplicationTask
 operation:
User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
 dms:DeleteReplicationTask
on resource: arn:aws:dms:us-east-1:152683116:task:UO3YR4N47DXH3ATT4YMWOIT
```

Uso de tags para controlar o acesso

AWS DMS define um conjunto de pares de valores-chave comuns que estão disponíveis para uso em políticas definidas pelo cliente sem nenhum requisito adicional de marcação. Para obter mais informações sobre a marcação de AWS DMS recursos, consulteMarcar recursos no AWS Database Migration Service.

A seguir, são listadas as tags padrão disponíveis para uso com AWS DMS:

- aws: CurrentTime — Representa a data e a hora da solicitação, permitindo a restrição de acesso com base em critérios temporais.
- aws: EpochTime — Essa tag é semelhante à CurrentTime tag aws: anterior, exceto que a hora atual é representada como o número de segundos decorridos desde a época do Unix.
- aws: MultiFactorAuthPresent — Essa é uma tag booleana que indica se a solicitação foi assinada por meio de autenticação multifatorial.
- aws: MultiFactorAuthAge — Fornece acesso à idade do token de autenticação multifatorial (em segundos).
- aws:principaltype: concede acesso ao tipo de entidade principal (usuário, conta, usuário federado etc.) da solicitação atual.
- aws: SourceIp — Representa o endereço IP de origem do usuário que está emitindo a solicitação.
- aws: UserAgent — Fornece informações sobre o aplicativo cliente solicitando um recurso.
- aws:userid: fornece acesso ao ID do usuário emissor da solicitação.
- aws:username: fornece acesso ao nome do usuário emissor da solicitação.
- dms: InstanceClass — Fornece acesso ao tamanho computacional do (s) host (s) da instância de replicação.
- dms: StorageSize — Fornece acesso ao tamanho do volume de armazenamento (em GB).

Você também pode definir suas próprias tags. As tags definidas pelo cliente são pares simples de valores-chave que persistem no serviço de marcação. AWS Você pode adicioná-las a recursos do

AWS DMS , incluindo instâncias de replicação, endpoints e tarefas. Essas tags são correspondidas utilizando instruções "Condicionais" do IAM em políticas, e são referenciadas utilizando uma tag condicional específica. As chaves das tags têm o prefixo "dms", o tipo de recurso e o prefixo "tag". Veja o formato da tag a seguir.

```
dms:{resource type}-tag/{tag key}={tag value}
```

Por exemplo, suponha que você queira definir uma política que permita que apenas uma chamada à API tenha êxito em uma instância de replicação que contém a tag "stage=production". A instrução·condicional a seguir corresponde a um recurso com a tag especificada.

```
"Condition":
{
    "streq":
        {
            "dms:rep-tag/stage":"production"
        }
}
```

Você adiciona a seguinte tag a uma instância de replicação que corresponde a essa condição de política.

```
stage production
```

Além das tags já atribuídas aos AWS DMS recursos, as políticas também podem ser escritas para limitar as chaves e os valores das tags que podem ser aplicados a um determinado recurso. Nesse caso, o prefixo da tag é "req".

Por exemplo, a seguinte instrução de política limita as tags que um usuário pode atribuir a determinado recurso a uma lista específica de valores permitidos.

```
 "Condition":
{
    "streq":
        {
            "dms:rep-tag/stage": [ "production", "development", "testing" ]
        }
}
```

Os exemplos de políticas a seguir limitam o acesso a um AWS DMS recurso com base em tags de recursos.

A seguinte política limita o acesso a uma instância de replicação em que o valor da tag é "Desktop", e a chave de tags é "Env":

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:rep-tag/Env": [
                        "Desktop"
                    ]
                }
            }
        }
    ]
}
```

Os comandos a seguir podem ter êxito ou falha de acordo com a política do IAM que restringe o acesso quando o valor da tag é "Desktop", e a chave da tag é "Env".

```
$ aws dms list-tags-for-resource
   --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
   --endpoint-url http://localhost:8000
{
    "TagList": [
        {
            "Value": "Desktop",
            "Key": "Env"
        }
    ]
```

```
}

$ aws dms delete-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN"
A client error (AccessDeniedException) occurred when calling the
 DeleteReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:DeleteReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN

$ aws dms modify-replication-instance
   --replication-instance-arn "arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN"

A client error (AccessDeniedException) occurred when calling the
 ModifyReplicationInstance
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:ModifyReplicationInstance on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN

$ aws dms add-tags-to-resource
   --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
   --tags Key=CostCenter,Value=1234

A client error (AccessDeniedException) occurred when calling the AddTagsToResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:AddTagsToResource on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN

$ aws dms remove-tags-from-resource
   --resource-name arn:aws:dms:us-east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
   --tag-keys Env

A client error (AccessDeniedException) occurred when calling the
 RemoveTagsFromResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:RemoveTagsFromResource on resource: arn:aws:dms:us-
east-1:152683116:rep:46DHOU7JOJYOJXWDOZNFEN
```

A política a seguir limita o acesso a um AWS DMS endpoint em que o valor da tag é “Desktop” e a chave da tag é “Env”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:endpoint-tag/Env": [
                        "Desktop"
                    ]
                }
            }
        }
    ]
}
```

Os comandos a seguir podem ter êxito ou falha de acordo com a política do IAM que restringe o acesso quando o valor da tag é "Desktop", e a chave da tag é "Env".

```
$ aws dms list-tags-for-resource
   --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
{
    "TagList": [
        {
            "Value": "Desktop",
            "Key": "Env"
        }
    ]
}

$ aws dms delete-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I"

A client error (AccessDeniedException) occurred when calling the DeleteEndpoint
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
```

```
dms:DeleteEndpoint on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I

$ aws dms modify-endpoint
   --endpoint-arn "arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I"


A client error (AccessDeniedException) occurred when calling the ModifyEndpoint
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:ModifyEndpoint on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I

$ aws dms add-tags-to-resource
   --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
   --tags Key=CostCenter,Value=1234

A client error (AccessDeniedException) occurred when calling the AddTagsToResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:AddTagsToResource on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I

$ aws dms remove-tags-from-resource
   --resource-name arn:aws:dms:us-east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
   --tag-keys Env

A client error (AccessDeniedException) occurred when calling the
 RemoveTagsFromResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:RemoveTagsFromResource on resource: arn:aws:dms:us-
east-1:152683116:endpoint:J2YCZPNGOLFY52344IZWA6I
```

A política a seguir limita o acesso a uma tarefa de replicação em que o valor da tag é "Desktop", e a chave da tag é "Env".

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dms:*"
            ],
```

```
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "dms:task-tag/Env": [
                        "Desktop"
                    ]
                }
            }
        }
    ]
}
```

Os comandos a seguir podem ter êxito ou falha de acordo com a política do IAM que restringe o acesso quando o valor da tag é "Desktop", e a chave da tag é "Env".

```
$ aws dms list-tags-for-resource
   --resource-name arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
{
    "TagList": [
        {
            "Value": "Desktop",
            "Key": "Env"
        }
    ]
}

$ aws dms delete-replication-task
   --replication-task-arn "arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3"

A client error (AccessDeniedException) occurred when calling the DeleteReplicationTask
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:DeleteReplicationTask on resource: arn:aws:dms:us-
east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3

$ aws dms add-tags-to-resource
   --resource-name arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
   --tags Key=CostCenter,Value=1234

A client error (AccessDeniedException) occurred when calling the AddTagsToResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
```

```
dms:AddTagsToResource on resource: arn:aws:dms:us-
east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3

$ aws dms remove-tags-from-resource
   --resource-name arn:aws:dms:us-east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
   --tag-keys Env

A client error (AccessDeniedException) occurred when calling the
 RemoveTagsFromResource
operation: User: arn:aws:iam::152683116:user/dmstestusr is not authorized to perform:
dms:RemoveTagsFromResource on resource: arn:aws:dms:us-
east-1:152683116:task:RB7N24J2XBUPS3RFABZTG3
```

Configurando uma chave de criptografia e especificando permissões AWS KMS

AWS DMS criptografa o armazenamento usado por uma instância de replicação e as informações de conexão do endpoint. Para criptografar o armazenamento usado por uma instância de replicação, AWS DMS use uma chave AWS Key Management Service (AWS KMS) exclusiva da sua AWS conta. Você pode visualizar e gerenciar essa chave com AWS KMS. Você pode usar a chave padrão do KMS na sua conta (`aws/dms`) ou criar uma chave personalizada do KMS. Se você tiver uma chave de criptografia existente do KMS, também será possível usá-la para criptografar.

> **Note**
>
> Qualquer AWS KMS chave personalizada ou existente que você usa como chave de criptografia deve ser uma chave simétrica. AWS DMS não suporta o uso de chaves de criptografia assimétricas. Para obter mais informações sobre as chaves de criptografia simétricas e assimétricas, consulte https://docs.aws.amazon.com/kms/latest/developerguide/symmetric-asymmetric.html no Guia do desenvolvedor do AWS Key Management Service .

A chave padrão do KMS (`aws/dms`) será criada ao executar uma instância de replicação pela primeira vez, se você não tiver selecionado uma chave personalizada do KMS na seção Avançado, na página Criar instância de replicação. Se você utilizar a chave padrão do KMS, as únicas permissões que deverão ser concedidas à conta de usuário do IAM que você está utilizando para a migração são `kms:ListAliases` e `kms:DescribeKey`. Para obter mais informações sobre o uso da chave padrão do KMS, consulte Permissões do IAM necessárias para utilizar o AWS DMS.

Para usar uma chave personalizada do KMS, atribua permissões a ela utilizando uma das seguintes opções:

- Adicione a conta de usuário do IAM usada para a migração como administrador da chave ou usuário da chave AWS KMS personalizada. Isso garantirá que as permissões necessárias do AWS KMS sejam concedidas à conta do usuário do IAM. Essa ação é uma adição às permissões do IAM que você concede à conta do usuário do IAM para usar o AWS DMS. Para obter mais informações sobre como conceder permissões a um usuário de chaves, consulte Permite que os usuários de chaves usem a chave do KMS no Guia do desenvolvedor do AWS Key Management Service .

- Se você não quiser adicionar a conta do usuário do IAM como um administrador de chaves ou usuário de chaves para a chave personalizada do KMS, adicione as seguintes permissões adicionais às permissões do IAM que devem ser concedidas à conta do usuário do IAM para usar o AWS DMS.

```
{
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:Encrypt",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        },
```

AWS DMS também funciona com aliases de chave KMS. Para obter mais informações sobre como criar suas próprias chaves do AWS KMS e conceder acesso aos usuários a uma chave do KMS, consulte o Guia do desenvolvedor do AWS KMS.

Se você não especificar um identificador de chave KMS, AWS DMS usará sua chave de criptografia padrão. AWS KMS cria a chave de criptografia padrão AWS DMS para sua AWS conta. Sua AWS conta tem uma chave de criptografia padrão diferente para cada AWS região.

Para gerenciar as AWS KMS chaves usadas para criptografar seus AWS DMS recursos, use o. AWS Key Management Service AWS KMS combina hardware e software seguros e de alta disponibilidade para fornecer um sistema de gerenciamento de chaves dimensionado para a nuvem. Usando AWS KMS, você pode criar chaves de criptografia e definir as políticas que controlam como essas chaves podem ser usadas.

Você pode encontrar AWS KMS no AWS Management Console

1. Faça login no console AWS Management Console e abra o AWS Key Management Service (AWS KMS) em https://console.aws.amazon.com/kms.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. Escolha uma das seguintes opções para trabalhar com AWS KMS teclas:

- Para ver as chaves em sua conta que AWS cria e gerencia para você, no painel de navegação, escolha chaves AWS gerenciadas.
- Para exibir as chaves em sua conta que você cria e gerencia, no painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente).

AWS KMS suporta AWS CloudTrail, para que você possa auditar o uso das chaves para verificar se as chaves estão sendo usadas adequadamente. Suas AWS KMS chaves podem ser usadas em combinação com AWS DMS AWS serviços compatíveis, como Amazon RDS, Amazon S3, Amazon Redshift e Amazon EBS.

Você também pode criar AWS KMS chaves personalizadas especificamente para criptografar dados de destino para os seguintes AWS DMS endpoints:

- Amazon Redshift: para obter mais informações, consulte Criar e utilizar as chaves do AWS KMS para criptografar os dados de destino do Amazon Redshift.
- Amazon S3: para obter mais informações, consulte Criar chaves do AWS KMS para criptografar objetos de destino do Amazon S3.

Depois de criar seus AWS DMS recursos com uma chave KMS, você não poderá alterar a chave de criptografia desses recursos. Certifique-se de determinar seus requisitos de chave de criptografia antes de criar seus AWS DMS recursos.

Segurança de rede para AWS Database Migration Service

Os requisitos de segurança para a rede que você cria ao usar AWS Database Migration Service dependem de como você configura a rede. As regras gerais de segurança de rede para AWS DMS são as seguintes:

- A instância de replicação deve ter acesso aos endpoints de origem e de destino. O security group da instância de replicação deve ter Network ACL ou regras que permitam a saída da instância da porta do banco de dados para os endpoints do banco de dados.
- Os endpoints de banco de dados devem incluir regras de Network ACLs e security group que permitam o acesso de entrada a partir da instância de replicação. Isso pode ser feito utilizando o grupo de segurança da instância de replicação, o endereço IP privado, o endereço IP público ou o endereço público do gateway NAT, dependendo da sua configuração.
- Se a rede utilizar um túnel de VPN, a instância do Amazon EC2 que atua como o gateway NAT deve utilizar um grupo de segurança que tenha regras que permitam que a instância de replicação envie tráfego por meio dele.

Por padrão, o security group da VPC usado pela instância de AWS DMS replicação tem regras que permitem a saída para 0.0.0.0/0 em todas as portas. Se você modificá-lo ou usar o seu próprio, a saída deverá, no mínimo, ser permitida para os endpoints de origem e de destino nas respectivas portas de banco de dados.

As configurações de rede que podem ser usadas para migrar bancos de dados exigem considerações específicas de segurança:

- Configuração com todos os componentes de migração de banco de dados em uma VPC: o grupo de segurança utilizado pelos endpoints deve permitir a entrada na porta do banco de dados da instância de replicação. Confirme se o security group usado pela instância de replicação tem entrada para os endpoints ou crie uma regra no security group usado pelos endpoints que permita ao endereço IP privado acessar a instância de replicação.
- Configuração com várias VPCs: o grupo de segurança utilizado pela instância de replicação deve ter uma regra para o intervalo da VPC e a porta do banco de dados no banco de dados.
- Configuração de uma rede para uma VPC usando AWS Direct Connect ou uma VPN: um túnel da VPN que permite o tráfego para o túnel da VPC em uma VPN on-premises. Nessa configuração, o VPC inclui uma regra de roteamento que envia o tráfego destinado a um endereço IP ou intervalo específico para um host que pode conectar o tráfego do VPC com a VPN local. Nesse caso, o host

NAT inclui suas próprias configurações de grupo de segurança, que devem permitir o tráfego do endereço IP privado da instância de replicação ou do grupo de segurança para a instância NAT.

- Configuração de uma rede para uma VPC utilizando a internet: o grupo de segurança da VPC deve incluir regras de roteamento que enviem o tráfego não destinado à VPC para o gateway da Internet. Nessa configuração, a conexão ao endpoint parecerá vir do endereço IP público na instância de replicação.
- Configuração com uma instância de banco de dados RDS fora de uma VPC para uma instância de banco de dados em uma VPC usando ClassicLink— Quando a instância de banco de dados Amazon RDS de origem ou de destino não está em uma VPC e não compartilha um grupo de segurança com a VPC em que a instância de replicação está localizada, você pode configurar um servidor proxy e ClassicLink usá-lo para conectar os bancos de dados de origem e de destino.
- O endpoint de origem está fora da VPC utilizada pela instância de replicação e utiliza um gateway NAT: é possível configurar um gateway de conversão de endereços de rede (NAT) utilizando um único endereço IP elástico associado a uma única interface de rede elástica. Essa interface de rede elástica recebe um identificador NAT (nat-#####). Se a VPC incluir uma rota padrão para o gateway NAT em vez do gateway da internet, a instância de replicação parecerá entrar em contato com o endpoint do banco de dados utilizando o endereço IP público do gateway da internet. Nesse caso, a entrada no endpoint do banco de dados fora da VPC precisa permitir a entrada do endereço NAT em vez do endereço IP público da instância de replicação.
- Endpoints da VPC para mecanismos não RDBMS: o AWS DMS não é compatível com endpoints da VPC para mecanismos não RDBMS.

Usando SSL com AWS Database Migration Service

É possível criptografar conexões para endpoints de origem e de destino utilizando Secure Sockets Layer (SSL). Para fazer isso, você pode usar o AWS DMS Management Console ou a AWS DMS API para atribuir um certificado a um endpoint. Você também pode usar o AWS DMS console para gerenciar seus certificados.

Nem todos os bancos de dados utilizam o SSL da mesma forma. A edição compatível com o MySQL do Amazon Aurora utiliza o nome do servidor e o endpoint da instância primária no cluster, como o endpoint para SSL. Um endpoint do Amazon Redshift já utiliza uma conexão SSL e não requer uma configuração de conexão SSL feita pelo AWS DMS. Um endpoint do Oracle exige etapas adicionais. Para obter mais informações, consulte Suporte de SSL para um endpoint do Oracle.

Tópicos

- Limitações ao uso de SSL com o AWS DMS
- Gerenciar certificados
- Ativar SSL para um endpoint de SQL Server ou PostgreSQL compatível com MySQL

Para atribuir um certificado a um endpoint, forneça o certificado raiz ou a cadeia de certificados de CA intermediários que levam à raiz (como um pacote de certificados), que foi usada para cadastrar o certificado SSL do servidor implementado ao seu endpoint. Os certificados só são aceitos como arquivos X509 com formato PEM. Ao importar um certificado, você recebe o Nome de recurso da Amazon (ARN), que pode ser usado para especificar o certificado para um endpoint. Se você utilizar o Amazon RDS, poderá baixar a CA raiz e o pacote de certificados fornecidos no arquivo `rds-combined-ca-bundle.pem` hospedado pelo Amazon RDS. Para obter mais informações sobre como baixar esse arquivo, consulte Utilizar o SSL/TLS para criptografar uma conexão com uma instância de banco de dados no Guia do usuário do Amazon RDS.

É possível escolher dentre vários modos SSL para usar na verificação de certificado SSL.

- none: a conexão não é criptografada. Essa opção não é segura, mas exige menos custos indiretos.
- require: a conexão é criptografada utilizando o SSL (TLS), mas nenhuma verificação da CA é feita. Esta opção é mais segura e exige mais custos indiretos.
- verify-ca: a conexão é criptografada. Esta opção é mais segura e exige mais custos indiretos. Esta opção verifica o certificado do servidor.

- verify-full: a conexão é criptografada. Esta opção é mais segura e exige mais custos indiretos. Esta opção verifica o certificado do servidor e se o hostname do servidor corresponde ao hostname do atributo do certificado.

Nem todos os modos SSL funcionam com todos os endpoints de banco de dados. A tabela a seguir mostra os modos SSL que são suportados por cada mecanismo de banco de dados.

Mecanismo de banco de dados	none	require	verify-ca	verify-full
MySQL/MariaDB/ Amazon Aurora MySQL	Padrão	Não compatível	Compatível	Compatível
Microsoft SQL Server	Padrão	Compatível	Sem suporte	Compatível
PostgreSQL	Padrão	Compatível	Compatível	Compatível
Amazon Redshift	Padrão	SSL não ativado	SSL não ativado	SSL não ativado
Oracle	Padrão	Não compatível	Compatível	Sem suporte
SAP ASE	Padrão	SSL não ativado	SSL não ativado	Compatível
MongoDB	Padrão	Compatível	Sem suporte	Compatível
Db2 LUW	Padrão	Sem suporte	Compatível	Sem suporte
Db2 for z/OS	Padrão	Sem suporte	Compatível	Sem suporte

> **Note**
>
> A opção Modo SSL no console ou na API do DMS não se aplica a alguns serviços de streaming de dados e de NoSQL, como o Kinesis e o DynamoDB. Eles são seguros por padrão, portanto, o DMS mostra que a configuração do modo SSL é igual a nenhum (Modo SSL= nenhum). Não é necessário fornecer nenhuma configuração adicional para que o

endpoint utilize o SSL. Por exemplo, ao utilizar o Kinesis como um endpoint de destino, ele é seguro por padrão. Todas as chamadas de API para o Kinesis usam SSL, portanto, não há necessidade de uma opção adicional de SSL no endpoint do DMS. É possível colocar e recuperar dados com segurança por meio de endpoints do SSL utilizando o protocolo HTTPS, que o DMS utiliza por padrão ao se conectar a um fluxo de dados Kinesis.

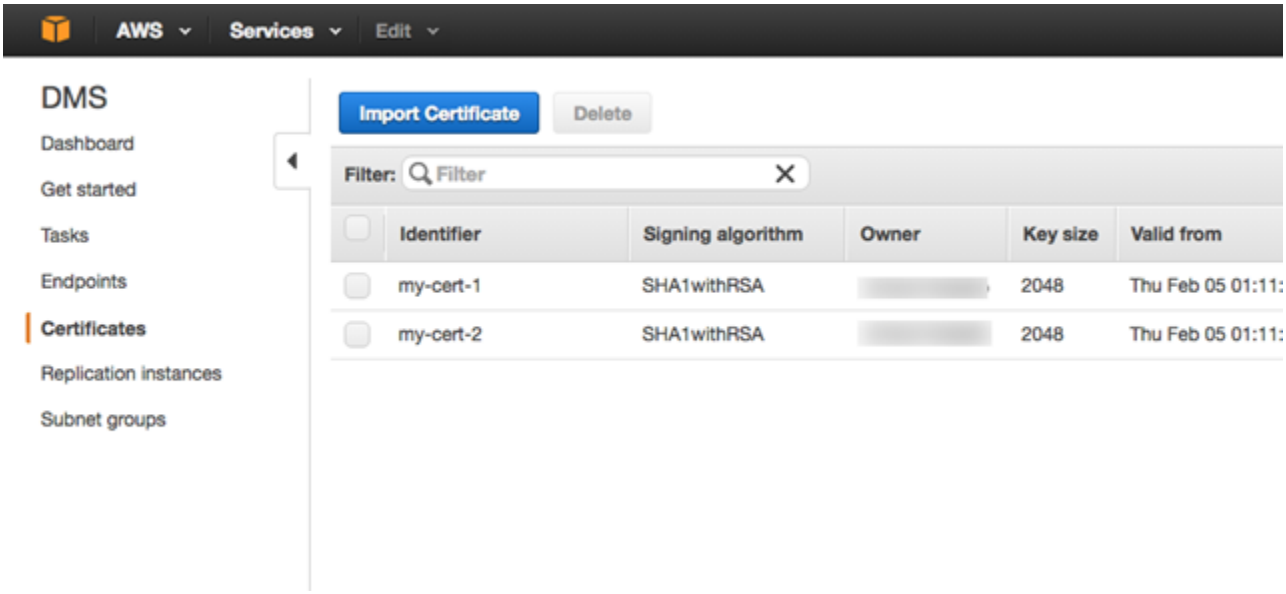
Limitações ao uso de SSL com o AWS DMS

A seguir estão as limitações do uso de SSL com AWS DMS:

- Não há suporte para conexões SSL com endpoints de destino do Amazon Redshift. AWS DMS usa um bucket do Amazon S3 para transferir dados para o banco de dados do Amazon Redshift. Por padrão, essa transmissão é criptografada pelo Amazon Redshift.
- Os limites de tempo do SQL podem ocorrer enquanto tarefas de CDC são executadas com endpoints do Oracle com SSL ativado. Se você tiver esse problema, em que contadores de CDC não refletem os números esperados, defina o parâmetro `MinimumTransactionSize` da seção `ChangeProcessingTuning` das configurações de tarefa, como um valor menor. É possível começar com um valor baixo como 100. Para obter mais informações sobre o parâmetro `MinimumTransactionSize`, consulte Configurações de ajuste de processamento de alterações.
- Você pode importar certificados somente nos formatos .pem e .sso (Oracle wallet).
- Em alguns casos, o certificado SSL do servidor pode ser assinado por uma autoridade de certificação (CA) intermediária. Em caso afirmativo, verifique se toda a cadeia de certificados da CA intermediária até a CA raiz é importada como um único arquivo .pem.
- Se você estiver utilizando certificados autoassinados no servidor, escolha require como modo SSL. O modo SSL require confia implicitamente no certificado SSL do servidor e não tentará validar se o certificado foi assinado por uma CA.

Gerenciar certificados

É possível usar o console do DMS para visualizar e gerenciar certificados SSL. Você também pode importar os certificados utilizando o console do DMS.



Ativar SSL para um endpoint de SQL Server ou PostgreSQL compatível com MySQL

É possível adicionar uma conexão SSL a um endpoint recém-criado ou existente.

Para criar um AWS DMS endpoint com SSL

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Se você estiver conectado como usuário AWS Identity and Access Management (IAM), verifique se você tem as permissões apropriadas para acessar AWS DMS. Para obter mais informações sobre as permissões necessárias para a migração de banco de dados, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, escolha Certificates.
3. Selecione Import Certificate.
4. Carregue o certificado que deseja usar para criptografar a conexão a um endpoint.

> **Note**
>
> Você também pode carregar um certificado usando o AWS DMS console ao criar ou modificar um endpoint selecionando Adicionar novo certificado CA na página Criar endpoint de banco de dados.

Para o Aurora Sem Servidor como destino, obtenha o certificado mencionado em Utilizar TLS/SSL com o Aurora Sem Servidor.

5. Crie um endpoint conforme descrito em Etapa 2: Especificar endpoints de origem e de destino

Para modificar um AWS DMS endpoint existente para usar SSL

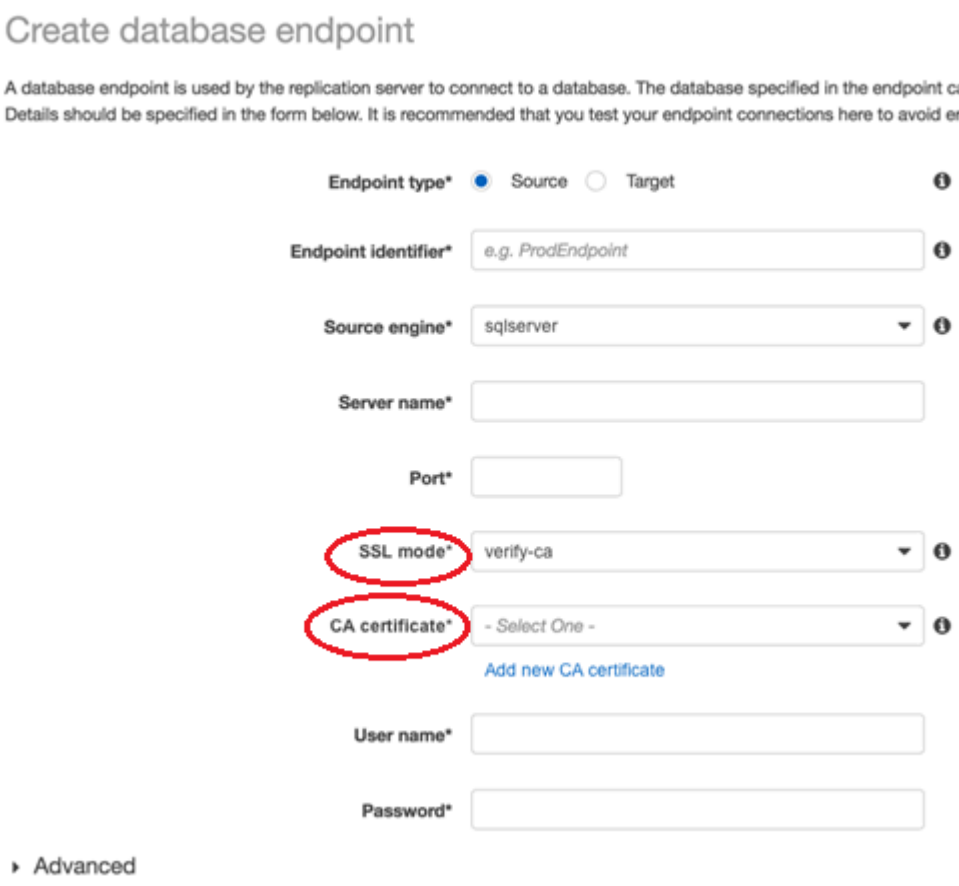
1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Se estiver conectado como um usuário do IAM, verifique se você possui as permissões necessárias para acessar o AWS DMS. Para obter mais informações sobre as permissões necessárias para a migração de banco de dados, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, escolha Certificates.
3. Selecione Import Certificate.
4. Carregue o certificado que deseja usar para criptografar a conexão a um endpoint.

   Note

   Você também pode carregar um certificado usando o AWS DMS console ao criar ou modificar um endpoint selecionando Adicionar novo certificado CA na página Criar endpoint de banco de dados.
5. No painel de navegação, selecione Endpoints, escolha o endpoint que você deseja modificar e selecione Modify.
6. Escolha um valor para o SSL mode (Modo SSL).

   Se você escolher o modo verify-ca ou verify-full, especifique o certificado que você deseja usar para CA certificate (Certificado CA), conforme mostrado a seguir.



7. Escolha Modificar.
8. Quando o endpoint tiver sido modificado, selecione-o e escolha Test connection (Testar conexão) para determinar se a conexão SSL está funcionando.

Após criar os endpoints de origem e de destino, crie uma tarefa que os use. Para obter mais informações sobre como criar uma tarefa, consulte Etapa 3: Criar uma tarefa e migrar os dados.

Alterar a senha do banco de dados

Na maioria das situações, alterar a senha do banco de dados do endpoint de origem ou de destino é simples. Se você precisar alterar a senha do banco de dados de um endpoint que está usando atualmente em uma tarefa de migração ou replicação, o processo precisará de algumas etapas adicionais. O procedimento a seguir mostra como fazer isso.

Como alterar a senha do banco de dados de um endpoint em uma tarefa de migração ou de replicação

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.

   Se estiver conectado como um usuário do IAM, verifique se você possui as permissões necessárias para acessar o AWS DMS. Para obter mais informações sobre as permissões necessárias, consulte Permissões do IAM necessárias para utilizar o AWS DMS.
2. No painel de navegação, escolha Tarefas de migração de banco de dados.
3. Escolha a tarefa que utiliza o endpoint cuja senha do banco de dados você deseja alterar e selecione Stop.
4. Enquanto a tarefa está interrompida, é possível alterar a senha do banco de dados do endpoint utilizando as ferramentas nativas usadas para trabalhar com o banco de dados.
5. Retorne ao DMS Management Console e selecione Endpoints no painel de navegação.
6. Escolha o endpoint do banco de dados cuja senha você modificou e selecione Modify.
7. Digite a nova senha na caixa Senha e selecione Modificar.
8. Escolha Tarefas de migração de banco de dados no painel de navegação.
9. Escolha a tarefa interrompida anteriormente e selecione Iniciar/retomar.
10. Selecione Reiniciar ou Retomar, dependendo de como você deseja continuar a tarefa, e escolha Iniciar tarefa.

Cotas para o AWS Database Migration Service

Veja a seguir as cotas de recursos e as restrições de nomenclatura do AWS Database Migration Service (AWS DMS).

O tamanho máximo de um banco de dados que o AWS DMS pode migrar depende de vários fatores. Do tamanho máximo do ambiente de origem, da distribuição dos dados no banco de dados de origem e de quanto o sistema está ocupado.

A melhor forma de determinar se o seu sistema específico é um candidato para o AWS DMS é testá-lo. Inicie lentamente para que seja possível estabelecer a configuração e adicione alguns objetos complexos. Por fim, tente uma carga máxima como teste.

Cotas de recursos para o AWS Database Migration Service

Cada conta da AWS tem cotas para cada região da AWS, quanto ao número de recursos do AWS DMS que podem ser criados. Depois que a cota de um recurso é atingida, as chamadas adicionais para criá-lo falham, com uma exceção.

A tabela a seguir lista os recursos do AWS DMS e suas cotas por região da AWS.

Recurso	Cota padrão
Controle de utilização de solicitações da API	Máximo de 200 solicitações por segundo
Taxa de atualização de solicitações de API	8 solicitações por segundo
Instâncias de replicação por conta de usuário	60
Quantidade total de armazenamento para uma instância de replicação	30.000 GB
Assinaturas de eventos por conta de usuário	60
Grupos de sub-rede de replicação por conta de usuário	60
Sub-redes por grupo de sub-rede de replicação	60

Recurso	Cota padrão
Endpoints por conta de usuário	1.000
Endpoints por instância de replicação	100
Tarefas por conta de usuário	600
Tarefas por instância de replicação	200
Certificados por conta de usuário	100
Provedores de dados por conta de usuário	1.000
Perfis de instância por conta de usuário	60
Projetos de migração por conta de usuário	10
Coletores de dados do DMS por conta de usuário	10
Recomendações de destino geradas uma vez	100
Número de arquivos que o coletor de dados do DMS pode carregar por hora	500
Migrações de dados homogêneas por conta de usuário	600
Migrações de dados homogêneas que são executadas uma vez	100
Migrações de dados homogêneas por projeto de migração	10
Replicações de tecnologia sem servidor	100

Para obter mais informações sobre a cota do controle de utilização de solicitações da API e a taxa de atualização, consulte Noções básicas sobre o controle de utilização de solicitações de API.

A cota de 30.000 GB para armazenamento se aplica a todas as instâncias de replicação do AWS DMS em uma determinada região da AWS. Esse armazenamento é utilizado para armazenar alterações em cache, se o destino não puder acompanhar a origem, e para armazenar as informações de logs.

Noções básicas sobre o controle de utilização de solicitações de API

O AWS DMS oferece suporte a uma cota de solicitações de API variável, mas um máximo de 200 chamadas de API por segundo. Ou seja, as solicitações de API são limitadas quando excedem essa taxa. Além disso, é possível se limitar a menos chamadas de API por segundo, dependendo do tempo que o AWS DMS demora para atualizar a sua cota antes de fazer outra solicitação de API. Essa cota se aplica quando você faz chamadas de API diretamente e quando elas são feitas em seu nome como parte da utilização do Console de gerenciamento do AWS DMS.

Para entender como funciona o controle de utilização de solicitações de API, é bom lembrar que o AWS DMS também mantém um bucket de token que rastreia as solicitações de API. Nesse cenário, cada token no bucket permite que você faça uma única chamada de API. Você não pode ter mais do que 200 tokens no bucket ao mesmo tempo. Quando você faz uma chamada de API, o AWS DMS remove um token do bucket. Se você fizer 200 chamadas de API em menos de um segundo, o bucket ficará vazio e haverá falha em qualquer tentativa de fazer outra chamada de API. Para cada segundo que você não faz uma chamada de API, o AWS DMS adiciona 8 tokens ao bucket, até o máximo de 200 tokens. Essa é a taxa de atualização de solicitações de API do AWS DMS. A qualquer momento após o controle de utilização, quando você tiver tokens adicionados ao bucket, é possível fazer quantas chamadas de API adicionais quantos tokens disponíveis até que as chamadas sejam limitadas pelo controle de utilização novamente.

Se você estiver utilizando o AWS CLI para executar chamadas de API que têm controle de utilização, o AWS DMS retornará um erro semelhante ao seguinte:

```
An error occurred (ThrottlingException) when calling the AwsDmsApiCall operation
 (reached max retries: 2): Rate exceeded
```

Aqui, *AwsDmsApiCall* é o nome da operação da API do AWS DMS que foi limitada pelo controle de utilização, por exemplo, `DescribeTableStatistics`. É possível tentar novamente ou fazer outra chamada após um atraso suficiente para evitar a limitação do controle de utilização.

> **Note**
>
> Ao contrário do controle de utilização de solicitações de API gerenciado por alguns outros serviços, como o Amazon EC2, não é possível pedir um aumento das cotas do controle de utilização gerenciado pelo AWS DMS.

Solução de problemas de tarefas de migração no AWS Database Migration Service

A seguir é possível encontrar tópicos sobre solução de problemas com o AWS Database Migration Service (AWS DMS). Esses tópicos podem ajudar a solucionar problemas comuns utilizando o AWS DMS e os bancos de dados de endpoints selecionados.

Se você abriu um caso de AWS Support, o engenheiro de suporte poderá identificar um possível problema com uma das configurações do banco de dados de endpoint. O engenheiro também poderá pedir que você execute um script de apoio para retornar informações de diagnóstico sobre o banco de dados. Para obter detalhes sobre como baixar, executar e fazer upload das informações de diagnóstico desse tipo de script de apoio, consulte Como trabalhar com scripts de suporte a diagnóstico no AWS DMS.

Para fins de solução de problemas, AWS DMS coleta arquivos de rastreamento e despejo na instância de replicação. Você pode fornecer esses arquivos ao AWS Support caso ocorra um problema que exija solução de problemas. Por padrão, o DMS limpa arquivos de rastreamento e despejo com mais de trinta dias. Para desativar a coleta de arquivos de rastreamento e despejo, abra um caso com o AWS Support.

Tópicos

- O número de tabelas por tarefa causa problemas
- Falha nas tarefas quando a chave primária é criada na coluna LOB
- Registros duplicados ocorrem na tabela de destino sem chave primária
- Os endpoints de origem ficam no intervalo IP reservado
- Os timestamps são distorcidos em consultas do Amazon Athena
- Solução de problemas com o Oracle
- Solução de problemas com o MySQL
- Solução de problemas com o PostgreSQL
- Solução de problemas com o Microsoft SQL Server
- Solução de problemas com o Amazon Redshift
- Solução de problemas com o MySQL do Amazon Aurora
- Solução de problemas com o SAP ASE
- Solução de problemas com o IBM Db2
- Solução de problemas de latência no AWS Database Migration Service
- Como trabalhar com scripts de suporte a diagnóstico no AWS DMS
- Trabalhando com o suporte AWS DMS de diagnóstico AMI

As tarefas de migração são executadas lentamente

Diversos problemas podem fazer com que uma tarefa de migração seja executada lentamente ou fazer com que tarefas subsequentes sejam executadas mais lentamente que a tarefa inicial.

O motivo mais comum da lentidão da execução de uma tarefa de migração é que há recursos inadequados alocados à instância de replicação do AWS DMSDMS. Para verificar se a instância tem recursos suficientes para as tarefas que você está executando nela, confira o uso de CPU, de memória, de troca de arquivos e de IOPS. Por exemplo, várias tarefas com o Amazon Redshift como endpoint têm uso intensivo de E/S. É possível aumentar a IOPS da instância de replicação ou separar as tarefas em várias instâncias de replicação para obter uma migração mais eficaz.

Para obter mais informações sobre como determinar o tamanho da instância de replicação, consulte Seleção do melhor tamanho para uma instância de replicação.

É possível aumentar a velocidade de um carregamento de migração inicial fazendo o seguinte:

- Se o destino for uma instância de banco de dados Amazon RDS, verifique se multi-AZ não está ativado para a instância de banco de dados de destino.
- Desligue todos os backups automáticos ou os registros em log no banco de dados de destino durante a carga e religue-os assim que a migração for concluída.
- Se o recurso estiver disponível no destino, utilize IOPS provisionadas.
- Se os dados de migração contiverem LOBs, verifique se a tarefa está otimizada para a migração de LOB. Para obter mais informações sobre a otimização de LOBs, consulte Configurações de tarefa de metadados de destino.

A barra de status da tarefa não se move

A barra de status de tarefa fornece uma estimativa do andamento da tarefa. A qualidade dessa estimativa depende da qualidade das estatísticas de tabela do banco de dados de origem: quanto melhores as estatísticas de tabela, mais precisa a estimativa.

Para uma tarefa com apenas uma tabela que não tem nenhuma estatística de linhas estimada, o AWS DMS não pode fornecer nenhum tipo de estimativa de porcentagem de conclusão. Nesse caso, utilize o estado da tarefa e a indicação das linhas carregadas para confirmar se a tarefa realmente está sendo executada e progredindo.

A tarefa foi concluída, mas nada foi migrado

Faça o seguinte se nada tiver sido migrado após a conclusão da tarefa.

- Verifique se o usuário que criou o endpoint tem acesso de leitura à tabela que você pretende migrar.
- Verifique se o objeto que você deseja migrar é uma tabela. Se for uma visualização, atualize os mapeamentos da tabela e especifique o localizador de objetos como “visualização” ou “tudo”. Para ter mais informações, consulte Especificar a seleção de tabelas e as regras de transformação no console.

Chaves estrangeiras e índices secundários ausentes

O AWS DMS cria tabelas, chaves primárias e, em alguns casos, índices exclusivos, mas não cria nenhum outro objeto que não seja necessário para migrar os dados da origem de forma eficiente.

Por exemplo, ele não cria índices secundários, restrições de chave não primária ou padrões de dados.

Para migrar objetos secundários do seu banco de dados, utilize as ferramentas nativas dele se estiver migrando para o mesmo mecanismo de banco de dados que o seu banco de dados de origem. Utilize o AWS Schema Conversion Tool (AWS SCT) se estiver migrando para um mecanismo de banco de dados diferente do usado pelo banco de dados de origem para migrar objetos secundários.

AWS DMSnão cria CloudWatch registros

Se sua tarefa de replicação não criar CloudWatch registros, verifique se sua conta tem a `dms-cloudwatch-logs-role` função. Se esse perfil não estiver presente, faça o seguinte para criá-lo:

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. Escolha a guia Perfis. Selecione Criar perfil.
3. Na seção Selecionar tipo de entidade confiável, escolha AWS service (Serviço da AWS).
4. Na seção Escolher um caso de uso, escolha DMS.
5. Escolha Próximo: permissões.
6. Entre **`AmazonDMSCloudWatchLogsRole`** no campo de pesquisa e marque a caixa ao lado do CloudWatchLogsRoleAmazonDMS. Isso concede AWS DMS permissões de acesso CloudWatch.
7. Escolha Próximo: etiquetas.
8. Escolha Próximo: revisar.
9. Insira **`dms-cloudwatch-logs-role`** em Nome do perfil. Esse nome não diferencia maiúsculas de minúsculas.
10. Selecione Criar perfil.

Ocorrem problemas com a conexão com o Amazon RDS

Pode haver vários motivos porque você não pode se conectar a uma instância de banco de dados Amazon RDS definida como origem ou destino. Seguem alguns itens a serem verificados:

- Verifique se a combinação do nome do usuário e da senha está correta.

- Verifique se o valor do endpoint mostrado no console do Amazon RDS para a instância é o mesmo do identificador do endpoint que você utilizou para criar o endpoint do AWS DMS.
- Verifique se o valor da porta mostrado no console do Amazon RDS para a instância é o mesmo da porta atribuída ao endpoint do AWS DMS.
- Verifique se o grupo de segurança atribuído à instância de banco de dados Amazon RDS permite conexões da instância de replicação do AWS DMS.
- Se a instância de replicação do AWS DMS e a instância de banco de dados Amazon RDS não estiverem na mesma nuvem privada virtual (VPC), verifique se a instância de banco de dados é publicamente acessível.

Mensagem de erro: string de conexão de thread incorreta: valor de thread incorreto 0

Esse erro costuma ocorrer quando você está testando a conexão a um endpoint. Esse erro indica que há um erro na string de conexão. Um exemplo é um espaço após o endereço IP do host. Outro é um caractere inválido copiado na string de conexão.

Ocorrem problemas de rede

O problema mais comum de rede envolve o grupo de segurança da VPC usado pela instância de replicação do AWS DMS. Por padrão, esse security group tem regras que permitem a saída para 0.0.0.0/0 em todas as portas. Em muitos casos, você modifica esse grupo de segurança ou utiliza o seu próprio grupo de segurança. Nesse caso, no mínimo, verifique se você fornece saída aos endpoints de origem e de destino nas respectivas portas de banco de dados.

Outros problemas relacionados à configuração podem incluir o seguinte:

- A instância de replicação e os endpoints de origem e de destino na mesma VPC: o grupo de segurança utilizado pelos endpoints deve permitir a entrada na porta do banco de dados da instância de replicação. Verifique se o grupo de segurança utilizado pela instância de replicação tem entrada para os endpoints. Ou crie uma regra no grupo de segurança utilizado pelos endpoints que permita acesso ao endereço IP privado da instância de replicação.
- O endpoint de origem está fora da VPC utilizada pela instância de replicação (utilizando um gateway da Internet): o grupo de segurança da VPC deve incluir regras de roteamento que enviam o tráfego não destinado à VPC para o gateway da Internet. Nessa configuração, a conexão ao endpoint parecerá vir do endereço IP público na instância de replicação.

- O endpoint de origem está fora da VPC utilizada pela instância de replicação (utilizando um gateway NAT): é possível configurar um gateway de conversão de endereços de rede (NAT) utilizando um único endereço IP elástico associado a uma única interface de rede elástica. Esse gateway NAT recebe um identificador NAT (nat-#####).

  Em alguns casos, a VPC inclui uma rota padrão para esse gateway NAT em vez do gateway da Internet. Nesses casos, a instância de replicação parece entrar em contato com o endpoint do banco de dados utilizando o endereço IP público do gateway NAT. Aqui, a entrada no endpoint do banco de dados fora da VPC precisa permitir a entrada do endereço NAT em vez do endereço IP público da instância de replicação.

Para obter informações sobre como utilizar seu próprio servidor de nomes on-premises, consulte Utilização do seu próprio servidor de nomes on-premises.

A CDC fica paralisada após carga máxima

As alterações de replicação lenta ou paralisada podem ocorrer após uma migração de carga máxima, quando várias configurações do AWS DMS entram em conflito entre si.

Por exemplo, suponha que o parâmetro do Modo de preparação da tabela de destino esteja definido como Não fazer nada ou Truncar. Nesse caso, você instruiu o AWS DMS a não fazer nenhuma configuração nas tabelas de destino, incluindo a criação de índices primários e exclusivos. Se você não criou chaves primárias ou exclusivas nas tabelas de destino, o AWS DMS fará uma varredura completa de tabela para cada atualização. Essa abordagem pode afetar significativamente o desempenho.

Erros de violação de chave primária ocorrem ao reiniciar uma tarefa

Esse erro pode ocorrer quando os dados permanecem no banco de dados de destino de uma tarefa de migração anterior. Se o parâmetro Modo de preparação de tabela de destino estiver definido como Não fazer nada, o AWS DMS não fará nenhuma preparação na tabela de destino, incluindo a limpeza de dados inseridos de uma tarefa anterior.

Para reiniciar a tarefa e evitar esses erros, remova as linhas inseridas nas tabelas de destino da execução anterior da tarefa.

Falha na carga inicial de um esquema

Em alguns casos, a carga inicial dos esquemas pode falhar com o erro `Operation:getSchemaListDetails:errType=, status=0, errMessage=, errDetails=`.

Nesses casos, a conta de usuário utilizada pelo AWS DMS para se conectar ao endpoint de origem não tem as permissões necessárias.

Falha em tarefas com erro desconhecido

A causa de tipos de erro desconhecidos pode ser variada. No entanto, frequentemente descobrimos que o problema envolve recursos insuficientes alocados para a instância de replicação do AWS DMS.

Para garantir que a instância de replicação tenha recursos suficientes para executar a migração, verifique o uso de CPU, de memória, de troca de arquivos e de IOPS das instâncias. Para obter mais informações sobre monitoramento, consulte Métricas do AWS Database Migration Service.

Tarefa recomeça o carregamento de tabelas desde o início

O AWS DMS reinicia a carga da tabela desde o início quando ainda não concluiu a carga inicial de uma tabela. Quando uma tarefa é reiniciada, o AWS DMS recarrega as tabelas desde o início, quando o carregamento inicial não foi concluído.

O número de tabelas por tarefa causa problemas

Não há limite definido para o número de tabelas por tarefa de replicação. No entanto, é recomendável limitar o número de tabelas em uma tarefa a menos de 60.000, como regra geral. A utilização de recursos pode representar um gargalo quando uma única tarefa utiliza mais de 60.000 tabelas.

Falha nas tarefas quando a chave primária é criada na coluna LOB

No modo FULL LOB ou LIMITED LOB, o AWS DMS não é compatível com replicação de chaves primárias que são tipos de dados LOB.

Inicialmente, o DMS migra uma linha com uma coluna LOB como nula e, posteriormente, atualiza a coluna LOB. Portanto, quando a chave primária é criada em uma coluna LOB, a inserção inicial falha, uma vez que a chave primária não pode ser nula. Como solução alternativa, adicione outra coluna como chave primária e remova a chave primária da coluna LOB.

Registros duplicados ocorrem na tabela de destino sem chave primária

A execução de uma tarefa de carga máxima e CDC pode criar registros duplicados em tabelas de destino sem uma chave primária ou um índice exclusivo. Para evitar a duplicação de registros nas tabelas de destino durante tarefas de carga máxima e CDC, verifique se as tabelas de destino têm uma chave primária ou um índice exclusivo.

Os endpoints de origem ficam no intervalo IP reservado

Se um banco de dados de origem do AWS DMS utilizar um endereço IP dentro do intervalo IP reservado de 192.168.0.0/24, o teste de conexão do endpoint de origem falhará. As etapas a seguir fornecem uma possível solução alternativa:

1. Localize uma instância do Amazon EC2 que não esteja no intervalo reservado que possa se comunicar com o banco de dados de origem em 192.168.0.0/24.
2. Instale um proxy socat e execute-o. Por exemplo:

```
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
&
```

Utilize o endereço IP da instância do Amazon EC2 e a porta do banco de dados fornecida anteriormente para o endpoint do AWS DMS. Verifique se o endpoint tem o grupo de segurança que permite que o AWS DMS acesse a porta do banco de dados. Observe que o proxy precisa estar em execução durante a execução da tarefa do DMS. Dependendo do caso de uso, talvez seja necessário automatizar a configuração do proxy.

Os timestamps são distorcidos em consultas do Amazon Athena

Se os carimbos de data/hora estiverem distorcidos nas consultas do Athena, use a ação ou AWS Management Console a para definir o valor ModifyEndpointdo seu endpoint do Amazon `parquetTimestampInMillisecond` S3 como. `true` Para obter mais informações, consulte S3Settings.

Solução de problemas com o Oracle

A seguir, você aprenderá sobre solução de problemas específicos para utilização do AWS DMS com bancos de dados Oracle.

Tópicos

- Extrair de dados de exibições
- Migração de LOBs do Oracle 12c
- Alternando entre Oracle LogMiner e Binary Reader
- Erro: Oracle CDC stopped 122301 Oracle CDC maximum retry counter exceeded.
- Adição automática de registro em log complementar a um endpoint de origem Oracle
- Alterações de LOB não estão sendo capturadas
- Erro: ORA-12899: value too large for column column-name
- Tipo de dados NUMBER sendo mal interpretado
- Registros ausentes durante a carga máxima
- Erro de tabela
- Erro: não é possível recuperar IDs de destino de Redo Log arquivados do Oracle
- Avaliação do desempenho de leitura de redo logs ou de arquivamento do Oracle

Extrair de dados de exibições

É possível extrair dados uma vez de uma visualização, mas não pode utilizá-la para replicação contínua. Para poder extrair dados de visualizações, adicione o código a seguir à seção Configurações de Endpoint da página de endpoint de origem do Oracle. Ao extrair dados de uma visualização, a visualização aparece como uma tabela no esquema de destino.

```
"ExposeViews": true
```

Migração de LOBs do Oracle 12c

AWS DMSpode usar dois métodos para capturar alterações em um banco de dados Oracle, Binary Reader e Oracle LogMiner. Por padrão, AWS DMS usa o Oracle LogMiner para capturar alterações. No entanto, no Oracle 12c, o Oracle LogMiner não oferece suporte a colunas LOB. Para capturar alterações em colunas de LOB no Oracle 12c, utilize o Binary Reader.

Alternando entre Oracle LogMiner e Binary Reader

AWS DMSpode usar dois métodos para capturar alterações em um banco de dados Oracle de origem, Binary Reader e Oracle LogMiner. O Oracle LogMiner é o padrão. Para alternar e usar o Binary Reader para capturar alterações, faça o seguinte:

Como usar o Binary Reader para capturar alterações

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Endpoints.
3. Escolha o endpoint de origem do Oracle em que deseja utilizar o Binary Reader.
4. Escolha Modificar.
5. Escolha Avançado e adicione o código a seguir a Atributos de conexão adicionais.

```
useLogminerReader=N
```

6. Utilize uma ferramenta de desenvolvedor do Oracle, como SQL-Plus, para conceder o seguinte privilégio adicional à conta de usuário do AWS DMS utilizada para conectar-se ao endpoint do Oracle.

```
SELECT ON V_$TRANSPORTABLE_PLATFORM
```

Erro: Oracle CDC stopped 122301 Oracle CDC maximum retry counter exceeded.

Esse erro ocorre quando os logs de arquivo do Oracle necessários foram removidos do servidor antes de o AWS DMS usá-los para capturar alterações. Aumente as políticas de retenção de logs no servidor de banco de dados. Para um banco de dados Amazon RDS, execute o seguinte procedimento para aumentar a retenção de logs. Por exemplo, o seguinte código aumenta a retenção de logs em uma instância de banco de dados Amazon RDS para 24 horas.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24);
```

Adição automática de registro em log complementar a um endpoint de origem Oracle

Por padrão, o AWS DMS fica com o registro suplementar desativado. Para ativar automaticamente o registro suplementar de um endpoint de origem do Oracle, faça o seguinte:

Como adicionar o registro complementar a um endpoint de origem do Oracle

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Endpoints.
3. Selecione o endpoint de origem do Oracle ao qual deseja adicionar o log complementar.
4. Escolha Modificar.
5. Selecione Avançado e adicione o seguinte código à caixa de texto Atributos de conexão extra:

   ```
   addSupplementalLogging=Y
   ```

6. Escolha Modificar.

Alterações de LOB não estão sendo capturadas

No momento, uma tabela deve ter uma chave primária para o AWS DMS capturar alterações de LOB. Se uma tabela que contém LOBs não tem uma chave primária, há várias ações que você pode realizar para capturar alterações de LOB:

- Adicione uma chave primária à tabela. Isso pode ser tão simples quanto adicionar uma coluna de ID e preenchê-la com uma sequência utilizando um trigger.
- Crie uma visão materializada da tabela que inclua um ID gerado pelo sistema como a chave primária e migre a visão materializada em vez da tabela.
- Crie uma espera lógica, adicione uma chave primária à tabela e migre a partir da espera lógica.

Erro: ORA-12899: value too large for column *column-name*

O erro “ORA-12899: valor muito grande para o *nome da coluna*” geralmente é causado por dois problemas.

Em um desses problemas, há uma incompatibilidade nos conjuntos de caracteres utilizados pelos bancos de dados de origem e de destino.

Em outro desses problemas, as configurações com compatibilidade ao idioma nacional (NLS) diferem entre os dois bancos de dados. Uma causa comum desse erro é quando o parâmetro NLS_LENGTH_SEMANTICS do banco de dados de origem é definido como CHAR e o parâmetro NLS_LENGTH_SEMANTICS do banco de dados de destino é definido como BYTE.

Tipo de dados NUMBER sendo mal interpretado

O tipo de dados NUMBER do Oracle é convertido em vários tipos de dados do AWS DMS, dependendo da precisão e da escala de NUMBER. Essas conversões estão documentadas aqui Tipos de dados de origem do Oracle. A forma como o tipo NUMBER é convertido também pode ser afetada pela utilização de configurações do endpoint do Oracle de origem. Essas configurações de endpoint são documentadas emConfigurações de endpoint ao usar o Oracle como fonte para AWS DMS.

Registros ausentes durante a carga máxima

Ao executar uma carga máxima, o AWS DMS procura transações abertas no nível do banco de dados e espera que a transação seja confirmada. Por exemplo, com base na configuração da tarefa

`TransactionConsistencyTimeout=600`, o AWS DMS espera por 10 minutos mesmo que a transação aberta esteja em uma tabela não incluída no mapeamento da tabela. Mas se a transação aberta estiver em uma tabela incluída no mapeamento da tabela e a transação não for confirmada a tempo, haverá registros ausentes na tabela de destino.

É possível modificar a configuração da tarefa `TransactionConsistencyTimeout` e aumentar o tempo de espera quando você sabe que as transações abertas levarão mais tempo para serem confirmadas.

Além disso, observe que o valor padrão da configuração da tarefa `FailOnTransactionConsistencyBreached` é `false`. Isso significa que o AWS DMS continua aplicando outras transações, mas as transações abertas são perdidas. Se você quiser que a tarefa falhe quando as transações abertas não forem fechadas a tempo, poderá definir `FailOnTransactionConsistencyBreached` como `true`.

Erro de tabela

`Table Error` aparecerá nas estatísticas da tabela durante a replicação se uma cláusula `WHERE` não fizer referência a uma coluna de chave primária e o registro em log suplementar não for usado para todas as colunas.

Para corrigir esse problema, ative o registro em log suplementar de todas as colunas da tabela referenciada. Para ter mais informações, consulte Configuração de registro em log suplementar.

Erro: não é possível recuperar IDs de destino de Redo Log arquivados do Oracle

Esse erro ocorre quando a origem Oracle não tem nenhum log de arquivamento gerado ou quando V $ARCHIVED_LOG está vazio. É possível resolver o erro trocando os logs manualmente.

Para um banco de dados Amazon RDS, execute o procedimento a seguir para trocar os arquivos de log. O procedimento `switch_logfile` não tem parâmetros.

```
exec rdsadmin.rdsadmin_util.switch_logfile;
```

Para um banco de dados de origem Oracle autogerenciado, utilize o comando a seguir para forçar uma troca de log.

```
ALTER SYSTEM SWITCH LOGFILE ;
```

Avaliação do desempenho de leitura de redo logs ou de arquivamento do Oracle

Se você tiver problemas de desempenho com a origem do Oracle, poderá avaliar o desempenho de leitura dos redo logs ou do arquivamento do Oracle para encontrar maneiras de melhorar o desempenho. Para testar o desempenho de leitura dos redo logs ou de arquivamento, utilize a imagem de máquina da Amazon (AMI) de diagnóstico do AWS DMS.

É possível utilizar a AMI de diagnóstico do AWS DMS para fazer o seguinte:

- Utilizar o método bFile para avaliar o desempenho do arquivo de redo log.
- Use o LogMiner método para avaliar o desempenho do arquivo de redo log.
- Utilizar o método PL/SQL (`dbms_lob.read`) para avaliar o desempenho do arquivo de redo log.
- Utilizar Single-thread para avaliar o desempenho de leitura no ASMFile.
- Utilizar Single-threads para avaliar o desempenho de leitura no ASMFile.
- Utilize o perfil Direct OS Readfile() Windows ou Pread64 Linux para avaliar o arquivo de redo log.

É possível tomar medidas corretivas com base nos resultados.

Para testar o desempenho de leitura em um arquivo de redo log ou de arquivamento do Oracle

1. Crie uma instância de diagnóstico da AMI do Amazon EC2 do AWS DMS e conecte-se a ela.

   Para obter mais informações, consulte Como trabalhar com a AMI de diagnóstico do AWS DMS.
2. Execute o comando awsreplperf.

   ```
   $ awsreplperf
   ```

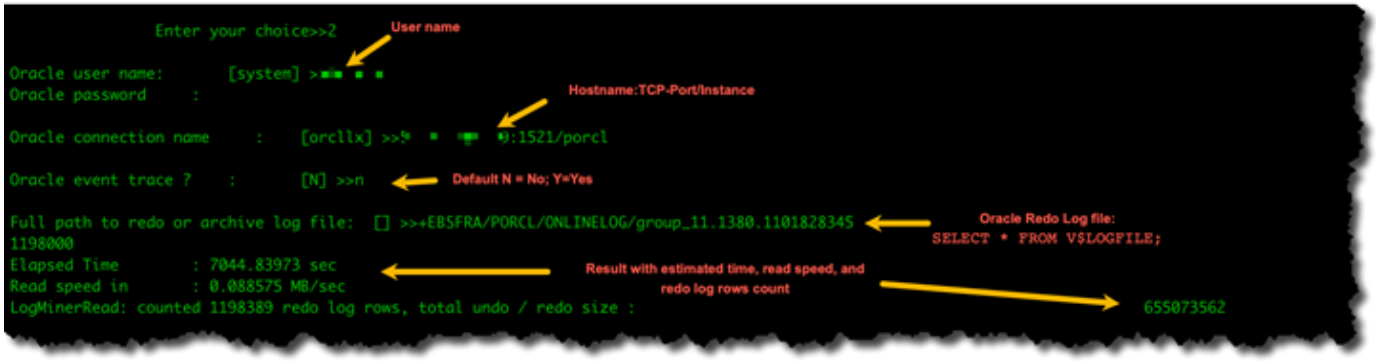
   O comando exibe as opções do AWS DMS do utilitário de desempenho de leitura do Oracle.

   ```
   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. Selecione uma opção na lista.
4. Insira as seguintes informações de conexão do banco de dados e do log de arquivamento.

```
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. Examine a saída exibida para obter informações relevantes sobre o desempenho de leitura. Por exemplo, o seguinte mostra a saída que pode resultar da seleção da opção número 2, Ler usando LogMiner.



6. Para sair do utilitário, insira 0 (zero).

Próximas etapas

- Quando os resultados mostrarem que a velocidade de leitura está abaixo de um limite aceitável, execute o Script apoio de diagnóstico do Oracle no endpoint, revise as seções Tempo de espera, Perfil de carga e Perfil de E/S. Ajuste qualquer configuração anormal que possa melhorar o desempenho de leitura. Por exemplo, se os arquivos de redo log tiverem até 2 GB, tente aumentar LOG_BUFFER para 200 MB para ajudar a melhorar o desempenho.
- Analise as Práticas recomendadas do AWS DMS para garantir que a instância, a tarefa e os endpoints de replicação do DMS estejam configurados de forma ideal.

Solução de problemas com o MySQL

A seguir, você aprenderá sobre a solução de problemas específicos para utilização do AWS DMS com bancos de dados MySQL.

Tópicos

Falha na tarefa de CDC para o endpoint da instância de banco de dados Amazon RDS porque o registro em log binário está desativado

Esse problema ocorre com instâncias do banco de dados Amazon RDS porque os backups automáticos estão desabilitados. Para habilitar backups automáticos, configure o período de retenção de backup para um valor diferente de zero.

Conexões a uma instância de destino MySQL são desconectadas durante uma tarefa

Se você tiver uma tarefa com LOBs que está sendo desconectada de um destino MySQL, poderá ver o seguinte tipo de erros no log de tarefas.

```
[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.
```

Nesse caso, poderá ser necessário ajustar algumas das configurações da tarefa.

Para resolver o problema em que uma tarefa está sendo desconectada de um destino MySQL, faça o seguinte:

- Confirme se a sua variável de banco de dados `max_allowed_packet` está definida como grande o suficiente para reter o maior LOB.
- Verifique se você tem as seguintes variáveis definidas para ter um valor de tempo limite grande. Sugerimos que você utilize um valor de, pelo menos, 5 minutos para cada uma dessas variáveis.
  - `net_read_timeout`
  - `net_write_timeout`
  - `wait_timeout`

Para obter informações sobre a configuração das variáveis de sistema do MySQL, consulte Variáveis do sistema do servidor na Documentação do MySQL.

Adicionar confirmação automática a um endpoint compatível com MySQL

Como adicionar autocommit a um endpoint de destino compatível com MySQL

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Endpoints.
3. Escolha o endpoint de destino compatível com MySQL ao qual você deseja adicionar autocommit.
4. Escolha Modificar.
5. Selecione Avançado e adicione o seguinte código à caixa de texto Atributos de conexão extra:

```
Initstmt= SET AUTOCOMMIT=1
```

6. Escolha Modificar.

Desabilitar chaves externas em um endpoint de destino compatível com MySQL

É possível desativar as verificações de chaves estrangeiras no MySQL adicionando o seguinte a Atributos de conexão adicionais, na seção Avançado do endpoint de destino MySQL, da edição compatível com MySQL do Amazon Aurora ou do MariaDB.

Como desabilitar chaves externas em um endpoint de destino compatível com MySQL

1. Faça login no AWS Management Console e abra o AWS DMS console em https://console.aws.amazon.com/dms/v2/.
2. Escolha Endpoints.
3. Escolha o endpoint de destino MySQL, Aurora MySQL ou MariaDB em que deseja desativar as chaves estrangeiras.
4. Escolha Modificar.
5. Selecione Avançado e adicione o seguinte código à caixa de texto Atributos de conexão extra:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0
```

6. Escolha Modificar.

Caracteres substituídos por interrogações

A situação que mais causa esse problema é quando os caracteres do endpoint de origem foram codificados por um conjunto de caracteres não compatíveis com o AWS DMS.

Entradas de log de "eventos inválidos"

As entradas "Eventos inválidos" nos logs de migração costumam indicar que houve tentativa de uma operação de linguagem de definição de dados (DDL) incompatível no endpoint do banco de dados de origem. Operações DDL incompatíveis provocam um evento que a instância de replicação não pode ignorar, portanto, um evento inválido é registrado em log.

Para corrigir esse problema, reinicie a tarefa desde o início. Isso recarrega as tabelas e começa a capturar as alterações em um ponto após a emissão da operação DDL incompatível.

Captura de dados de alteração com MySQL 5.5

A captura de dados de alteração (CDC) do AWS DMS de bancos de dados do Amazon RDS compatível com MySQL exige registro em log binário baseado em linhas de imagem completa, o que não é compatível com o MySQL versão 5.5 ou inferior. Para usar o CDC do AWS DMS, você deve atualizar a sua instância de banco de dados Amazon RDS para MySQL versão 5.6.

Aumentar a retenção de log binário para instâncias de banco de dados Amazon RDS

O AWS DMS exige a retenção de arquivos de log binário para a captura de dados de alteração. Para aumentar a retenção de logs em uma instância de banco de dados Amazon RDS, utilize o procedimento a seguir. O exemplo a seguir aumenta a retenção de log binário para 24 horas.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Mensagem de log: Algumas alterações do banco de dados de origem não tiverem impacto ao serem aplicadas ao banco de dados de destino.

Quando o AWS DMS atualiza um valor da coluna do banco de dados MySQL para o valor existente, uma mensagem de `zero rows affected` é gerada pelo MySQL. Esse comportamento é diferente de outros mecanismos de banco de dados, como o Oracle e o SQL Server. Esses mecanismos atualizam uma linha, mesmo quando o valor de substituição é igual ao atual.

Erro: Identifier too long

O erro a seguir ocorre quando um identificador é muito longo:

```
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)
```

Em alguns casos, você configura o AWS DMS para criar as tabelas e as chaves primárias no banco de dados de destino. Nesses casos, atualmente o DMS não utiliza os mesmos nomes para as chaves primárias que foram utilizadas no banco de dados de origem. Em vez disso, o DMS cria o nome da chave primária baseado no nome da tabela. Quando o nome da tabela é longo, o identificador gerado automaticamente pode ser mais longo que os limites permitidos para o MySQL.

Para solucionar esse problema, a abordagem atual é primeiro pré-criar as tabelas e as chaves primárias no banco de dados de destino. Utilize uma tarefa com a configuração de tarefa Modo de preparação da tabela de destino definido como Não fazer nada ou Truncar para preencher as tabelas de destino.

Erro: conjunto de caracteres incompatível causa falha na conversão de dados de campos

O erro a seguir ocorre quando um conjunto de caracteres não suportado causa a falha de uma conversão de dados de campo:

```
"[SOURCE_CAPTURE ]E: Column 'column-name' uses an unsupported character set [120112]
A field data conversion failed. (mysql_endpoint_capture.c:2154)
```

Verifique os parâmetros do banco de dados relativos a conexões. O comando a seguir pode ser utilizado para definir esses parâmetros:

```
SHOW VARIABLES LIKE '%char%';
```

Erro: página de código 1252 para UTF8 [120112] Uma conversão de dados de campo falhou

O erro a seguir pode ocorrer durante uma migração se você tiver caracteres que não sejam da página de código 1252 no banco de dados MySQL de origem.

```
[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)
```

Como alternativa, você pode usar o atributo de conexão extra `CharsetMapping` com o endpoint MySQL de origem para especificar o mapeamento de conjuntos de caracteres. Talvez seja necessário reiniciar a tarefa de migração do AWS DMS desde o início se você adicionar essa configuração de endpoint.

Por exemplo, a configuração de endpoint a seguir pode ser utilizada para um endpoint do MySQL de origem em que o conjunto de caracteres de origem é `Utf8` ou `latin1`. 65001 é o identificador da página de código UTF8.

```
CharsetMapping=utf8,65001
CharsetMapping=latin1,65001
```

Índices, chaves estrangeiras ou atualizações ou exclusões em cascata não migrados

O AWS DMS não é compatível com a migração de objetos secundários, como índices e chaves estrangeiras. Para replicar as alterações feitas em tabelas secundárias em uma operação de atualização ou de exclusão em cascata, você precisa ter a restrição de chave estrangeira acionadora ativa na tabela de destino. Para contornar essa limitação, crie a chave estrangeira manualmente na tabela de destino. Crie uma única tarefa para carga máxima e CDC, ou duas tarefas separadas para carga máxima e CDC, conforme descrito a seguir:

Criar uma única tarefa compatível com carga máxima e CDC

Este procedimento descreve como migrar chaves estrangeiras e índices utilizando uma única tarefa de carga máxima e CDC.

Criar uma tarefa de carga máxima e CDC

1. Crie manualmente as tabelas com chaves estrangeiras e índices no destino para que correspondam às tabelas de origem.
2. Adicione o ECA a seguir ao endpoint de destino do AWS DMS:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. Crie a tarefa do AWS DMS com `TargetTablePrepMode` definido como `DO_NOTHING`.
4. Defina a configuração `Stop task after full load completes` como `StopTaskCachedChangesApplied`.
5. Inicie a tarefa. O AWS DMS interrompe a tarefa automaticamente depois que ela conclui a carga máxima e aplica todas as alterações em cache.
6. Remova o ECA do `SET FOREIGN_KEY_CHECKS` adicionado anteriormente.
7. Retome a tarefa. A tarefa entra na fase de CDC e aplica as alterações em andamento no banco de dados de origem para o de destino.

Crie tarefas de carga máxima e de CDC separadamente

Este procedimento descreve como migrar chaves estrangeiras e índices utilizando tarefas separadas de carga máxima e de CDC.

Criar uma tarefa de carga máxima

1. Crie manualmente as tabelas com chaves estrangeiras e índices no destino para que correspondam às tabelas de origem.
2. Adicione o ECA a seguir ao endpoint de destino do AWS DMS:

```
Initstmt=SET FOREIGN_KEY_CHECKS=0;
```

3. Crie a tarefa do AWS DMS com o parâmetro `TargetTablePrepMode` definido como `DO_NOTHING` e `EnableValidation` definido como `FALSE`.

4. Inicie a tarefa. O AWS DMS interrompe a tarefa automaticamente depois que ela conclui a carga máxima e aplica todas as alterações em cache.
5. Depois que a tarefa for concluída, anote a hora de início da tarefa de carga máxima em UTC ou o nome e a posição do arquivo de log binário, para iniciar somente a tarefa de CDC. Consulte os logs para obter o timestamp em UTC do horário de início da carga máxima.

Criar uma tarefa somente de CDC

1. Remova o ECA do `SET FOREIGN_KEY_CHECKS` definido anteriormente.
2. Crie a tarefa somente de CDC com a posição de início definida como a hora de início da carga máxima indicada na etapa anterior. Como alternativa, é possível utilizar a posição do log binário registrada na etapa anterior. Defina a configuração `TargetTablePrepMode` como `DO_NOTHING`. Para ativar a validação de dados, defina a configuração do `EnableValidation` como `TRUE` se necessário.
3. Inicie a tarefa somente de CDC e monitore os logs para verificar erros.

> **Note**
>
> Essa solução alternativa só se aplica a uma migração de MySQL para MySQL. Não é possível utilizar esse método com o recurso de aplicação em lotes, porque a aplicação em lotes exige que as tabelas de destino não tenham chaves estrangeiras ativas.

Solução de problemas com o PostgreSQL

A seguir, você aprenderá sobre a solução de problemas específicos para utilização do AWS DMS com bancos de dados PostgreSQL.

Tópicos

- Impedir que o PostgreSQL capture DDL
- Selecionar o esquema em que os objetos de banco de dados para a captura DDL são criados
- Tabelas do Oracle ausentes após a migração para o PostgreSQL
- ReplicationSlotDiskUsage aumenta e restart_lsn para de avançar durante transações longas, como cargas de trabalho de ETL
- Tarefa utilizando visualização como uma origem não tem nenhuma linha copiada

Tipos de dados JSON que estão sendo truncados

O AWS DMS trata os tipos de dados JSON no PostgreSQL como uma coluna de tipos de dados LOB. Isso significa que a limitação de tamanho de LOB ao utilizar o modo LOB limitado se aplica a dados JSON.

Por exemplo, suponha que o modo LOB limitado esteja definido como 4.096 KB. Nesse caso, qualquer dado JSON maior que 4.096 KB é truncado no limite de 4.096 KB e falha no teste de validação no PostgreSQL.

Por exemplo, as seguintes informações de log mostram o JSON que foi truncado devido à configuração do modo LOB limitado e à falha na validação.

```
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)
```

Colunas de tipo de dados definido pelo usuário não estão sendo migradas corretamente

Ao replicar de uma origem do PostgreSQL, o AWS DMS cria a tabela de destino com os mesmos tipos de dados para todas as colunas, além das colunas com tipos de dados definidos pelo usuário. Nesses casos, o tipo de dados é criado como "variante de caractere" no destino.

Erro: No schema has been selected to create in

Em alguns casos, você pode ver o erro “SQL_ERROR SqlState: 3F000:7 Message NativeError: ERROR: no schema has selected to create in”.

Esse erro pode ocorrer quando o mapeamento da tabela JSON contém um valor curinga para o esquema, mas o banco de dados de origem não é compatível com esse valor.

Exclusões e atualizações em uma tabela não estão sendo replicadas utilizando a CDC

As operações de exclusão e de atualização durante a captura de dados de alteração (CDC) serão ignoradas se a tabela de origem não tiver uma chave primária. O AWS DMS é compatível com a captura de dados de alteração (CDC) para tabelas do PostgreSQL com chaves primárias.

Se uma tabela não tiver uma chave primária, os logs de gravação antecipada (WAL) não incluirão uma imagem anterior da linha do banco de dados. Nesse caso, o AWS DMS não atualiza a tabela. Para operações de exclusão a serem replicadas, crie uma chave primária na tabela de origem.

Instruções de truncamento não estão sendo propagadas

Ao utilizar a captura de dados de alteração (CDC), as operações TRUNCATE não são compatíveis com o AWS DMS.

Impedir que o PostgreSQL capture DDL

É possível impedir que um endpoint de destino do PostgreSQL capture instruções DDL adicionando a seguinte instrução de Configuração de endpoint.

```
"CaptureDDLs": "N"
```

Selecionar o esquema em que os objetos de banco de dados para a captura DDL são criados

É possível controlar o schema onde os objetos de banco de dados relacionados à captura DDL são criados. Adicione a seguinte instrução de Configuração de endpoint. O parâmetro de Configuração de endpoint está disponível na guia do endpoint de origem.

```
"DdlArtifactsSchema: "xyzddlschema"
```

Tabelas do Oracle ausentes após a migração para o PostgreSQL

Nesse caso, as tabelas e dados geralmente ainda estão acessíveis.

O Oracle padroniza os nomes de tabelas com letras maiúsculas, enquanto o PostgreSQL as padroniza com minúsculas. Ao executar uma migração do Oracle para o PostgreSQL, sugerimos fornecer certas regras de transformação na seção de mapeamento de tabela da tarefa. Essas são as regras de transformação para converter maiúsculas e minúsculas dos nomes de tabelas.

Se você migrou as tabelas sem utilizar as regras de transformação para converter maiúsculas e minúsculas dos nomes das tabelas, será necessário inserir os nomes de tabelas entre aspas ao se referir a elas.

ReplicationSlotDiskUsage aumenta e restart_lsn para de avançar durante transações longas, como cargas de trabalho de ETL

Quando a replicação lógica está ativada, o número máximo de alterações mantidas na memória por transação é de 4 MB. Depois disso, as alterações são transferidas para o disco. Como resultado, o `ReplicationSlotDiskUsage` aumenta, e o `restart_lsn` não avança até que a transação seja concluída/abortada e a reversão seja concluída. Como é uma transação longa, ela pode demorar muito tempo para reverter.

Portanto, evite transações de longa execução quando a replicação lógica estiver ativada. Em vez disso, tente dividir a transação em várias transações menores.

Tarefa utilizando visualização como uma origem não tem nenhuma linha copiada

Para migrar uma visualização, defina `table-type` como `all` ou `view`. Para ter mais informações, consulte Especificar a seleção de tabelas e as regras de transformação no console.

As origens compatíveis com visualizações incluem o seguinte.

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- IBM Db2 LUW
- SAP Adaptive Server Enterprise (ASE)

Solução de problemas com o Microsoft SQL Server

A seguir, você aprenderá sobre a solução de problemas específicos para utilizar o AWS DMS com bancos de dados Microsoft SQL Server.

Tópicos

- Erros ao capturar alterações de banco de dados SQL Server
- Colunas de identidade ausentes
- Erro: o SQL Server não é compatível com publicações
- As alterações não são exibidas no destino
- Tabela não uniforme mapeada entre partições

Erros ao capturar alterações de banco de dados SQL Server

Os erros durante a captura de dados de alteração (CDC) podem indicar que um dos pré-requisitos não foi atendido. Por exemplo, o pré-requisito que mais passa despercebido é o backup completo do banco de dados. O log de tarefas indica essa omissão com o seguinte erro:

```
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)
```

Revise os pré-requisitos listados para a utilização do SQL Server como origem em Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS.

Colunas de identidade ausentes

O AWS DMS não é compatível com colunas de identidade quando você cria um esquema de destino. Você deve adicioná-las após a conclusão do carregamento inicial.

Erro: o SQL Server não é compatível com publicações

O erro a seguir é gerado quando você utiliza o SQL Server Express como endpoint de origem:

```
RetCode: SQL_ERROR SqlState: HY000 NativeError: 21106
Message: This edition of SQL Server does not support publications.
```

No momento, o AWS DMS não é compatível com o SQL Server Express como origem ou destino.

As alterações não são exibidas no destino

O AWS DMS requer que um banco de dados SQL Server de origem esteja no modelo de recuperação de dados "FULL" ou "BULK LOGGED" para capturar as alterações de maneira consistente. O modelo 'SIMPLE' não é compatível.

O modelo de recuperação SIMPLE registra o mínimo de informações necessárias para permitir que os usuários recuperem seus bancos de dados. Todas as entradas de log inativas são truncadas automaticamente quando um ponto de verificação ocorre.

Todas as operações ainda são registradas em log. No entanto, assim que ocorre um ponto de verificação, o log é automaticamente truncado. Esse truncamento significa que o log fica disponível para reutilização e as entradas mais antigas do log podem ser substituídas. Quando as entradas do log são substituídas, as alterações não podem ser capturadas. Esse problema é o motivo pelo qual o AWS DMS não é compatível com o modelo de recuperação de dados SIMPLE. Para obter informações sobre outros pré-requisitos necessários para utilizar o SQL Server como origem, consulte Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS.

Tabela não uniforme mapeada entre partições

Durante a captura de dados de alteração (CDC), a migração de uma tabela com uma estrutura especializada será suspensa quando o AWS DMS não puder executar corretamente a CDC na tabela. Mensagens como estas são emitidas:

```
[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)
```

Ao executar a CDC em tabelas do SQL Server, o AWS DMS analisa os tlogs do SQL Server. Em cada registro do tlog, o AWS DMS analisa os valores hexadecimais contendo dados das colunas que foram inseridas, atualizadas ou excluídas durante uma alteração.

Para analisar o registo hexadecimal, o AWS DMS lê os metadados da tabela das tabelas do sistema do SQL Server. Essas tabelas do sistema identificam o que são as colunas da tabela especialmente estruturadas e revelam algumas de suas propriedades internas, como "xoffset" e "posição de bit nulo".

O AWS DMS espera que os metadados sejam os mesmos para todas as partições brutas da tabela. Mas, em alguns casos, tabelas especialmente estruturadas não têm os mesmos metadados em todas as partições. Nesses casos, o AWS DMS pode suspender a CDC nessa tabela para evitar a análise incorreta das alterações e o fornecimento de dados incorretos ao destino. As soluções alternativas incluem o seguinte:

- Se a tabela tiver um índice clusterizado, execute uma recompilação do índice.
- Se a tabela não tiver um índice clusterizado, adicione um índice clusterizado à tabela (você poderá descartá-lo mais tarde se quiser).

Solução de problemas com o Amazon Redshift

A seguir, você aprenderá sobre a solução de problemas específicos ao utilizar o AWS DMS com bancos de dados do Amazon Redshift.

Tópicos

Carga em um cluster do Amazon Redshift em uma região da AWS diferente

Não é possível carregar em um cluster do Amazon Redshift em uma região da AWS diferente da instância de replicação do AWS DMS. O DMS exige que a instância de replicação e o cluster do Amazon Redshift estejam na mesma região.

Erro: Relation "awsdms_apply_exceptions" already exists

O erro "Relation 'awsdms_apply_exceptions' already exists" costuma ocorrer quando um endpoint do Redshift é especificado como endpoint do PostgreSQL. Por corrigir esse problema, modifique o endpoint e altere Target engine para "redshift".

Erros com tabelas cujos nomes começam com "awsdms_changes"

Mensagens de erro de tabelas com nomes que começam com "awsdms_changes" podem ocorrer quando duas tarefas que estão tentando carregar dados no mesmo cluster do Amazon Redshift são executadas simultaneamente. Devido à forma como tabelas temporárias são nomeadas, tarefas simultâneas podem entrar em conflito ao atualizar a mesma tabela.

Ver tabelas em cluster com nomes como dms.awsdms_changes000000000XXXX

O AWS DMS cria tabelas temporárias quando os dados estão sendo carregados de arquivos armazenados no S3. O nome dessas tabelas temporárias tem o prefixo `dms.awsdms_changes`. Elas são necessárias para o AWS DMS armazenar dados quando eles são carregados pela primeira vez e antes de serem colocados na tabela de destino final.

Permissões necessárias para trabalhar com o Amazon Redshift

Para utilizar o AWS DMS com o Amazon Redshift, a conta de usuário utilizada para acessar o Amazon Redshift deve ter as seguintes permissões:

- CRUD (escolher, inserir, atualizar, excluir)
- Carga em massa
- Criar, alterar, descartar (se necessário pela definição da tarefa)

Para ver todos os pré-requisitos necessários para usar o Amazon Redshift como destino, consulte Utilizar um banco de dados Amazon Redshift como destino do AWS Database Migration Service.

Solução de problemas com o MySQL do Amazon Aurora

A seguir, você poderá aprender sobre a solução de problemas específicos para utilizar o AWS DMS com os bancos de dados do MySQL do Amazon Aurora.

Tópicos

- Erro: campos CHARACTER SET UTF8 terminados por ',' inseridos em "" linhas terminadas em '\n'

Erro: campos CHARACTER SET UTF8 terminados por ',' inseridos em "" linhas terminadas em '\n'

Se você estiver utilizando o MySQL do Amazon Aurora como destino, poderá ver um erro como o seguinte nos logs. Esse tipo de erro geralmente indica que você tem ANSI_QUOTES como parte do parâmetro SQL_MODE. Ter ANSI_QUOTES como parte do parâmetro SQL_MODE faz com que aspas duplas sejam tratadas como aspas simples, o que pode criar problemas ao executar uma tarefa.

Por corrigir esse erro, remova ANSI_QUOTES do parâmetro SQL_MODE.

```
2016-11-02T14:23:48 [TARGET_LOAD ]E: Load data sql statement. load data local infile
"/rdsdbdata/data/tasks/7XO4FJHCVON7TYTLQ6RX3CQHDU/data_files/4/LOAD000001DF.csv" into
 table
`VOSPUSER`.`SANDBOX_SRC_FILE` CHARACTER SET UTF8 fields terminated by ','
enclosed by '"' lines terminated by '\n'( `SANDBOX_SRC_FILE_ID`,`SANDBOX_ID`,
`FILENAME`,`LOCAL_PATH`,`LINES_OF_CODE`,`INSERT_TS`,`MODIFIED_TS`,`MODIFIED_BY`,
`RECORD_VER`,`REF_GUID`,`PLATFORM_GENERATED`,`ANALYSIS_TYPE`,`SANITIZED`,`DYN_TYPE`,
`CRAWL_STATUS`,`ORIG_EXEC_UNIT_VER_ID` ) ; (provider_syntax_manager.c:2561)
```

Solução de problemas com o SAP ASE

A seguir, você poderá aprender sobre a solução de problemas específicos para utilizar o AWS DMS com bancos de dados SAP ASE.

Erro: as colunas LOB têm valores NULL quando a origem tem um índice exclusivo composto com valores NULL

Ao usar o SAP ASE como origem com tabelas configuradas com um índice exclusivo composto que permite valores NULL, os valores LOB podem não serem migrados durante a replicação contínua. Esse comportamento geralmente é o resultado de ANSI_NULL definido como 1 por padrão no cliente da instância de replicação do DMS.

Para garantir que os campos LOB migrem corretamente, inclua a Configuração de endpoint `'AnsiNull=0'` no endpoint de origem do AWS DMS da tarefa.

Solução de problemas com o IBM Db2

A seguir, você poderá aprender sobre a solução de problemas específicos para utilizar o AWS DMS com bancos de dados IBM Db2.

Erro: a retomada a partir do timestamp não é uma tarefa compatível

Para replicação contínua (CDC), se você planejar iniciar a replicação a partir de um timestamp específico, defina o atributo de conexão do `StartFromContext` com o timestamp requerido. Para obter mais informações, consulte Configurações de endpoint ao utilizar o Db2 LUW. A configuração `StartFromContext` com o timestamp necessário evita o seguinte problema:

```
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.
```

Solução de problemas de latência no AWS Database Migration Service

Esta seção fornece uma visão geral das causas comuns de latência de tarefas do AWS DMS durante a fase de replicação contínua (CDC). O AWS DMS replica os dados de forma assíncrona. Latência é o atraso entre o momento em que uma alteração foi confirmada na origem e o momento em que a alteração foi replicada no destino. A latência pode ser causada por uma configuração incorreta dos componentes de replicação, como os seguintes:

- Endpoint de origem ou fonte de dados
- Endpoint de destino ou fonte de dados
- Instâncias de replicação
- A rede entre esses componentes

É recomendável utilizar uma migração de teste como prova de conceito para coletar informações sobre a replicação. É possível utilizar essas informações para ajustar a configuração de replicação para minimizar a latência. Para obter informações sobre como executar uma migração de prova de conceito, consulte Execução de uma prova de conceito.

Tópicos

Tipos de latência da CDC

Esta seção contém os tipos de latência de replicação que podem ocorrer durante a CDC.

Latência de origem

O intervalo, em segundos, entre o tempo de confirmação do último evento capturado no endpoint de origem e o timestamp atual do sistema da instância de replicação. Você pode monitorar a latência entre a fonte de dados e sua instância de replicação usando a `CDCLatencySource` CloudWatch métrica. Uma métrica `CDCLatencySource` alta indica que o processo de captura de alterações da origem está atrasado. Por exemplo, se a aplicação confirmar uma inserção na origem às 10h, e o AWS DMS consumir a alteração às 10h02, a métrica `CDCLatencySource` será de 120 segundos.

Para obter informações sobre CloudWatch métricas paraAWS DMS, consulteMétricas de tarefas de replicação.

Latência de destino

O intervalo, em segundos, entre a hora de confirmação na origem do primeiro evento aguardando confirmação no destino e o timestamp atual da instância de replicação do DMS. Você pode monitorar a latência entre as confirmações na fonte de dados e seu destino de dados usando a `CDCLatencyTarget` CloudWatch métrica. Isso significa que `CDCLatencyTarget` inclui qualquer atraso na leitura na origem. Como resultado, `CDCLatencyTarget` é sempre maior ou igual a `CDCLatencySource`.

Por exemplo, se a aplicação confirmar uma inserção na origem às 10h, e o AWS DMS consumir a alteração às 10h02 e gravá-la no destino às 10h05, a métrica `CDCLatencyTarget` será de 300 segundos.

Causas comuns de latência da CDC

Esta seção contém as causas de latência de replicação que podem ocorrer durante a CDC.

Tópicos

- Recursos de endpoints
- Recursos de instâncias de replicação
- Velocidade e largura de banda da rede
- Configuração do DMS
- Cenários de replicação

Recursos de endpoints

Os seguintes fatores afetam significativamente o desempenho e a latência da replicação:

- Configurações dos bancos de dados de origem e de destino
- Tamanho da instância
- Datastores de origem e de destino subprovisionados ou mal configurados

Para identificar as causas da latência causada por problemas de endpoint em fontes e destinos AWS hospedados, monitore as seguintes métricas: CloudWatch

- `FreeMemory`
- `CPUUtilization`
- Métricas de throughput e de E/S, como `WriteIOPS`, `WriteThroughput` ou `ReadLatency`
- Métricas de volume de transações, como `CDCIncomingChanges`.

Para obter informações sobre CloudWatch métricas de monitoramento, consulteMétricas do AWS Database Migration Service.

Recursos de instâncias de replicação

Os recursos de instâncias de replicação são essenciais para a replicação, e você deve garantir que não haja gargalos de recursos, pois eles podem levar à latência na origem e no destino.

Para identificar gargalos de recursos na instância de replicação, verifique o seguinte:

- CloudWatch Métricas críticas, como CPU, memória, E/S por segundo e armazenamento, não estão apresentando picos ou valores consistentemente altos.
- A instância de replicação está dimensionada de forma adequada para a workload. Para obter informações sobre como determinar o tamanho correto da instância de replicação, consulte Seleção do melhor tamanho para uma instância de replicação.

Velocidade e largura de banda da rede

A largura de banda da rede é um fator que afeta a transmissão de dados. Para analisar o desempenho da rede da replicação, siga um destes procedimentos:

- Verifique as métricas `ReadThroughput` e `WriteThroughput` no nível da instância. Para obter informações sobre CloudWatch métricas de monitoramento, consulteMétricas do AWS Database Migration Service.
- Utilize a AMI de apoio diagnóstico do AWS DMS. Se a AMI de apoio diagnóstico não estiver disponível na região, será possível baixá-la de qualquer região compatível e copiá-la na sua região para executar a análise de rede. Para obter informações sobre a AMI de apoio diagnóstico, consulte Trabalhando com o suporte AWS DMS de diagnóstico AMI.

A CDC no AWS DMS é de thread único para garantir a consistência dos dados. Como resultado, é possível determinar o volume de dados compatível com a rede calculando a taxa de transferência

de dados de thread único. Por exemplo, se a tarefa se conectar à origem utilizando uma rede de 100 Mbps (megabits por segundo), a replicação terá uma alocação de largura de banda máxima teórica de 12,5 MBps (megabytes por segundo). Isso equivale a 45 gigabits por hora. Se a taxa de geração do log de transações na origem for maior que 45 gigabits por hora, isso significará que a tarefa tem latência de CDC. Para uma rede de 100 MBps, essas taxas são máximas teóricas. Outros fatores, como tráfego de rede e sobrecarga de recursos na origem e no destino, reduzem a largura de banda real disponível.

Configuração do DMS

Esta seção contém as configurações de replicação recomendadas que podem ajudar a reduzir a latência.

- Configurações de endpoint: as configurações de endpoints de origem e de destino podem fazer com que a instância de replicação tenha um desempenho inadequado. As configurações de endpoint que ativam recursos com consumo excessivo afetarão o desempenho. Por exemplo, para um endpoint Oracle, desabilitar LogMiner e usar o Binary Reader melhora o desempenho, pois LogMiner consome muitos recursos. A configuração de endpoint a seguir melhora o desempenho de um endpoint do Oracle:

  ```
  useLogminerReader=N;useBfile=Y
  ```

  Para obter mais informações sobre configurações de endpoint, consulte a documentação do mecanismo de endpoint de origem e de destino no tópico Como trabalhar com endpoints do AWS DMS.
- Configurações de tarefas: algumas configurações de tarefas para o cenário de replicação específico podem fazer com que a instância de replicação tenha um desempenho inadequado. Por exemplo, o AWS DMS utiliza o modo de aplicação transacional por padrão (`BatchApplyEnabled=false`) para CDC em todos os endpoints, exceto no Amazon Redshift. No entanto, para origens com um grande número de alterações, a definição de `BatchApplyEnabled` como `true` pode melhorar o desempenho.

  Para obter mais informações sobre as configurações de tarefas, consulte Especificando configurações de tarefas para tarefas do AWS Database Migration Service.
- Posição inicial de uma tarefa somente de CDC: iniciar uma tarefa somente de CDC em uma posição ou timestamp no passado iniciará a tarefa com latência da origem da CDC aumentada. Dependendo do volume de alterações na origem, a latência da tarefa demorará algum tempo para diminuir.

- Configurações de LOB: tipos de dados de objetos grandes podem prejudicar o desempenho da replicação devido à forma como o AWS DMS replica dados binários grandes. Para obter informações, consulte os tópicos a seguir:
  - Configurando o suporte LOB para bancos de dados de origem em uma tarefa AWS DMS
  - Migração de objetos binários grandes (LOBs).

Cenários de replicação

Esta seção descreve cenários específicos de replicação e como eles podem afetar a latência.

Tópicos

- Interromper uma tarefa por um período estendido
- Alterações armazenadas em cache
- Replicação entre regiões

Interromper uma tarefa por um período estendido

Ao interromper uma tarefa, o AWS DMS salva a posição do último log de transações lido da origem. Quando você retoma a tarefa, o DMS tenta continuar lendo a partir da mesma posição do log de transações. A retomada de uma tarefa após várias horas ou dias faz com que a latência da origem da CDC aumente até que o DMS conclua o consumo do backlog de transações.

Alterações armazenadas em cache

Alterações armazenadas em cache são alterações que a aplicação grava na fonte de dados enquanto o AWS DMS executa a fase de replicação de carga máxima. O DMS não aplica essas alterações até que a fase de carga máxima seja concluída e a fase de CDC seja iniciada. Para uma origem com um grande número de transações, as alterações armazenadas em cache demoram mais para serem aplicadas, portanto, a latência da origem aumenta quando a fase de CDC é iniciada. É recomendável executar a fase de carga máxima quando os volumes de transações estiverem baixos para minimizar o número de alterações em cache.

Replicação entre regiões

A localização dos endpoints do DMS ou da instância de replicação em diferentes regiões da AWS aumenta a latência da rede. Isso aumenta a latência da replicação. Para obter o melhor desempenho, localize o endpoint de origem, o endpoint de destino e a instância de replicação na mesma região da AWS.

Solução de problemas de latência

Esta seção contém as etapas da solução de problemas de latência da replicação.

Para solucionar problemas de latência, faça o seguinte:

- Primeiro, determine o tipo e a quantidade de latência da tarefa. Verifique a seção Estatísticas da tabela da tarefa no console do DMS ou na CLI. Se os contadores estiverem mudando, a transmissão de dados estará em andamento. Verifique as métricas `CDCLatencySource` e `CDCLatencyTarget` em conjunto para determinar se há um gargalo durante a CDC.
- Se as métricas `CDCLatencySource` ou `CDCLatencyTarget` altas indicarem um gargalo na replicação, verifique o seguinte:
  - Se `CDCLatencySource` estiver alta e `CDCLatencyTarget` estiver igual a `CDCLatencySource`, isso indicará que há um gargalo no endpoint de origem, e o AWS DMS está gravando dados no destino sem problemas. Consulte Solução de problemas de latência da origem a seguir.
  - Se `CDCLatencySource` for baixa e `CDCLatencyTarget` for alta, isso indica que há um gargalo no endpoint de destino e que o AWS DMS está lendo os dados da origem sem problemas. Consulte Solução de problemas de latência no destino a seguir.
  - Se `CDCLatencySource` for alta `CDCLatencyTarget` for significativamente mais alta que`CDCLatencySource`, isso indica gargalos nas leituras de origem e nas gravações de destino. Investigue primeiro a latência da origem e investigue a latência de destino.

Para obter informações sobre como monitorar métricas das tarefas do DMS, consulte Monitoramento de tarefas do AWS DMS.

Solução de problemas de latência da origem

Os tópicos a seguir descrevem cenários de replicação específicos para tipos de endpoint de origem.

Tópicos

- Solução de problemas do endpoint do Oracle
- Solução de problemas do endpoint do MySQL
- Solução de problemas do endpoint do PostgreSQL
- Solução de problemas do endpoint do SQL Server

Solução de problemas do endpoint do Oracle

Esta seção contém cenários de replicação específicos do Oracle.

Leitura da origem pausada

O AWS DMS interrompe a leitura de uma origem do Oracle nos cenários a seguir. Esse comportamento é por projeto. É possível investigar as causas disso utilizando o log de tarefas. Procure mensagens semelhantes às seguintes no log de tarefas. Para obter mais informações sobre como trabalhar com o log, consulte o Visualização e gerenciamento dos logs de tarefas do AWS.

- Mensagem do SORTER: indica que o DMS está armazenando transações em cache na instância de replicação. Para obter mais informações, consulte Mensagem de SORTER no log de tarefas a seguir.
- Logs de tarefas de depuração: se o DMS interromper o processo de leitura, a tarefa gravará repetidamente a seguinte mensagem nos logs de tarefas de depuração, sem alterar o campo de contexto ou o timestamp:
  - 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
 '000000000F2CECAA01000010005A8F500000275016C0000000000000F2CEC58'
xid [000014e06411d996] timestamp '2021-08-12 09:20:32' thread 1
 (oracdc_reader.c:2269)
```

- O AWS DMS registra em log a seguinte mensagem para cada nova operação do redo log ou de arquivamento.

```
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)
```

Se a origem tiver novas operações de refazer ou de arquivamento, e o AWS DMS não estiver gravando essas mensagens no log, isso significa que a tarefa não está processando eventos.

Alta geração de redo

Se a tarefa estiver processando redo logs e de arquivamento, mas a latência da origem permanecer alta, tente identificar a taxa de geração de redo log e os padrões de geração. Se você tiver um alto nível de geração de redo log, isso aumentará a latência da origem, pois a tarefa lê todos os redo logs e de arquivamento para buscar as alterações relacionadas às tabelas replicadas.

Para determinar a taxa de geração de refazer, utilize as consultas a seguir.

- Taxa de geração de refazer por dia:

```
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;
```

- Taxa de geração de refazer por hora:

```
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 ;
```

Para solucionar problemas de latência nesse cenário, verifique o seguinte:

- Verifique a largura de banda da rede e o desempenho de thread único da replicação para garantir que a rede subjacente possa suportar a taxa de geração de refazer de origem. Para obter informações sobre como a largura de banda da rede pode afetar o desempenho da replicação, consulte o artigo Velocidade e largura de banda da rede anterior.
- Verifique se você configurou o registro em log suplementar corretamente. Evite registros em log adicionais na origem, como a ativação do registro em log em todas as colunas de uma tabela. Para

obter informações sobre a configuração do registro em log suplementar, consulte Configuração de registro em log suplementar.

- Verifique se está utilizando a API correta para ler os redo logs ou de arquivamento. Você pode usar o Oracle LogMiner ou o AWS DMS Binary Reader. Enquanto LogMiner lê os redo logs on-line e os arquivos de redo log arquivados, o Binary Reader lê e analisa diretamente os arquivos de redo log brutos. Como resultado, o Binary Reader tem desempenho melhor. É recomendável utilizar o Binary Reader se a geração do redo log for superior a 10 GB/hora. Para ter mais informações, consulte Usando Oracle LogMiner ou AWS DMS Binary Reader para CDC.
- Verifique se você definiu `ArchivedLogsOnly` como `Y`. Se essa configuração de endpoint estiver definida, o AWS DMS lerá os redo logs arquivados. Isso aumenta a latência da origem, pois o AWS DMS espera que o redo log on-line seja arquivado antes da leitura. Para obter mais informações, consulte ArchivedLogsOnly.
- Se a origem do Oracle utilizar o gerenciamento automático de armazenamento (ASM), consulte Armazenando REDO no Oracle ASM ao usar o Oracle como fonte para AWS DMS para obter informações sobre como configurar adequadamente o datastore. Também é possível otimizar ainda mais o desempenho de leitura utilizando o atributo de conexão adicional (ECA) `asmUsePLSQLArray`. Para obter informações sobre como utilizar o `asmUsePLSQLArray`, consulte Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.

Solução de problemas do endpoint do MySQL

Esta seção contém cenários de replicação específicos do MySQL. O AWS DMSexamina o log binário do MySQL periodicamente para replicar as alterações. Esse processo pode aumentar a latência nos seguintes cenários:

Tópicos

- Transações de execução prolongada na origem
- Alta workload na origem
- Contenção do log binário

Transações de execução prolongada na origem

Como o MySQL só grava transações confirmadas no log binário, as transações de execução prolongada causam picos de latência proporcionais ao tempo de execução da consulta.

Para identificar transações de execução prolongada, utilize a consulta a seguir ou utilize o log de consultas lentas:

```
SHOW FULL PROCESSLIST;
```

Para obter informações sobre como utilizar o log de consultas lentas, consulte O log de consultas lentas na Documentação do MySQL.

Para evitar picos de latência em transações de execução prolongada, reestruture as transações de origem para reduzir o tempo de execução de consultas ou aumentar a frequência de confirmação.

Alta workload na origem

Como a CDC do DMS é de um único thread, um grande número de transações pode aumentar a latência da origem. Para identificar se a latência da origem se deve a uma workload pesada, compare o número e o tamanho dos logs binários gerados durante o período de latência com os logs gerados antes da latência. Para verificar os logs binários e o status do thread da CDC do DMS, utilize as seguintes consultas:

```
SHOW BINARY LOGS;
SHOW PROCESSLIST;
```

Para obter mais informações sobre os estados dos threads de despejo de logs binários da CDC, consulte Estados dos threads da origem de replicação.

É possível determinar a latência comparando a posição mais recente do log binário gerado na origem com o evento que o DMS está processando no momento. Para identificar o log binário mais recente na origem, faça o seguinte:

- Ative os logs de depuração no componente SOURCE_CAPTURE.
- Recupere o log binário de processamento do DMS e os detalhes de posição dos logs de depuração da tarefa.
- Utilize a consulta a seguir para identificar o log binário mais recente na origem:

  ```
  SHOW MASTER STATUS;
  ```

Para otimizar ainda mais o desempenho, ajuste o `EventsPollInterval`. Por padrão, o DMS pesquisa o log binário a cada cinco segundos, mas é possível melhorar o desempenho reduzindo

esse valor. Para obter mais informações sobre a configuração de `EventsPollInterval`, consulte Configurações de endpoint ao usar o MySQL como fonte para AWS DMS.

Contenção do log binário

Ao migrar várias tabelas com uma grande quantidade de dados, é recomendável dividir as tabelas em tarefas separadas para o MySQL 5.7.2 ou posterior. No MySQL 5.7.2 e posterior, o thread de despejo mestre cria menos contenções de bloqueio e melhora o throughput. Como resultado, o encadeamento de despejo não bloqueia mais o log binário sempre que ele lê um evento. Isso significa que vários threads de despejo podem ler o arquivo de log binário simultaneamente. Isso também significa que os threads de despejo podem ler o log binário enquanto os clientes gravam nele. Para obter mais informações sobre threads de despejo, consulte Threads de replicação e as Notas de release do MySQL 5.7.2.

Para melhorar o desempenho da replicação das versões de origem do MySQL anteriores à 5.7.2, tente consolidar tarefas com componentes de CDC.

Solução de problemas do endpoint do PostgreSQL

Esta seção contém cenários de replicação específicos do PostgreSQL.

Tópicos

- Transações de execução prolongada na origem
- Alta workload na origem
- Alto throughput de rede
- Derrame arquivos no Aurora PostgreSQL

Transações de execução prolongada na origem

Quando há transações de execução prolongada no banco de dados de origem, como algumas milhares de inserções em uma única transação, os contadores de eventos e transações da CDC do DMS não aumentam até que a transação seja concluída. Esse atraso pode causar problemas de latência que é possível medir utilizando a métrica `CDCLatencyTarget`.

Para revisar transações de execução prolongada, siga um destes procedimentos:

- Utilize a visualização `pg_replication_slots`. Se o valor de `restart_lsn` não estiver sendo atualizado, é provável que o PostgreSQL não possa liberar logs de gravação antecipada (WALs)

devido a transações ativas de execução prolongada. Para obter informações sobre a visualização `pg_replication_slots`, consulte pg_replication_slots na Documentação do PostgreSQL 15.4.

- Utilize a consulta a seguir para retornar uma lista de todas as consultas ativas no banco de dados, junto com as informações relacionadas:

```
SELECT pid, age(clock_timestamp(), query_start), usename, query
FROM pg_stat_activity WHERE query != '<IDLE>'
AND query NOT ILIKE '%pg_stat_activity%'
ORDER BY query_start desc;
```

  Nos resultados da consulta, o campo `age` mostra a duração ativa de cada consulta, que pode ser utilizada para identificar consultas de execução prolongada.

Alta workload na origem

Se o PostgreSQL de origem tiver uma workload alta, verifique o seguinte para reduzir a latência:

- É possível experimentar alta latência ao utilizar o plug-in `test_decoding` ao migrar um subconjunto de tabelas do banco de dados de origem com um alto valor de transações por segundo (TPS). Isso ocorre porque o plug-in `test_decoding` envia todas as alterações do banco de dados para a instância de replicação que o DMS filtra com base no mapeamento de tabelas da tarefa. Eventos para tabelas que não fazem parte do mapeamento de tabelas da tarefa podem aumentar a latência da origem.
- Verifique o throughput de TPS utilizando um dos métodos a seguir.
  - Para fontes do Aurora PostgreSQL, use a métrica. `CommitThroughput` CloudWatch
  - Para o PostgreSQL executado no Amazon RDS ou on-premises, utilize a seguinte consulta com um cliente PSQL versão 11 ou superior (pressione **`enter`** durante a consulta para avançar os resultados):

```
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";
```

- Para reduzir a latência ao utilizar o plug-in `test_decoding`, considere utilizar o plug-in `pglogical` em vez disso. Ao contrário do plug-in `test_decoding`, o plug-in `pglogical` filtra as alterações do log gravação antecipada (WAL) na origem e envia apenas as alterações relevantes

para a instância de replicação. Para obter informações sobre como utilizar o plug-in `pglogical` com o AWS DMS, consulte Configurar o plug-in pglogical.

Alto throughput de rede

A replicação pode ter alta utilização de largura de banda de rede ao utilizar o plug-in `test_decoding`, especialmente durante transações de alto volume. Isso ocorre porque o plug-in `test_decoding` processa as alterações e as converte em um formato legível por humanos que é maior que o formato binário original.

Para melhorar o desempenho, considere utilizar o plug-in `pglogical`, que é um plug-in binário. Ao contrário do plug-in `test_decoding`, o plug-in `pglogical` gera uma saída em formato binário, resultando em alterações do fluxo compactado de log de gravação antecipada (WAL).

Derrame arquivos no Aurora PostgreSQL

No PostgreSQL versão 13 e superior, o parâmetro determina `logical_decoding_work_mem` a alocação de memória para decodificação e streaming. Para obter mais informações sobre o `logical_decoding_work_mem` parâmetro, consulte Consumo de recursos no PostgreSQL na documentação do PostgreSQL 13.13.

A replicação lógica acumula alterações em todas as transações na memória até que essas transações sejam confirmadas. Se a quantidade de dados armazenados em todas as transações exceder a quantidade especificada pelo parâmetro do banco de dados`logical_decoding_work_mem`, o DMS transferirá os dados da transação para o disco para liberar memória para novos dados de decodificação.

Transações de longa execução, ou muitas subtransações, podem fazer com que o DMS consuma mais memória de decodificação lógica. Esse aumento no uso de memória faz com que o DMS crie arquivos vazados no disco, o que causa alta latência na fonte durante a replicação.

Para reduzir o impacto de um aumento na carga de trabalho de origem, faça o seguinte:

- Reduza as transações de longa duração.
- Reduza o número de subtransações.
- Evite realizar operações que gerem uma grande quantidade de registros de log, como excluir ou atualizar uma tabela inteira em uma única transação. Em vez disso, execute operações em lotes menores.

Você pode usar as seguintes CloudWatch métricas para monitorar a carga de trabalho na fonte:

- `TransactionLogsDiskUsage`: o número de bytes atualmente ocupados pelo WAL lógico. Esse valor aumenta monotonicamente se os slots de replicação lógica não conseguirem acompanhar o ritmo de novas gravações ou se alguma transação de longa execução impedir a coleta de lixo de arquivos antigos.
- `ReplicationSlotDiskUsage`: a quantidade de espaço em disco que os slots de replicação lógica usam atualmente.

Você pode reduzir a latência da fonte ajustando o `logical_decoding_work_mem` parâmetro. O valor padrão para esse parâmetro é 64 MB. Esse parâmetro limita a quantidade de memória usada por cada conexão lógica de replicação de streaming. Recomendamos definir um `logical_decoding_work_mem` valor significativamente maior do que o `work_mem` valor para reduzir a quantidade de alterações decodificadas que o DMS grava no disco.

Recomendamos que você verifique periodicamente se há vazamentos de arquivos, especialmente durante períodos de intensa atividade de migração ou latência. Se o DMS estiver criando um número significativo de arquivos vazados, isso significa que a decodificação lógica não está operando de forma eficiente, o que pode aumentar a latência. Para mitigar isso, aumente o valor do `logical_decoding_work_mem` parâmetro.

Você pode verificar o estouro atual da transação com a `aurora_stat_file` função. Para obter mais informações, consulte Como ajustar a memória de trabalho para decodificação lógica no Guia do desenvolvedor do Amazon Relational Database Service.

Solução de problemas do endpoint do SQL Server

Esta seção contém cenários de replicação específicos do SQL Server. Para determinar quais alterações devem ser replicadas do SQL Server, o AWS DMS lê os logs de transações e executa verificações periódicas no banco de dados de origem. A latência da replicação geralmente resulta no controle de utilização dessas verificações pelo SQL Server devido a restrições de recursos. Também pode resultar de um aumento significativo no número de eventos gravados no log de transações em um curto espaço de tempo.

Tópicos

- Reconstruções de índices
- Transações grandes

- Intervalo de pesquisa do MS-CDC configurado incorretamente para o Amazon RDS SQL Server
- Várias tarefas da CDC replicando no mesmo banco de dados de origem

Reconstruções de índices

Quando o SQL Server reconstrói um índice grande, ele utiliza uma única transação. Isso gera muitos eventos e pode utilizar uma grande quantidade de espaço de log se o SQL Server reconstruir vários índices ao mesmo tempo. Quando isso ocorrer, é possível esperar breves picos de replicação. Se a origem do SQL Server tiver picos sustentados de log, verifique o seguinte:

- Primeiro, verifique o período de tempo dos picos de latência usando as `CDCLatencySource` CloudWatch métricas `CDCLatencySource` e ou verificando as mensagens do Throughput Monitoring nos registros de tarefas. Para obter informações sobre CloudWatch métricas paraAWS DMS, consulteMétricas de tarefas de replicação.
- Verifique se o tamanho dos logs de transações ativos ou os backups de logs aumentou durante o pico de latência. Verifique também se um trabalho de manutenção ou de reconstrução foi executado durante esse período. Para obter informações sobre como verificar o tamanho do log de transações, consulte Monitorar a utilização do espaço de log na Documentação técnica do SQL Server.
- Verifique se o plano de manutenção segue as práticas recomendadas do SQL Server. Para obter informações sobre as práticas recomendadas de manutenção do SQL Server, consulte Estratégia de manutenção de índices na Documentação técnica do SQL Server.

Para corrigir problemas de latência durante as reconstruções de índices, experimente o seguinte:

- Utilize o modelo de recuperação `BULK_LOGGED` para reconstruções off-line para reduzir os eventos que uma tarefa precisa processar.
- Se possível, interrompa a tarefa durante as reconstruções de índices. Ou tente programar as reconstruções de índices fora do horário de pico para mitigar o impacto de um pico de latência.
- Tente identificar os gargalos de recursos que estão retardando as leituras do DMS, como latência do disco ou throughput de E/S, e resolvê-los.

Transações grandes

Transações com muitos eventos ou transações de execução prolongada fazem com que o log de transações aumente. Isso faz com que as leituras do DMS demorem mais, resultando em latência. Isso é semelhante ao efeito que as reconstruções de índices têm no desempenho da replicação.

É possível que seja difícil identificar esse problema se você não estiver familiarizado com a workload típica no banco de dados de origem. Para solucionar esse problema, faça o seguinte:

- Primeiro, identifique o tempo em que a latência aumentou usando as `WriteThroughput` CloudWatch métricas `ReadThroughput` e ou verificando as mensagens do Throughput Monitoring nos registros de tarefas.
- Verifique se há alguma consulta de execução prolongada no banco de dados de origem durante o pico de latência. Para obter informações sobre consultas de execução prolongada, consulte Solucionar problemas de consultas de execução lenta no SQL Server na Documentação técnica do SQL Server.
- Verifique se o tamanho dos logs de transações ativos ou dos backups de logs aumentou. Para obter mais informações, consulte Monitorar a utilização do espaço de log na Documentação técnica do SQL Server.

Para solucionar esse problema, faça o seguinte:

- A melhor solução é reestruturar as transações no lado da aplicação para que sejam concluídas rapidamente.
- Se não for possível reestruturar as transações, uma solução alternativa de curto prazo é verificar se há gargalos de recursos, como esperas de disco ou contenção de CPU. Se você encontrar gargalos no banco de dados de origem, poderá reduzir a latência aumentando os recursos de disco, CPU e memória do banco de dados de origem. Isso reduz a disputa por recursos do sistema, permitindo que as consultas do DMS sejam concluídas mais rapidamente.

Intervalo de pesquisa do MS-CDC configurado incorretamente para o Amazon RDS SQL Server

A configuração incorreta de um intervalo de sondagem nas instâncias do Amazon RDS pode fazer com que o log de transações aumente. Isso ocorre porque a replicação impede o truncamento do log. Embora as tarefas em execução possam continuar a replicação com latência mínima, interromper e retomar tarefas, ou iniciar tarefas somente de CDC, pode causar falhas nas tarefas. Isso ocorre devido aos tempos limite durante a verificação do log de transações grande.

Para solucionar problemas de um intervalo de sondagem mal configurado, faça o seguinte:

- Verifique se o tamanho do log de transações ativo está aumentando e se a utilização do log está próxima de 100%. Para obter mais informações, consulte Monitorar a utilização do espaço de log na Documentação técnica do SQL Server.
- Verifique se o truncamento do log está atrasado com um `log_reuse_wait_desc value` de `REPLICATION`. Para obter mais informações, consulte O log de transações (SQL Server) na Documentação técnica do SQL Server.

Se você encontrar problemas com qualquer um dos itens da lista anterior, ajuste o intervalo de sondagem do MS-CDC. Para obter informações sobre como ajustar o intervalo de sondagem, consulte Configurações recomendadas ao usar o Amazon RDS for SQL Server como fonte para AWS DMS.

Várias tarefas da CDC replicando no mesmo banco de dados de origem

Durante a fase de carga máxima, é recomendável dividir as tabelas entre tarefas para melhorar o desempenho, separar as tabelas dependentes logicamente e mitigar o impacto de uma falha na tarefa. No entanto, durante a fase de CDC, é recomendável consolidar as tarefas para minimizar as verificações do DMS. Durante a fase CDC, cada tarefa do DMS verifica os logs de transações em busca de novos eventos várias vezes por minuto. Como cada tarefa é executada de forma independente, cada tarefa verifica cada log de transações individualmente. Isso aumenta a utilização do disco e da CPU no banco de dados SQL Server de origem. Como resultado, um grande número de tarefas executadas em paralelo pode fazer com que o SQL Server controle a utilização das leituras do DMS, aumentando a latência.

Poderá ser difícil identificar esse problema se várias tarefas começarem gradualmente. O sintoma mais comum desse problema é que a maioria das verificações de tarefas começa a demorar mais. Isso resulta em maior latência para essas verificações. O SQL Server prioriza algumas das verificações de tarefas, portanto, algumas das tarefas mostram latência normal. Para solucionar esse problema, verifique a métrica `CDCLatencySource` de todas as tarefas. Se algumas tarefas tiverem um aumento de `CDCLatencySource`, enquanto algumas tarefas tiverem uma `CDCLatencySource` baixa, é provável que o SQL Server esteja controlando a utilização das leituras do DMS de algumas das tarefas.

Se o SQL Server estiver controlando a utilização das leituras de tarefas durante a CDC, consolide as tarefas para minimizar o número de verificações do DMS. O número máximo de tarefas que podem se conectar ao banco de dados de origem sem criar contenção depende de fatores, como

a capacidade do banco de dados de origem, a taxa de crescimento do log de transações ou o número de tabelas. Para determinar o número ideal de tarefas para o cenário de replicação, teste a replicação em um ambiente de teste semelhante ao ambiente de produção.

Solução de problemas de latência no destino

Esta seção contém cenários que podem contribuir para a latência no destino.

Tópicos

Indexação de problemas

Durante a fase de CDC, o AWS DMS replica as alterações na origem executando instruções DML (inserir, atualizar e excluir) no destino. Para migrações heterogêneas utilizando o DMS, as diferenças nas otimizações de índice na origem e no destino podem fazer com que as gravações no destino demorem mais. Isso resulta em problemas de latência e de desempenho.

Para solucionar esses problemas, faça o seguinte: Os procedimentos para essas etapas variam para diferentes mecanismos de banco de dados.

- Monitore o tempo de consulta do banco de dados de destino. A comparação do tempo de execução da consulta no destino e na origem pode indicar quais índices precisam ser otimizados.
- Ative o registro em log de consultas de execução lenta.

Para corrigir problemas de indexação de replicações de execução prolongada, faça o seguinte:

- Ajuste os índices nos bancos de dados de origem e de destino para que o tempo de execução da consulta seja semelhante na origem e no destino.
- Compare os índices secundários utilizados nas consultas DML para a origem e o destino. Verifique se o desempenho do DML no destino é comparável ou melhor do que o desempenho do DML na origem.

Observe que o procedimento de otimização de índices é específico para o mecanismo de banco de dados. Não há nenhum recurso do DMS para ajustar os índices de origem e de destino.

Mensagem de SORTER no log de tarefas

Se um endpoint de destino não puder acompanhar o volume de alterações que o AWS DMS grava nele, a tarefa armazena as alterações em cache na instância de replicação. Se o cache aumentar mais do que um limite interno, a tarefa interromperá as leituras de outras alterações na origem. O DMS faz isso para evitar que a instância de replicação fique sem armazenamento ou que a tarefa fique paralisada durante a leitura de um grande volume de eventos pendentes.

Para solucionar esse problema, verifique se há uma mensagem semelhante a uma das seguintes nos CloudWatch registros:

```
[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)
```

Se os logs contiverem uma mensagem semelhante à primeira mensagem, desative qualquer registro em log de rastreamento da tarefa e aumente o armazenamento da instância de replicação. Para obter informações sobre como aumentar o armazenamento de instâncias de replicação, consulte Modificar uma instância de replicação.

Caso os logs contenham uma mensagem semelhante à segunda, faça o seguinte:

- Mova as tabelas com várias transações ou operações DML de execução prolongada para uma tarefa separada, caso elas não tenham nenhuma dependência de outras tabelas na tarefa.
- Aumente as configurações de `MemoryLimitTotal` e de `MemoryKeepTime` para manter a transação por mais tempo na memória. Isso não ajudará se a latência for sustentada, mas poderá ajudar a manter a latência baixa durante intermitências curtas do volume transacional. Para obter informações sobre as configurações dessas tarefas, consulte Configurações de ajuste de processamento de alterações.
- Avalie se é possível utilizar a aplicação em lote à transação definindo `BatchApplyEnabled` como `true`. Para obter mais informações sobre a configuração de `BatchApplyEnabled`, consulte Configurações de tarefa de metadados de destino.

Bloqueio de banco de dados

Se uma aplicação acessar um banco de dados que o AWS DMS está utilizando como destino da replicação, a aplicação poderá bloquear uma tabela que o DMS está tentando acessar. Isso cria uma contenção de bloqueio. Como o DMS grava as alterações no banco de dados de destino na ordem em que elas ocorreram na origem, atrasos na gravação em uma tabela devido a contenções de bloqueio criam atrasos na gravação em todas as tabelas.

Para solucionar esse problema, consulte o banco de dados de destino para verificar se uma contenção de bloqueio está bloqueando as transações de gravação do DMS. Se o banco de dados de destino estiver bloqueando transações de gravação do DMS, execute um ou mais dos seguintes procedimentos:

- Reestruture as consultas para confirmar as alterações com mais frequência.
- Modifique as configurações de tempo limite de bloqueio.
- Particione as tabelas para minimizar as contenções de bloqueio.

Observe que o procedimento de otimização de contenções de bloqueio é específico para o mecanismo de banco de dados. Não há nenhum recurso do DMS para ajustar as contenções de bloqueio.

Pesquisas de LOB lentas

Quando o AWS DMS replica uma coluna de objeto grande (LOB), ele executa uma pesquisa na origem antes de gravar as alterações no destino. Essa pesquisa normalmente não causa nenhuma latência no destino, mas se o banco de dados de origem atrasar a pesquisa devido ao bloqueio, a latência de destino poderá aumentar.

Normalmente, esse problema é difícil de diagnosticar. Para solucionar esse problema, ative a depuração detalhada nos logs de tarefas e compare os timestamps das chamadas de pesquisa de LOB do DMS. Para obter informações sobre como ativar o log de depuração detalhado, consulte Visualização e gerenciamento dos logs de tarefas do AWS.

Para solucionar esse problema, tente o seguinte:

- Melhore o desempenho da consulta SELECT no banco de dados de origem.
- Ajuste as configurações de LOB do DMS. Para obter mais informações sobre como ajustar as configurações de LOB, consulte Migração de objetos binários grandes (LOBs).

Multi-AZ, registro em log e backups de auditoria

Para destinos do Amazon RDS, a latência do destino pode aumentar durante o seguinte:

- Backups
- Depois de ativar várias zonas de disponibilidade (multi-AZ)
- Depois de ativar o registro em log do banco de dados, como logs de auditoria ou de consultas lentas.

Normalmente, esses problemas são difíceis de diagnosticar. Para solucionar esses problemas, monitore a latência em busca de picos periódicos durante as janelas de manutenção do Amazon RDS ou períodos de carga pesada do banco de dados.

Para solucionar esses problemas, tente o seguinte:

- Se possível, durante a migração de curto prazo, desative o multi-AZ, os backups ou o registro em log.
- Reprograme as janelas de manutenção para períodos de baixa atividade.

Como trabalhar com scripts de suporte a diagnóstico no AWS DMS

Se você encontrar algum problema ao trabalhar com o AWS DMS, seu engenheiro de suporte pode precisar de mais informações sobre o banco de dados de origem ou de destino. Queremos garantir que o AWS Support receba o máximo possível das informações necessárias no menor tempo possível. Portanto, desenvolvemos scripts para consultar essas informações para vários dos principais mecanismos de bancos de dados relacionais.

Se um script de suporte estiver disponível para o banco de dados, baixe-o utilizando o link no tópico do script correspondente descrito a seguir. Depois de verificar e analisar o script (descrito a seguir), é possível executá-lo de acordo com o procedimento descrito no tópico do script. Quando a execução do script estiver concluída, é possível fazer upload da saída para o seu caso do AWS Support (novamente, descrito a seguir).

Antes de executar o script, é possível detectar quaisquer erros que possam ter sido introduzidos ao baixar ou armazenar o script de suporte. Para fazer isso, compare a soma de verificação do arquivo de script com um valor fornecido pelo AWS. A AWS utiliza o algoritmo SHA256 para a soma de verificação.

Como verificar o arquivo de script de suporte utilizando uma soma de verificação

1. Abra o arquivo de soma de verificação mais recente fornecido para verificar esses scripts de suporte em https://d2pwp9zz55emqw.cloudfront.net/sha256Check.txt. Por exemplo, o arquivo pode ter conteúdo como o seguinte.

```
MYSQL  dfafd0d511477c699f96c64693ad0b1547d47e74d5c5f2f2025b790b1422e3c8
ORACLE  6c41ebcfc99518cfa8a10cb2ce8943b153b2cc7049117183d0b5de3d551bc312
POSTGRES  6ccd274863d14f6f3146fbdbbba43f2d8d4c6a4c25380d7b41c71883aa4f9790
SQL_SERVER  971a6f2c46aec8d083d2b3b6549b1e9990af3a15fe4b922e319f4fdd358debe7
```

2. Execute o comando de validação SHA256 para o sistema operacional no diretório que contém o arquivo de suporte. Por exemplo, no sistema operacional macOS, é possível executar o comando a seguir em um script de suporte do Oracle descrito posteriormente neste tópico.

```
shasum -a 256 awsdms_support_collector_oracle.sql
```

3. Compare os resultados do comando com o valor mostrado no arquivo `sha256Check.txt` aberto mais recentemente. Os dois valores devem corresponder. Caso contrário, entre em contato com o seu engenheiro de suporte sobre a incompatibilidade e sobre como é possível obter um arquivo de script de suporte limpo.

Se você tiver um arquivo de script de suporte limpo, antes de executar o script, leia e compreenda o SQL da perspectiva de desempenho e de segurança. Se você não se sentir confortável para executar algum SQL nesse script, poderá comentar ou remover o SQL problemático. Também é possível consultar o seu engenheiro de suporte sobre quaisquer soluções alternativas aceitáveis.

Após a conclusão bem-sucedida e salvo indicação em contrário, o script retorna a saída em um formato HTML legível. O script foi projetado para excluir desse HTML quaisquer dados ou detalhes de segurança que possam comprometer a sua empresa. Ele também não faz modificações no banco de dados ou no ambiente. No entanto, se você encontrar alguma informação no HTML que não se sente à vontade para compartilhar, sinta-se à vontade para remover as informações do problema antes de fazer o upload do HTML. Quando o HTML for aceitável, faça upload dele utilizando os Anexos nos Detalhes do caso do seu caso de suporte.

Cada um dos tópicos a seguir descreve os scripts disponíveis para um banco de dados compatível com o AWS DMS e como executá-los. O seu engenheiro de suporte direcionará você para um script específico documentado a seguir.

Tópicos

Scripts de suporte a diagnóstico do Oracle

A seguir, é possível encontrar os scripts de apoio diagnóstico disponíveis para analisar um banco de dados on-premises ou do Amazon RDS para Oracle na configuração da migração do AWS DMS. Esses scripts funcionam com um endpoint de origem ou de destino. Todos os scripts são escritos para serem executados no utilitário de linha de comando SQL*Plus. Para obter mais informações sobre como utilizar esse utilitário, consulte Utilizar a linha de comandos do SQL na documentação do Oracle.

Antes de executar o script, verifique se a conta de usuário que você utiliza tem as permissões necessárias para acessar o banco de dados Oracle. As configurações de permissões mostradas pressupõem que um usuário tenha sido criado da seguinte forma.

```
CREATE USER script_user IDENTIFIED BY password;
```

Para um banco de dados on-premises, defina as permissões mínimas conforme mostrado a seguir para *script_user*.

```
GRANT CREATE SESSION TO script_user;
GRANT SELECT on V$DATABASE to script_user;
GRANT SELECT on V$VERSION to script_user;
GRANT SELECT on GV$SGA to script_user;
GRANT SELECT on GV$INSTANCE to script_user;
GRANT SELECT on GV$DATAGUARD_CONFIG to script_user;
GRANT SELECT on GV$LOG to script_user;
GRANT SELECT on DBA_TABLESPACES to script_user;
GRANT SELECT on DBA_DATA_FILES to script_user;
GRANT SELECT on DBA_SEGMENTS to script_user;
GRANT SELECT on DBA_LOBS to script_user;
GRANT SELECT on V$ARCHIVED_LOG to script_user;
GRANT SELECT on DBA_TAB_MODIFICATIONS to script_user;
GRANT SELECT on DBA_TABLES to script_user;
```

```
GRANT SELECT on DBA_TAB_PARTITIONS to script_user;
GRANT SELECT on DBA_MVIEWS to script_user;
GRANT SELECT on DBA_OBJECTS to script_user;
GRANT SELECT on DBA_TAB_COLUMNS to script_user;
GRANT SELECT on DBA_LOG_GROUPS to script_user;
GRANT SELECT on DBA_LOG_GROUP_COLUMNS to script_user;
GRANT SELECT on V$ARCHIVE_DEST to script_user;
GRANT SELECT on DBA_SYS_PRIVS to script_user;
GRANT SELECT on DBA_TAB_PRIVS to script_user;
GRANT SELECT on DBA_TYPES to script_user;
GRANT SELECT on DBA_CONSTRAINTS to script_user;
GRANT SELECT on V$TRANSACTION to script_user;
GRANT SELECT on GV$ASM_DISK_STAT to script_user;
GRANT SELECT on GV$SESSION to script_user;
GRANT SELECT on GV$SQL to script_user;
GRANT SELECT on DBA_ENCRYPTED_COLUMNS to script_user;
GRANT SELECT on DBA_PDBS to script_user;

GRANT EXECUTE on dbms_utility to script_user;
```

Para um banco de dados Amazon RDS, defina as permissões mínimas conforme mostrado a seguir.

```
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');
```

A seguir, é possível encontrar descrições de como baixar, revisar e executar cada script de suporte do SQL*Plus disponível para Oracle. Também é possível encontrar como analisar e enviar o resultado para o seu caso de AWS Support.

Tópicos

- awsdms_support_collector_oracle.sql script

awsdms_support_collector_oracle.sql script

Baixe o script awsdms_support_collector_oracle.sql.

Esse script coleta informações sobre a configuração do banco de dados Oracle. Lembre-se de verificar a soma de verificação no script e, se a soma de verificação estiver verificada, revise o código SQL no script para comentar qualquer código que você não se sente à vontade para executar. Quando estiver satisfeito com a integridade e o conteúdo do script, será possível executá-lo.

Como executar o script e fazer upload dos resultados para o caso de suporte

1. Execute o script em seu ambiente de banco de dados utilizando a seguinte linha de comando do SQL*Plus.

```
SQL> @awsdms_support_collector_oracle.sql
```

<result>

O script exibe uma breve descrição e uma solicitação para continuar ou abortar a execução. Pressione [Enter] para continuar.
</result>

2. No prompt a seguir, insira o nome de somente um dos esquemas que deseja migrar.
3. No prompt a seguir, insira o nome do usuário (*`script_user`*) definido para se conectar ao banco de dados.
4. No prompt a seguir, insira o número de dias dos dados que você deseja examinar ou aceite o padrão. O script coleta os dados especificados no banco de dados.
<result>

Depois de concluído, o script exibirá o nome do arquivo HTML de saída, por exemplo `dms_support_oracle-2020-06-22-13-20-39-ORCL.html`. O script salva esse arquivo em seu diretório de trabalho.
</result>
5. Revise esse arquivo HTML e remova todas as informações que você não se sente à vontade para compartilhar. Quando o HTML for aceitável para compartilhar, faça upload do arquivo para o caso do AWS Support. Para obter mais informações sobre como fazer upload desse arquivo, consulte Como trabalhar com scripts de suporte a diagnóstico no AWS DMS.

Scripts de suporte de diagnóstico do SQL Server

A seguir, é possível encontrar os scripts de apoio diagnóstico disponíveis para analisar um banco de dados on-premises ou do Amazon RDS para SQL Server na configuração da migração do AWS DMS. Esses scripts funcionam com um endpoint de origem ou de destino. Para um banco de dados on-premises, execute esses scripts no utilitário de linha de comando sqlcmd. Para obter mais informações sobre como usar esse utilitário, consulte sqlcmd: utilizar o utilitário na documentação da Microsoft.

Para um banco de dados Amazon RDS, não é possível se conectar utilizando o utilitário de linha de comando sqlcmd. Em vez disso, execute esses scripts utilizando qualquer ferramenta de cliente que se conecte ao SQL Server do Amazon RDS.

Antes de executar o script, verifique se a conta de usuário utilizada tem as permissões necessárias para acessar o banco de dados SQL Server. Para um banco de dados on-premises e para um banco de dados Amazon RDS, é possível utilizar as mesmas permissões utilizadas para acessar o banco de dados SQL Server sem o perfil `SysAdmin`.

Tópicos

Configurar permissões mínimas para um banco de dados SQL Server on-premises

Configurar permissões mínimas para executar um banco de dados SQL Server on-premises

1. Crie uma conta do SQL Server com autenticação por senha utilizando o SQL Server Management Studio (SSMS), por exemplo *`on-prem-user`*.
2. Na seção Mapeamentos de usuário do SSMS, escolha os bancos de dados MSDB e MASTER (que fornecem permissão pública) e atribua o perfil `DB_OWNER` ao banco de dados no qual você deseja executar a replicação contínua.
3. Abra o menu de contexto (clique com o botão direito do mouse) para a nova conta e escolha Segurança para conceder explicitamente o privilégio `Connect SQL`.
4. Execute os seguintes comandos de concessão.

```
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;
```

Configurar permissões mínimas para um banco de dados SQL Server do Amazon RDS

Como executar com as permissões mínimas para um banco de dados SQL Server do Amazon RDS

1. Crie uma conta do SQL Server com autenticação por senha utilizando o SQL Server Management Studio (SSMS), por exemplo *rds-user*.
2. Na seção Mapeamentos de usuário do SSMS, escolha os bancos de dados do MSDB (que fornece permissão pública) e atribua o perfil DB_OWNER ao banco de dados em que você deseja executar o script.
3. Abra o menu de contexto (clique com o botão direito do mouse) para a nova conta e escolha Segurança para conceder explicitamente o privilégio Connect SQL.
4. Execute os seguintes comandos de concessão.

```
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;
```

Configurar a replicação contínua em um SQL Server autônomo: sem o perfil sysadmin

Esta seção descreve como configurar a replicação contínua para uma origem de banco de dados SQL Server autônomo que não exige que a conta do usuário tenha privilégios de sysadmin.

> **Note**
>
> Depois de executar as etapas desta seção, o usuário do DMS que não for administrador de sistema terá permissões para fazer o seguinte:
>
> - Ler as alterações do arquivo de log de transações on-line.
> - Acessar o disco para ler as alterações dos arquivos de backup do log de transações.
> - Adicionar ou alterar a publicação que o DMS usa.
> - Adicionar artigos à publicação.

1. Configure o Microsoft SQL Server para replicação conforme descrito em Capturar dados alterados no SQL Server autogerenciado on-premises ou no Amazon EC2.
2. Ative MS-REPLICATION no banco de dados de origem. Isso pode ser feito manualmente ou executando a tarefa uma vez como usuário sysadmin.
3. Crie o esquema `awsdms` no banco de dados de origem utilizando o seguinte script:

```
use master
go
create schema awsdms
go


-- Create the table valued function [awsdms].[split_partition_list] on the Master
 database, as follows:
USE [master]
GO

set ansi_nulls on
go

set quoted_identifier on
go

if (object_id('[awsdms].[split_partition_list]','TF')) is not null

drop function [awsdms].[split_partition_list];

go

create function [awsdms].[split_partition_list]

(

@plist varchar(8000), –A delimited list of partitions

@dlm nvarchar(1) –Delimiting character

)

returns @partitionsTable table –Table holding the BIGINT values of the string
 fragments
```

```
(

pid bigint primary key

)

as

begin

declare @partition_id bigint;

declare @dlm_pos integer;

declare @dlm_len integer;

set @dlm_len = len(@dlm);

while (charindex(@dlm,@plist)>0)

begin

set @dlm_pos = charindex(@dlm,@plist);

set @partition_id = cast( ltrim(rtrim(substring(@plist,1,@dlm_pos-1))) as bigint);

insert into @partitionsTable (pid) values (@partition_id)

set @plist = substring(@plist,@dlm_pos+@dlm_len,len(@plist));

end

set @partition_id = cast (ltrim(rtrim(@plist)) as bigint);

insert into @partitionsTable (pid) values ( @partition_id );

return

end

GO
```

4. Crie o procedimento `[awsdms].[rtm_dump_dblog]` no banco de dados mestre utilizando o seguinte script:

```
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)

where [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp collate
 SQL_Latin1_General_CP1_CI_AS

and

(

( [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )

or
```

```
( [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')

and

( ( [context] in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') ) or ([context] =
 'LCX_TEXT_MIX' and (datalength([RowLog Contents 0]) in (0,1))))

and [PartitionID] in ( select * from master.awsdms.split_partition_list
 (@partition_list,','))

)

or

([operation] = 'LOP_HOBT_DDL')

)


else


SELECT

[Current LSN],

[operation],

[Context],

[Transaction ID],

[Transaction Name],

[Begin Time],

[End Time],

[Flag Bits],

[PartitionID],

[Page ID],
```

```
[Slot ID],

[RowLog Contents 0],

[Log Record],

[RowLog Contents 1] – After Image

FROM

fn_dump_dblog (

@start_lsn, NULL, N'DISK', @seqno, @filename,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default,

default, default, default, default, default, default, default)

where [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp collate
 SQL_Latin1_General_CP1_CI_AS

and

(

( [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )

or
```

```
( [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')

and

( ( [context] in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') ) or ([context] =
 'LCX_TEXT_MIX' and (datalength([RowLog Contents 0]) in (0,1))))

and ([PartitionID] is not null) and ([PartitionID] >= @minPartition and
 [PartitionID]<=@maxPartition)

)

or

([operation] = 'LOP_HOBT_DDL')

)



SET NOCOUNT OFF – Re-enable "rows affected display"

end

GO
```

5. Crie o certificado no banco de dados mestre utilizando o seguinte script:

```
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. Crie o login no certificado utilizando o seguinte script:

```
Use [master]
Go

CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
 [awsdms_rtm_dump_dblog_cert];
```

7. Adicione o login ao perfil do servidor sysadmin utilizando o seguinte script:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

8. Adicione a assinatura ao [master].[awdms].[rtm_dump_dblog] utilizando o certificado e o seguinte script:

```
Use [master]
GO
ADD SIGNATURE
TO [master].[awsdms].[rtm_dump_dblog] BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
 WITH PASSWORD = '@5trongpassword';
```

> **Note**
>
> Se você recriar o procedimento armazenado, será necessário adicionar a assinatura novamente.

9. Crie o [awsdms].[rtm_position_1st_timestamp] no banco de dados principal usando o seguinte script:

```
use [master]
    if object_id('[awsdms].[rtm_position_1st_timestamp]','P') is not null
    DROP PROCEDURE [awsdms].[rtm_position_1st_timestamp];
    go
    create procedure [awsdms].[rtm_position_1st_timestamp]
    (
    @dbname                sysname,      -- Database name
    @seqno                 integer,      -- Backup set sequence/position number
 within file
    @filename              varchar(260), -- The backup filename
    @1stTimeStamp          varchar(40)   -- The timestamp to position by
    )
    as begin

    SET NOCOUNT ON       -- Disable "rows affected display"

    declare @firstMatching table
    (
    cLsn varchar(32),
    bTim datetime
```

```
    )

    declare @sql nvarchar(4000)
    declare @nl                         char(2)
    declare @tb                         char(2)
    declare @fnameVar                   nvarchar(254) = 'NULL'

    set @nl  = char(10); -- New line
    set @tb  = char(9)   -- Tab separator

    if (@filename is not null)
    set @fnameVar = ''''+@filename +''''

    set @sql='use ['+@dbname+'];'+@nl+
    'select top 1 [Current LSN],[Begin Time]'+@nl+
    'FROM fn_dump_dblog (NULL, NULL, NULL, '+ cast(@seqno as varchar(10))+','+
@fnameVar+','+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default)'+@nl+
    'where operation=''LOP_BEGIN_XACT''' +@nl+
    'and [Begin Time]>= cast('+''''+@1stTimeStamp+''''+' as datetime)'+@nl

    --print @sql
    delete from  @firstMatching
    insert into @firstMatching  exec sp_executesql @sql    -- Get them all

    select top 1 cLsn as [matching LSN],convert(varchar,bTim,121) as [matching
Timestamp] from @firstMatching;

    SET NOCOUNT OFF      -- Re-enable "rows affected display"

    end
    GO
```

10. Crie o certificado no banco de dados mestre utilizando o seguinte script:

```
Use [master]
```

```
Go
CREATE CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
ENCRYPTION BY PASSWORD = '@5trongpassword'
WITH SUBJECT = N'Certificate for FN_POSITION_1st_TIMESTAMP Permissions';
```

11. Crie o login no certificado utilizando o seguinte script:

```
Use [master]
Go
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
 [awsdms_rtm_position_1st_timestamp_cert];
```

12. Adicione o login ao perfil sysadmin utilizando o seguinte script:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_position_1st_timestamp_login];
```

13. Adicione a assinatura ao [master].[awsdms].[rtm_position_1st_timestamp] utilizando o certificado e o seguinte script:

```
Use [master]
    GO
    ADD SIGNATURE
    TO [master].[awsdms].[rtm_position_1st_timestamp]
    BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
    WITH PASSWORD = '@5trongpassword';
```

14. Conceda ao usuário do DMS acesso de execução ao novo procedimento armazenado usando o seguinte script:

```
use master
go
GRANT execute on [awsdms].[rtm_position_1st_timestamp] to dms_user;
```

15. Crie um usuário com as seguintes permissões e perfis em cada um dos seguintes bancos de dados:

> **Note**
>
> Crie a conta de usuário dmsnosysadmin com o mesmo SID em cada réplica. A consulta SQL a seguir pode ajudar a verificar o valor do SID da conta dmsnosysadmin em cada réplica. Para obter mais informações sobre como criar um usuário, consulte CREATE USER (Transact-SQL) na Documentação do Microsoft SQL Server. Para obter mais

informações sobre a criação de contas de usuário do SQL para o banco de dados SQL do Azure, consulte Replicação geográfica ativa.

```
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]
```

Execute o seguinte script no banco de dados de origem:

```
EXEC sp_addrolemember N'db_owner', N'DMS_user'
use Source_DB
go
```

16. Por fim, adicione um atributo de conexão extra (ECA) ao endpoint do SQL Server de origem:

```
enableNonSysadminWrapper=true;
```

Configurar a replicação contínua em um SQL Server em um ambiente de grupo de disponibilidade: sem o perfil sysadmin

Esta seção descreve como configurar a replicação contínua para uma origem de banco de dados SQL Server em um ambiente de grupo de disponibilidade que não exige que a conta do usuário tenha privilégios de sysadmin.

> **Note**
>
> Depois de executar as etapas desta seção, o usuário do DMS que não for administrador de sistema terá permissões para fazer o seguinte:
>
> - Ler as alterações do arquivo de log de transações on-line.
> - Acessar o disco para ler as alterações dos arquivos de backup do log de transações.
> - Adicionar ou alterar a publicação que o DMS usa.
> - Adicionar artigos à publicação.

Como configurar a replicação contínua sem utilizar o usuário sysadmin em um ambiente de grupo de disponibilidade

1. Configure o Microsoft SQL Server para replicação conforme descrito em Capturar dados alterados no SQL Server autogerenciado on-premises ou no Amazon EC2.
2. Ative MS-REPLICATION no banco de dados de origem. Isso pode ser feito manualmente ou executando a tarefa uma vez utilizando um usuário sysadmin.

   > **Note**
   >
   > Configure o distribuidor MS-REPLICATION como local ou de uma forma que permita acesso a usuários que não sejam administradores de sistema por meio do servidor vinculado associado.

3. Se a opção do endpoint Usar exclusivamente sp_repldone em uma única tarefa estiver ativada, interrompa o trabalho do MS-REPLICATION Log Reader.
4. Execute as seguintes etapas em cada réplica:

   1. Crie o esquema `[awsdms]`[awsdms] no banco de dados mestre:

      ```
      CREATE SCHEMA [awsdms]
      ```

   2. Crie o perfil com valor de tabela `[awsdms].[split_partition_list]` no banco de dados mestre:

      ```
      USE [master]
      GO
      ```

```
SET ansi_nulls on
GO

SET quoted_identifier on
GO

IF (object_id('[awsdms].[split_partition_list]','TF')) is not null
  DROP FUNCTION [awsdms].[split_partition_list];
GO

CREATE FUNCTION [awsdms].[split_partition_list]
(
  @plist varchar(8000),    --A delimited list of partitions
  @dlm nvarchar(1)    --Delimiting character
)
RETURNS @partitionsTable table --Table holding the BIGINT values of the string
 fragments
(
  pid bigint primary key
)
AS
BEGIN
  DECLARE @partition_id bigint;
  DECLARE @dlm_pos integer;
  DECLARE @dlm_len integer;
  SET @dlm_len = len(@dlm);
  WHILE (charindex(@dlm,@plist)>0)
  BEGIN
    SET @dlm_pos = charindex(@dlm,@plist);
    SET @partition_id = cast( ltrim(rtrim(substring(@plist,1,@dlm_pos-1))) as
 bigint);
    INSERT into @partitionsTable (pid) values (@partition_id)
    SET @plist = substring(@plist,@dlm_pos+@dlm_len,len(@plist));
  END
  SET @partition_id = cast (ltrim(rtrim(@plist)) as bigint);
  INSERT into @partitionsTable (pid) values (  @partition_id  );
  RETURN
END
GO
```

3. Crie o procedimento `[awsdms].[rtm_dump_dblog]` no banco de dados mestre:

```
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)
   WHERE
     [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp collate
SQL_Latin1_General_CP1_CI_AS -- This aims for implementing FN_DBLOG based on GT
comparator.
     AND
     (
       (  [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )
       OR
       (  [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')
         AND
         ( ( [context]   in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') )
or ([context] = 'LCX_TEXT_MIX') )
         AND
         [PartitionID] in ( select * from master.awsdms.split_partition_list
(@partition_list,','))
       )
     OR
```

```
      ([operation] = 'LOP_HOBT_DDL')
    )
    ELSE
      SELECT
        [Current LSN],
        [operation],
        [Context],
        [Transaction ID],
        [Transaction Name],
        [Begin Time],
        [End Time],
        [Flag Bits],
        [PartitionID],
        [Page ID],
        [Slot ID],
        [RowLog Contents 0],
        [Log Record],
        [RowLog Contents 1] -- After Image
      FROM
        fn_dump_dblog (
          @start_lsn, NULL, N'DISK', @seqno, @filename,
          default, default, default, default, default, default, default,
          default, default, default, default, default, default, default,
          default, default, default, default, default, default, default,
          default, default, default, default, default, default, default,
          default, default, default, default, default, default, default,
          default, default, default, default, default, default, default,
          default, default, default, default, default, default, default,
          default, default, default, default, default, default, default)
      WHERE [Current LSN] collate SQL_Latin1_General_CP1_CI_AS > @start_lsn_cmp
collate SQL_Latin1_General_CP1_CI_AS -- This aims for implementing FN_DBLOG
based on GT comparator.
      AND
      (
        (  [operation] in ('LOP_BEGIN_XACT','LOP_COMMIT_XACT','LOP_ABORT_XACT') )
        OR
        (  [operation] in ('LOP_INSERT_ROWS','LOP_DELETE_ROWS','LOP_MODIFY_ROW')
          AND
          ( ( [context]   in ('LCX_HEAP','LCX_CLUSTERED','LCX_MARK_AS_GHOST') )
or ([context] = 'LCX_TEXT_MIX') )
          AND ([PartitionID] is not null) and ([PartitionID] >= @minPartition and
[PartitionID]<=@maxPartition)
        )
```

```
        OR
        ([operation] = 'LOP_HOBT_DDL')
      )
      SET NOCOUNT OFF -- Re-enable "rows affected display"
END
GO
```

4. Crie um certificado no banco de dados mestre:

```
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. Crie um login no certificado:

```
USE [master]
GO
CREATE LOGIN awsdms_rtm_dump_dblog_login FROM CERTIFICATE
  [awsdms_rtm_dump_dblog_cert];
```

6. Adicione o login ao perfil do servidor sysadmin:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [awsdms_rtm_dump_dblog_login];
```

7. Adicione a assinatura ao procedimento [master].[awdms].[rtm_dump_dblog] utilizando o certificado:

```
USE [master]
GO

ADD SIGNATURE
  TO [master].[awsdms].[rtm_dump_dblog]
  BY CERTIFICATE [awsdms_rtm_dump_dblog_cert]
  WITH PASSWORD = '@hardpassword1';
```

> **Note**
>
> Se você recriar o procedimento armazenado, será necessário adicionar a assinatura novamente.

8. Crie o procedimento `[awsdms].[rtm_position_1st_timestamp]` no banco de dados mestre:

```
USE [master]
IF object_id('[awsdms].[rtm_position_1st_timestamp]','P') is not null
  DROP PROCEDURE [awsdms].[rtm_position_1st_timestamp];
GO
CREATE PROCEDURE [awsdms].[rtm_position_1st_timestamp]
(
  @dbname                sysname,      -- Database name
  @seqno                 integer,      -- Backup set sequence/position number
 within file
  @filename              varchar(260), -- The backup filename
  @1stTimeStamp          varchar(40)   -- The timestamp to position by
)
AS
BEGIN
  SET NOCOUNT ON       -- Disable "rows affected display"

  DECLARE @firstMatching table
  (
    cLsn varchar(32),
    bTim datetime
  )
  DECLARE @sql nvarchar(4000)
  DECLARE @nl                       char(2)
  DECLARE @tb                       char(2)
  DECLARE @fnameVar                 sysname = 'NULL'

  SET @nl  = char(10); -- New line
  SET @tb  = char(9)   -- Tab separator

  IF (@filename is not null)
    SET @fnameVar = ''''+@filename +''''
  SET @filename = ''''+@filename +''''
  SET @sql='use ['+@dbname+'];'+@nl+
    'SELECT TOP 1 [Current LSN],[Begin Time]'+@nl+
    'FROM fn_dump_dblog (NULL, NULL, NULL, '+ cast(@seqno as varchar(10))+','+
 @filename +','+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
```

```
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default,'+@nl+
    @tb+'default, default, default, default, default, default, default)'+@nl+
    'WHERE operation=''LOP_BEGIN_XACT''' +@nl+
    'AND [Begin Time]>= cast('+''''+@1stTimeStamp+''''+' as datetime)'+@nl

    --print @sql
    DELETE FROM @firstMatching
    INSERT INTO @firstMatching  exec sp_executesql @sql    -- Get them all
    SELECT TOP 1 cLsn as [matching LSN],convert(varchar,bTim,121) AS[matching
 Timestamp] FROM @firstMatching;

    SET NOCOUNT OFF      -- Re-enable "rows affected display"

END
GO
```

9. Crie um certificado no banco de dados mestre:

```
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. Crie um login no certificado:

```
USE [master]
GO
CREATE LOGIN awsdms_rtm_position_1st_timestamp_login FROM CERTIFICATE
  [awsdms_rtm_position_1st_timestamp_cert];
```

11. Adicione o login ao perfil do servidor sysadmin:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER
 [awsdms_rtm_position_1st_timestamp_login];
```

12. Adicione a assinatura ao procedimento `[master].[awsdms].[rtm_position_1st_timestamp]` utilizando o certificado:

```
USE [master]
```

```
GO
ADD SIGNATURE
  TO [master].[awsdms].[rtm_position_1st_timestamp]
  BY CERTIFICATE [awsdms_rtm_position_1st_timestamp_cert]
  WITH PASSWORD = '@hardpassword1';
```

> **Note**
>
> Se você recriar o procedimento armazenado, será necessário adicionar a assinatura novamente.

13. Crie um usuário com as seguintes permissões/perfis em cada um dos seguintes bancos de dados:

> **Note**
>
> Crie a conta de usuário dmsnosysadmin com o mesmo SID em cada réplica. A consulta SQL a seguir pode ajudar a verificar o valor do SID da conta dmsnosysadmin em cada réplica. Para obter mais informações sobre como criar um usuário, consulte CREATE USER (Transact-SQL) na Documentação do Microsoft SQL Server. Para obter mais informações sobre a criação de contas de usuário do SQL para o banco de dados SQL do Azure, consulte Replicação geográfica ativa.

```
SELECT @@servername servername, name, sid, create_date, modify_date
  FROM sys.server_principals
  WHERE name = 'dmsnosysadmin';
```

14. Conceda permissões no banco de dados mestre em cada réplica:

```
USE master
GO

GRANT select on sys.fn_dblog to dmsnosysadmin;
GRANT view any definition to dmsnosysadmin;
GRANT view server state to dmsnosysadmin -- (should be granted to the login).
GRANT execute on sp_repldone to dmsnosysadmin;
GRANT execute on sp_replincrementlsn to dmsnosysadmin;
GRANT execute on sp_addpublication to dmsnosysadmin;
GRANT execute on sp_addarticle to dmsnosysadmin;
```

```
GRANT execute on sp_articlefilter to dmsnosysadmin;
GRANT select on [awsdms].[split_partition_list] to dmsnosysadmin;
GRANT execute on [awsdms].[rtm_dump_dblog] to dmsnosysadmin;
GRANT execute on [awsdms].[rtm_position_1st_timestamp] to dmsnosysadmin;
```

15. Conceda permissões no banco de dados msdb em cada réplica:

```
USE msdb
GO
GRANT select on msdb.dbo.backupset to dmsnosysadmin
GRANT select on msdb.dbo.backupmediafamily to dmsnosysadmin
GRANT select on msdb.dbo.backupfile to dmsnosysadmin
```

16. Adicione o perfil `db_owner` ao `dmsnosysadmin` no banco de dados de origem. Como o banco de dados é sincronizado, é possível adicionar o perfil somente na réplica primária.

```
use <source DB>
GO
EXEC sp_addrolemember N'db_owner', N'dmsnosysadmin'
```

Scripts de suporte do SQL Server

Os tópicos a seguir descrevem como baixar, revisar e executar cada script de suporte disponível para o SQL Server. Eles também descrevem como analisar e fazer upload do resultado do script para o seu caso do AWS Support.

Tópicos

- awsdms_support_collector_sql_server.sql script

awsdms_support_collector_sql_server.sql script

Baixe o script `awsdms_support_collector_sql_server.sql`.

> **Note**
>
> Execute esse script de apoio diagnóstico somente no SQL Server 2014 e em versões superiores.

Esse script coleta informações sobre a configuração do banco de dados SQL Server. Lembre-se de verificar a soma de verificação no script e, se a soma de verificação estiver verificada, revise o código SQL no script para comentar qualquer código que você não se sente à vontade para executar. Quando estiver satisfeito com a integridade e o conteúdo do script, será possível executá-lo.

Como executar o script para um banco de dados SQL Server on-premises

1. Execute o script utilizando a linha de comando sqlcmd a seguir.

```
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
```

   Os parâmetros do comando sqlcmd especificados incluem o seguinte:

   - `-U`: nome do usuário do banco de dados.
   - `-P`: senha do usuário do banco de dados.
   - `-S`: nome do servidor do banco de dados SQL Server.
   - `-y`: largura máxima das colunas de saída do utilitário sqlcmd. Um valor de 0 especifica colunas de largura ilimitada.
   - `-i`: caminho do script de suporte a ser executado, neste caso `awsdms_support_collector_sql_server.sql`.
   - `-o`: caminho do arquivo HTML de saída, com um nome de arquivo que você especifica, contendo as informações de configuração do banco de dados coletado.
   - `-d`: nome do banco de dados SQL Server.

2. Depois que o script for concluído, revise o arquivo HTML de saída e remova todas as informações que você não se sente à vontade para compartilhar. Quando o HTML for aceitável para compartilhar, faça upload do arquivo para o caso do AWS Support. Para obter mais informações sobre como fazer upload desse arquivo, consulte Como trabalhar com scripts de suporte a diagnóstico no AWS DMS.

Com o Amazon RDS para SQL Server, não é possível se conectar utilizando o utilitário de linha de comando sqlcmd, portanto, utilize o procedimento a seguir.

Como executar o script para um banco de dados RDS SQL Server

1. Execute o script utilizando qualquer ferramenta de cliente que permita conectar-se ao RDS SQL Server como o usuário `Master` e salve a saída como um arquivo HTML.
2. Revise o arquivo HTML de saída e remova todas as informações que você não se sente à vontade para compartilhar. Quando o HTML for aceitável para compartilhar, faça upload do arquivo para o caso do AWS Support. Para obter mais informações sobre como fazer upload desse arquivo, consulte Como trabalhar com scripts de suporte a diagnóstico no AWS DMS.

Scripts de suporte de diagnóstico para bancos de dados compatíveis com o MySQL

A seguir, é possível encontrar os scripts de apoio diagnóstico disponíveis para analisar um banco de dados on-premises ou compatível com o Amazon RDS para MySQL na configuração de migração do AWS DMS. Esses scripts funcionam com um endpoint de origem ou de destino. Todos os scripts são escritos para serem executados na linha de comando do MySQL.

Para obter informações sobre como instalar o cliente MySQL, consulte Instalar o shell do MySQL na documentação do MySQL. Para obter informações sobre como utilizar o cliente MySQL, consulte Utilizar comandos do shell do MySQL na documentação do MySQL.

Antes de executar um script, verifique se a conta de usuário utilizada possui as permissões necessárias para acessar o banco de dados compatível com MySQL. Utilize o procedimento a seguir para criar uma conta de usuário e fornecer as permissões mínimas necessárias para executar esse script.

Como configurar uma conta de usuário com as permissões mínimas para executar esses scripts

1. Crie o usuário para executar os scripts.

```
create user 'username'@'hostname' identified by password;
```

2. Conceda o comando `select` nos bancos de dados para analisá-los.

```
grant select on database-name.* to username;
grant replication client on *.* to username;
```

3.

```
grant execute on procedure mysql.rds_show_configuration to username;
```

Os tópicos a seguir descrevem como baixar, revisar e executar cada script de suporte disponível para um banco de dados compatível com o MySQL. Eles também descrevem como analisar e fazer upload do resultado do script para o seu caso do AWS Support.

Tópicos

awsdms_support_collector_MySQL.sql script

Baixe o script `awsdms_support_collector_MySQL.sql`.

Esse script coleta informações sobre a configuração do banco de dados compatível com o MySQL. Lembre-se de verificar a soma de verificação no script e, se a soma de verificação estiver verificada, revise o código SQL no script para comentar qualquer código que você não se sente à vontade para executar. Quando estiver satisfeito com a integridade e o conteúdo do script, será possível executá-lo.

Execute o script depois de se conectar ao ambiente de banco de dados utilizando a linha de comando.

Como executar esse script e fazer upload dos resultados para o caso de suporte

1. Conecte-se ao banco de dados utilizando o seguinte comando `mysql`.

   ```
   mysql -h hostname -P port -u username database-name
   ```

2. Execute o script utilizando o seguinte comando `source`.

   ```
   mysql> source awsdms_support_collector_MySQL_compatible_DB.sql
   ```

   Analise o relatório gerado e remova todas as informações que você não se sente à vontade para compartilhar. Quando o conteúdo for aceitável para compartilhar, faça upload do arquivo para o seu caso do AWS Support. Para obter mais informações sobre como fazer upload desse arquivo, consulte Como trabalhar com scripts de suporte a diagnóstico no AWS DMS.

> **Note**
>
> - Se você já tiver uma conta de usuário com os privilégios necessários descritos em Scripts de suporte de diagnóstico para bancos de dados compatíveis com o MySQL , poderá utilizar a conta de usuário existente para executar o script.
> - Lembre-se de se conectar ao banco de dados antes de executar o script.
> - O script gera sua saída no formato de texto.
> - Com as práticas recomendadas de segurança em mente, se você criar uma nova conta de usuário somente para executar esse script de apoio diagnóstico do MySQL. É recomendável excluir essa conta de usuário após a execução bem-sucedida do script.

Scripts de apoio diagnóstico do PostgreSQL

A seguir, é possível encontrar os scripts de apoio diagnóstico disponíveis para analisar qualquer PostgreSQL RDBMS (on-premises, Amazon RDS ou Aurora PostgreSQL) na configuração da migração do AWS DMS. Esses scripts funcionam com um endpoint de origem ou de destino. Os scripts são todos escritos para serem executados no utilitário de linha de comando psql.

Antes de executar esses scripts, verifique se a conta de usuário utilizada possui as seguintes permissões necessárias para acessar qualquer PostgreSQL RDBMS:

- PostgreSQL 10.x ou superior: uma conta de usuário com permissão de execução no perfil `pg_catalog.pg_ls_waldir`.
- PostgreSQL 9.x ou anterior: uma conta de usuário com permissões padrão.

É recomendável utilizar uma conta existente com as permissões apropriadas para executar esses scripts.

Se for necessário criar uma conta de usuário ou conceder permissões a uma conta existente para executar esses scripts, é possível executar os seguintes comandos SQL para qualquer PostgreSQL RDBMS com base na versão do PostgreSQL.

Como conceder permissões à conta para executar esses scripts em bancos de dados PostgreSQL versão 10.x ou superior

- Execute um dos seguintes procedimentos:

- Para uma nova conta de usuário, execute o seguinte.

```
CREATE USER script_user WITH PASSWORD 'password';
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

- Para uma conta de usuário existente, execute o seguinte.

```
GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_waldir TO script_user;
```

Como conceder permissões à conta para executar esses scripts para um banco de dados PostgreSQL 9.x ou superior

- Execute um destes procedimentos:
  - Para uma nova conta de usuário, execute o seguinte com as permissões padrão.

```
CREATE USER script_user WITH PASSWORD password;
```

  - Para uma conta de usuário existente, utilize as permissões existentes.

> **Note**
>
> Esses scripts não são compatíveis com determinadas funcionalidades relacionadas à localização do tamanho do WAL para bancos de dados PostgreSQL 9.x e anteriores. Para obter mais informações, consulte o AWS Support.

Os tópicos a seguir descrevem como baixar, revisar e executar cada script de apoio disponível para o PostgreSQL. Eles também descrevem como revisar e fazer upload da saída do script para o caso do AWS Support.

Tópicos

awsdms_support_collector_postgres.sql script

Baixe o script awsdms_support_collector_postgres.sql.

Esse script coleta informações sobre a configuração do banco de dados PostgreSQL. Lembre-se de verificar a soma de verificação no script. Se a soma de verificação for verificada, revise o código SQL no script para comentar qualquer código que você não se sente à vontade para executar. Quando estiver satisfeito com a integridade e o conteúdo do script, será possível executá-lo.

> **Note**
>
> É possível executar esse script com a versão 10 ou superior do cliente psql.

É possível usar os procedimentos a seguir para executar esse script no ambiente de banco de dados ou na linha de comando. Em ambos os casos, é possível fazer upload do arquivo para o AWS Support mais tarde.

Como executar esse script e fazer upload dos resultados para o caso de suporte

1. Execute um destes procedimentos:

   - Execute o script no ambiente de banco de dados utilizando a seguinte linha de comando psql.

     ```
     dbname=# \i awsdms_support_collector_postgres.sql
     ```

     No prompt a seguir, insira o nome de somente um dos esquemas que deseja migrar.

     No prompt a seguir, insira o nome do usuário (*script_user*) definido para se conectar ao banco de dados.

   - Execute o script a seguir diretamente na linha de comando. Essa opção evita qualquer solicitação anterior à execução do script.

     ```
     psql -h database-hostname -p port -U script_user -d database-name -f
      awsdms_support_collector_postgres.sql
     ```

2. Revise o arquivo HTML de saída e remova todas as informações que você não se sente à vontade para compartilhar. Quando o HTML for aceitável para compartilhar, faça upload do arquivo para o caso do AWS Support. Para obter mais informações sobre como fazer upload desse arquivo, consulte Como trabalhar com scripts de suporte a diagnóstico no AWS DMS.

Trabalhando com o suporte AWS DMS de diagnóstico AMI

Se você encontrar um problema relacionado à rede ao trabalhar com AWS DMS, seu engenheiro de suporte pode precisar de mais informações sobre sua configuração de rede. Queremos garantir que o AWS Support receba o máximo possível das informações necessárias no menor tempo possível. Portanto, desenvolvemos uma AMI pré-construída do Amazon EC2 com ferramentas de diagnóstico para AWS DMS testar seu ambiente de rede.

Os testes de diagnóstico instalados na imagem de máquina da Amazon (AMI) incluem o seguinte:

- Nuvem privada virtual (VPC)
- Perda de pacotes de rede
- Latência de rede
- Tamanho da unidade de transmissão máxima (MTU)

Tópicos

> **Note**
>
> Se você tiver problemas de desempenho com a origem do Oracle, poderá avaliar o desempenho de leitura dos redo logs ou do arquivamento do Oracle para encontrar maneiras de melhorar o desempenho. Para obter mais informações, consulte Avaliação do desempenho de leitura de logs de redo ou de arquivamento do Oracle.

Inicie uma nova instância AWS DMS de diagnóstico do Amazon EC2

Nesta seção, você inicia uma nova instância do Amazon EC2. Para obter informações sobre como iniciar uma instância do Amazon EC2, consulte o tutorial Começar a usar instâncias do Linux do Amazon EC2 no Guia do usuário do Amazon EC2.

Inicie uma instância do Amazon EC2 com as seguintes configurações:

- Em Aplicação e imagens do sistema operacional (imagem de máquina da Amazon), pesquise a AMI do DMS-DIAG-AMI. Se você estiver conectado ao console, poderá pesquisar a AMI com esta consulta. Para obter o ID da AMI de AWS diagnóstico em sua região, consulte a IDs de AMIs por região seguir.
- Em Tipo de instância, é recomendável escolher t2.micro.
- Em Configurações de rede, escolha a mesma VPC utilizada pela instância de replicação.

Depois que a instância estiver ativa, conecte-se à instância. Para obter informações sobre como se conectar a uma instância do Linux do Amazon EC2, consulte Conectar-se à sua instância do Linux.

Criar um perfil do IAM

Se quiser executar os testes de diagnóstico na instância de replicação utilizando as permissões mínimas necessárias, crie um perfil do IAM que utilize a seguinte política de permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "dms:DescribeEndpoints",
                "dms:DescribeTableStatistics",
                "dms:DescribeReplicationInstances",
                "dms:DescribeReplicationTasks",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "*"
        }
    ]
}
```

Anexe o perfil a um novo usuário do IAM. Para obter informações sobre como criar perfis, políticas e usuários do IAM, consulte as seguintes seções no Guia do usuário do IAM:

- Conceitos básicos do IAM

- Criar perfis do IAM
- Criar políticas do IAM

Executar os testes de diagnóstico

Depois de criar uma instância do Amazon EC2 e de se conectar a ela, faça o seguinte para executar testes de diagnóstico na instância de replicação.

1. Configure a AWS CLI:

   ```
   $ aws configure
   ```

   Forneça as credenciais de acesso para a conta de AWS usuário que você deseja usar para executar os testes de diagnóstico. Forneça a região da VPC e a instância de replicação.

2. Exiba as AWS DMS tarefas disponíveis na sua região. Substitua a região de exemplo pela sua região.

   ```
   $ dms-report -r us-east-1 -l
   ```

   Esse comando exibe o status das suas tarefas.

3. Exiba os endpoints e as configurações da tarefa. Substitua *<DMS-Task-ARN>* pelo nome do recurso da Amazon (ARN) da tarefa.

```
$ dms-report -t <DMS-Task-ARN>
```

Esse comando exibe os endpoints e as configurações da tarefa.

4. Execute os testes de diagnóstico. Substitua *<DMS-Task-ARN>* pelo ARN da tarefa.

```
$ dms-report -t <DMS-Task-ARN> -n y
```

Esse comando exibe os dados de diagnóstico sobre a VPC da instância de replicação, a transmissão de pacotes de rede, a latência da rede e o tamanho da unidade máxima de transmissão (MTU) da rede.


```
########
#
#
#       AWS DMS Diagnostic
#       Date: 07-13-2022
#
#
#       aws region: us-east-2
#
#
########
==== DMS DIAG Info ====
Public IP: 3.
Private IP: 172.30.0.240
Instance ID: i-04829b2beb8214602
Instance MAC: 02:58:04:b5:52:28
Instane Type: t2.micro
Instance Sec Group: D
Instance AWS Region: us-east-2
Instance VPC Id: vpc-08ba020355d8a952e


==== Network Packet Check ====
1.) Check DMS EC2 MetaData service
>>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
     Looks good with no issue. <<<<<

2.) Check Source endpoint (d
>>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
     Looks good with no issue. <<<<<

3.) Check Target endpoint (r
>>>>>Result: 10 packets transmitted, 10 packets received, 0% packet loss
     Looks good with no issue. <<<<<

==== End network packet check ====


==== Network Latency Check ====
1.) Check DMS MetaData Service
>>>>>Result: round-trip min/avg/max = 0.4/0.4/0.5 ms
     Looks good with no issue. <<<<<

2.) Check Source endpoint (d
>>>>>Result: round-trip min/avg/max = 1.0/1.1/1.2 ms
     Looks good with no issue. <<<<<

3.) Check Target endpoint (r
>>>>>Result: round-trip min/avg/max = 1.4/1.4/1.5 ms
     Looks good with no issue. <<<<<

==== End network latency check ====


==== Network MTU Check ====
1.) Check DMS MetaData Service
>>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<


2.) Check Source endpoint (d
>>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<


3.) Check Target endpoint (r
>>>>>Result: MTU setting looks good. Local MTU (9001) matches remote MTU (9001) <<<<<


==== End network MTU check ====
```

Próximos Passos

As seções a seguir descrevem as informações de solução de problemas com base nos resultados dos testes de diagnóstico da rede:

Testes da VPC

Esse teste verifica se a instância de diagnóstico do Amazon EC2 está na mesma VPC da instância de replicação. Se a instância de diagnóstico do Amazon EC2 não estiver na mesma VPC que a instância de replicação, encerre-a e crie-a novamente na VPC correta. Não é possível alterar a VPC de uma instância do Amazon EC2 depois de criá-la.

Testes de perda de pacotes de rede

Esse teste envia 10 pacotes para os seguintes endpoints e verifica se há perda de pacotes:

- O serviço de metadados do AWS DMS Amazon EC2 na porta 80
- O endpoint de origem
- O endpoint de destino

Todos os pacotes devem chegar com êxito. Se um pacote for perdido, consulte um engenheiro de rede para determinar o problema e encontrar uma solução.

Teste de latência da rede

Esse teste envia 10 pacotes para os mesmos endpoints do teste anterior e verifica se há latência de pacote. Todos os pacotes devem ter uma latência de menos de 100 milissegundos. Se um pacote tiver uma latência maior que 100 milissegundos, consulte um engenheiro de rede para determinar o problema e encontrar uma solução.

Teste de tamanho da unidade de transmissão máxima (MTU)

Esse teste detecta o tamanho da MTU utilizando a ferramenta Traceroute nos mesmos endpoints do teste anterior. Todos os pacotes no teste devem ter o mesmo tamanho de MTU. Se um pacote tiver um tamanho de MTU diferente, consulte um especialista do sistema para determinar o problema e encontrar uma solução.

IDs de AMIs por região

Para ver uma lista das AMIs de diagnóstico do DMS disponíveis na sua AWS região, execute o seguinte exemplo de CLI AWS .

```
aws ec2 describe-images --owners 343299325021 --filters "Name=name, Values=DMS-DIAG*"
 --query "sort_by(Images, &CreationDate)[-1].[Name, ImageId, CreationDate]" --output
 text
```

Se a saída não mostrar resultados, significa que a AMI de diagnóstico do DMS não está disponível na sua AWS região. A solução alternativa é seguir as etapas abaixo para copiar a AMI de diagnóstico de outra região. Para obter mais informações, consulte Copiar uma AMI.

- Execute uma instância na região disponível.
- Crie a imagem. A imagem será de sua propriedade.
- Copie a AMI para sua região, por exemplo, região do Oriente Médio (EAU).
- Execute a instância na sua região local.

Referência do AWS DMS

Nesta seção de referência, você encontra informações adicionais que podem ser necessárias ao utilizar o AWS Database Migration Service (AWS DMS), incluindo informações sobre conversões de tipos de dados.

O AWS DMS mantém os tipos de dados quando você faz uma migração de banco de dados homogênea em que a origem e o destino utilizam o mesmo tipo de mecanismo. Ao realizar uma migração heterogênea, em que migra de um tipo de mecanismo de banco de dados para outro, os tipos de dados são convertidos em um tipo de dados intermediário. Para ver como os tipos de dados são exibidos no banco de dados de destino, consulte as tabelas de tipo de dados para os mecanismos de bancos de dados de origem e de destino.

Lembre-se de que existem informações importantes sobre tipos de dados ao migrar um banco de dados:

- O tipo de dados FLOAT é essencialmente uma aproximação. Ao inserir um valor específico em FLOAT, ele pode ser representado de forma diferente no banco de dados. Essa diferença existe porque FLOAT não é um tipo de dados exato, como um tipo de dados decimal como NUMBER ou NUMBER(p,s). Como resultado, o valor interno de FLOAT armazenado no banco de dados pode ser diferente do valor que você inserir. Assim, o valor migrado de um FLOAT pode não corresponder exatamente ao valor no banco de dados de origem.

  Para obter mais informações sobre isso, consulte os seguintes artigos:

  - Ponto flutuante IEEE na Wikipedia
  - Representação do ponto flutuante IEEE no Microsoft Learn
  - Por que ps números de pontos flutuantes podem perder a precisão no Microsoft Learn

Tópicos

Tipos de dados do AWS Database Migration Service

O AWS Database Migration Service os utiliza tipos de dados integrados para migrar dados de um tipo de mecanismo de banco de dados de origem para um tipo de mecanismo de destino. A tabela a seguir mostra os tipos de dados internos e suas descrições.

Tipos de dados do AWS DMS	Descrição
STRING	Uma string de caracteres.
WSTRING	Uma string de caracteres de dois bytes.
BOOLEAN	Um valor booleano.
BYTE	Um valor binário.
DATE	Um valor de data: ano, mês, dia.
TIME	Um valor de tempo: horas, minutos, segundos.
DATETIME	Um valor de time stamp: ano, mês, dia, horas, minutos, segundos, frações de segundos. As frações de segundo têm escala máxima de 9 dígitos. O seguinte formato é compatível: AAAA:MM:DD HH:MM:SS.F(9). Para o Amazon S3 Select e o Amazon S3 Glacier Select, o formato do tipo de dados de DATETIME é diferente. Para obter mais informações, consulte a descrição do tipo de dados primitivo de `timestamp` em Tipos de dados compatíveis do Guia do usuário do Amazon Simple Storage Service.
INT1	Um valor de um byte, inteiro e com sinal.
INT2	Um valor de dois bytes, inteiro e com sinal.
INT4	Um valor de quatro bytes, inteiro e com sinal.
INT8	Um valor de oito bytes, inteiro e com sinal.
NUMERIC	Um valor numérico exato com precisão e escala fixas.

Tipos de dados do AWS DMS	Descrição
REAL4	Um valor com ponto flutuante e precisão simples.
REAL8	Um valor com ponto flutuante e precisão dupla.
UINT1	Um valor de um byte, inteiro e sem sinal.
UINT2	Um valor de dois bytes, inteiro e sem sinal.
UINT4	Um valor de quatro bytes, inteiro e sem sinal.
UINT8	Um valor de oito bytes, inteiro e sem sinal.
BLOB	Objeto grande binário.
CLOB	Objeto grande de caracteres.
NCLOB	Objeto grande de caracteres nativos.

> **Note**
>
> O AWS DMS não pode migrar nenhum tipo de dados LOB para um endpoint do Apache Kafka.

AWS Notas de versão do DMS

A seguir, você encontrará notas de lançamento das versões atuais e anteriores do AWS Database Migration Service (AWS DMS).

AWS DMS não diferencia entre versões principais e secundárias quando você ativa a atualização automática de versões para sua instância de replicação. O DMS atualiza automaticamente a versão da instância de replicação durante a janela de manutenção se a versão for descontinuada.

Observe que, para atualizar a versão da sua instância de replicação manualmente (usando a API ou a CLI) da versão 3.4.x para a 3.5.x, você deve definir o parâmetro como. `AllowMajorVersionUpgrade` `true` Para obter informações sobre o `AllowMajorVersionUpgrade` parâmetro, consulte ModifyReplicationInstancea documentação da API DMS.

> **Note**
>
> A versão atual do mecanismo padrão para AWS DMS é 3.5.1.

A tabela a seguir mostra as seguintes datas para as versões ativas do DMS:

- A data de lançamento da versão
- A data após a qual você não pode criar novas instâncias com a versão
- A data em que o DMS atualiza automaticamente as instâncias dessa versão (a data de EOL)

Version (Versão)	Data de lançamento	Nenhuma nova data de instância	Data de EOL
3.5.3	17 de maio de 2024	31 de agosto de 2025	31 de outubro de 2025
3.5.2	29 de outubro de 2023	30 de março de 2025	29 de abril de 2025
3.5.1	30 de junho de 2023	30 de novembro de 2024	30 de janeiro de 2025

Version (Versão)	Data de lançamento	Nenhuma nova data de instância	Data de EOL
3.4.7	31 de maio de 2022	30 de julho de 2024	29 de agosto de 2024
3.4.6	30 de novembro de 2021	26 de maio de 2024	27 de junho de 2024

AWS Notas de versão do Database Migration Service 3.5.3

Novos recursos na AWS DMS versão 3.5.3

Novo recurso ou aprimoramento	Descrição
Endpoint de origem PostgreSQL aprimorado para suporte ao Babelfish	AWS DMS aprimorou seu endpoint de origem PostgreSQL para oferecer suporte aos tipos de dados Babelfish. Para ter mais informações, consulte Utilizar o banco de dados PostgreSQL como origem do AWS DMS.
Support para S3 Parquet como fonte	AWS DMS suporta S3 Parquet como fonte. Para obter mais informações, consulte Usando o Amazon S3 como fonte para AWS DMS.
Support para PostgreSQL 16.x	AWS DMS suporta a versão 16.x do PostgreSQL. Para obter mais informações, consulte Utilizar o banco de dados PostgreSQL como origem do AWS DMS e Utilizar um banco de dados PostgreSQL como destino do AWS Database Migration Service.
Taxa de transferência aprimorada para migrações de carga completa de Oracle para Amazon Redshift	AWS DMS O Serverless fornece um desempenho de taxa de transferência significativamente aprimorado para migrações de carga total da Oracle para o Amazon Redshift. Para ter mais informações, consulte Taxa de transferência aprimorada para migrações de carga completa de Oracle para Amazon Redshift.

AWS DMS a versão 3.5.3 inclui os seguintes problemas resolvidos:

Problemas resolvidos na versão 3.5.3 do DMS, datada de 17 de maio de 2024

Problema resolvido	Descrição
Função de substituição de validação de dados	Corrigido um problema no recurso de validação de dados em que o DMS não respeitava a filtragem de origem quando uma ação de regra era definida `override-validation-function` em mapeamentos de tabela.
Erros do CDC de origem do MySQL	Corrigido um problema no MySQL como fonte em que a migração do CDC falhava com a codificação UTF-16.
Diferenças de comparação de validação de dados	Corrigido um problema no recurso de validação de dados em que o DMS não aplicava adequadamente a configuração da `HandleCollationDiff` tarefa quando a filtragem de colunas era usada.
Tarefa de validação de dados suspensa.	Corrigido um problema no recurso de validação de dados em que a tarefa do DMS travava com um erro “`target`é nulo”.
Falhas de tarefas na replicação do PostgreSQL para o PostgreSQL.	Corrigido um problema nas migrações de PostgreSQL para PostgreSQL em que uma tarefa do DMS falhava ao inserir dados LOB no destino durante a replicação do CDC.
Perda de dados com o PostgreSQL como fonte	Corrigido um problema no PostgreSQL como fonte em que a perda de dados ocorria em determinados cenários extremos.
Erros de CDC de origem do MySQL 5.5	Corrigido um problema no MySQL como fonte em que a replicação do CDC falhava com a versão 5.5 do MySQL.
Problema na tabela IOT de origem Oracle.	Corrigido um problema no Oracle como fonte em que o DMS não replicava as `UPDATE` instruções corretamente para tabelas de IOT com o registro suplementar ativado em todas as colunas.
LOBS de origem MySQL	Corrigido um problema nas migrações do MySQL para o Redshift em que a tarefa do DMS falhava devido a LOBs excederem o tamanho máximo permitido pelo Redshift.
Problema de validação com `SkipLobColumns`	Corrigido um problema no recurso de validação de dados em que a tarefa do DMS falhava `SkipLobColumns = true` quando uma chave primária estava na última coluna da tabela de origem.

Problema resolvido	Descrição
Ignore a validação onde a chave exclusiva está `null`	Corrigido um problema no recurso de validação de dados em que o DMS não pulava linhas com chaves exclusivas nulas corretamente.
Melhorias na validação de dados para o `COLLATE` operador Oracle.	Corrigido um problema no recurso de validação de dados em que a validação falhava com um erro de sintaxe nas versões do Oracle anteriores à 12.2.
Tratamento de erros durante a carga total	Corrigido um problema no PostgreSQL como destino em que a tarefa travava durante a fase de carregamento total após um erro de tabela causado por dados inválidos.
Revalidação de tarefas somente de validação do CDC	Aprimorou o recurso de validação de dados para permitir a revalidação em uma tarefa somente de validação do CDC.
S3 como um problema alvo `CdcMaxBatchInterval Out of Memory`	Corrigido um problema no S3 como destino em que a tarefa do DMS falhava com uma condição de falta de memória definida. CdcMaxBatchInterval
Driver de origem Oracle	Atualizou o driver de origem do DMS Oracle da v12.2 para a v19.18.
Aviso de truncamento de LOB com fonte do SQL Server	Registro aprimorado para o SQL Server como fonte para mostrar avisos sobre truncamento de LOB durante o CDC.
Aprimoramentos do leitor binário Oracle	O leitor binário de origem Oracle foi aprimorado para oferecer suporte ao seguinte: • Plataforma Big Endian • Dicas paralelas de DML com compressão HCC • Compressões Oracle avançadas com Golden Gate habilitado

AWS Notas de versão do Database Migration Service 3.5.2

Novos recursos na AWS DMS versão 3.5.2

Novo recurso ou aprimoramento	Descrição
Validação de dados do Redshift	AWS DMS agora oferece suporte à validação de dados em destinos do Redshift.
Compatibilidade adicionada para o Microsoft SQL Server versão 2022 como origem e destino.	AWS DMS agora oferece suporte ao uso do Microsoft SQL Server versão 2022 como origem e destino.
IBM Db2 LUW como destino	AWS DMS agora oferece suporte ao IBM Db2 LUW como destino. Usando AWS DMS, agora você pode realizar migrações ao vivo do IBM Db2 LUW para o IBM Db2 LUW.

AWS DMS a versão 3.5.2 inclui os seguintes problemas resolvidos:

Problemas resolvidos na versão de manutenção do DMS 3.5.2, datada de 29 de abril de 2024

Problema resolvido	Descrição
O IBM Db2 tem como alvo a carga total segmentada	Suporte adicional para carga total segmentada com o IBM Db2 como destino.
Amazon Timestream como configurações de destino	Melhorou o tratamento de configurações de carimbo de data/hora inválidas e operações de tabela não suportadas para Timestream como destino.
Falha na tarefa com filtro de coluna	Corrigido um problema em que uma tarefa falhava ao usar um filtro em uma coluna que o DMS adicionava dinamicamente usando uma regra de transformação.
Registrando a leitura do arquivo de troca de transações	Registro adicionado para mostrar quando o DMS está lendo arquivos de troca de transações.

Problema resolvido	Descrição
S3 como alvo com `CdcInsertsAndUpdates`	Corrigido um problema no S3 como alvo em que uma tarefa travava quando `CdcInsertsAndUpdates` está `true` e `PreserveTransactions` está `true`.
Operadores negativos do filtro de origem	Corrigido um problema em que o operador de filtro de origem, quando definido como um operador negativo, apresentava um comportamento incorreto se a mesma coluna tivesse uma regra de transformação definida.
Registro adicionado para quando o DMS pausa a leitura da fonte	Registro aprimorado para mostrar quando o DMS pausa temporariamente a leitura da fonte para melhorar o desempenho.
Filtros de origem com caracteres escapados	Corrigido um problema nos filtros de origem em que o DMS aplica caracteres de escape às tabelas recém-criadas durante o CDC.
PostgreSQL como destino, exclusões replicadas incorretamente	Corrigido um problema no PostgreSQL como destino em que o DMS replica as exclusões como valores nulos.
Oracle como fonte: melhorias no registro	Registro aprimorado para Oracle como fonte para remover códigos de erro estranhos.
Registro aprimorado das limitações do XMLTYPE	Registro aprimorado para Oracle como fonte para mostrar a falta de suporte do DMS para o modo LOB completo para o `XMLTYPE` tipo de dados.
Perda de dados do MySQL	Corrigido um problema no MySQL como destino em que metadados de coluna corrompidos podiam causar falhas na tarefa ou perda de dados.
Filtro aplicado a uma nova coluna	Corrigido um problema durante o carregamento total em que o DMS ignora um filtro que uma regra de transformação adiciona a uma nova coluna.

Problema resolvido	Descrição
S3 como alvo: problema de validação	Corrigido um problema no S3 como destino em que a validação de dados falhava ao migrar várias tabelas com diferentes definições de particionamento de validação.
Falha na tarefa somente em CDC	Corrigido um problema em tarefas somente de CDC em que a tarefa travava quando `TaskRecoveryTableEnabled` estava. `true`
Agrupamentos incompatíveis de MySQL para MariaDB	Corrigido um problema nas migrações do MySQL para o MariaDB em que o DMS não migra tabelas do MySQL v8 com agrupamento. `tf8mb4_0900_ai_ci`
A tarefa falha com `BatchApplyEnabled`	Corrigido um problema no recurso Batch Apply em que a tarefa falhava sob determinadas condições.
Caracteres não UTF-8 no Amazon DocumentDB	Foi adicionado suporte para caracteres não UTF-8 para endpoints do Amazon DocumentDB.
Falha na tarefa Batch Apply	Corrigido um problema no recurso Batch Apply em que a tarefa do DMS falhava ao replicar grandes transações.
Tratamento de reversão de transações do Db2	Corrigido um problema do Db2 como fonte em que o DMS replicava um `INSERT` para o destino, apesar de ter sido revertido na origem.
Validação com filtros de origem	Corrigido um problema em que a validação não respeitava os filtros de origem.

AWS Notas de versão do Database Migration Service 3.5.1

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos no AWS Database Migration Service (AWS DMS) versão 3.5.1.

Novo recurso ou aprimoramento	Descrição
Compatibilidade com o PostgreSQL 15.x	AWS DMS a versão 3.5.1 suporta a versão 15.x do PostgreSQL. Para obter mais informações, consulte Usar o PostgreSQL como origem e Usar o PostgreSQL como destino.
Compatibilidade com o Amazon DocumentDB Elastic Clusters com coleções fragmentadas	AWS DMS a versão 3.5.1 é compatível com Amazon DocumentDB Elastic Clusters com coleções fragmentadas. Para ter mais informações, consulte Utilizar o Amazon DocumentDB como destino para o AWS Database Migration Service.
Redshift sem servidor como destino	Compatibilidade com a utilização do Amazon Redshift sem servidor como endpoint de destino. Para ter mais informações, consulte Utilizar um banco de dados Amazon Redshift como destino do AWS Database Migration Service.
Configurações do endpoint do Babelfish	Configurações aprimoradas do endpoint de destino do PostgreSQL para fornecer suporte ao Babelfish. Para ter mais informações, consulte Utilizar um banco de dados PostgreSQL como destino do AWS Database Migration Service.
Transações abertas de origem Oracle	AWS DMS 3.5.1 melhora a metodologia de lidar com transações abertas ao iniciar uma tarefa somente de CDC a partir da posição inicial de uma fonte Oracle. Para obter mais informações, consulte `OpenTransactionWindow` na seção Configurações de endpoint ao usar o Oracle como fonte para AWS DMS.
Amazon Timestream como alvo	Support para usar o Amazon Timestream como um endpoint de destino. Para ter mais informações, consulte Utilizar o Amazon Timestream como destino para o AWS Database Migration Service.

AWS DMS a versão 3.5.1 inclui os seguintes problemas resolvidos:

Problema resolvido	Descrição
Oracle como fonte de crescimento de sessões inativas	Corrigido um problema na fonte Oracle em que tarefas somente CDC tinham sessões inativas em crescimento contínuo, resultando na seguinte exceção:. `ORA-00020: maximum number of processes exceeded on the source database`
Replicando as alterações do UPDATE no DocumentDB	Corrigido um problema no DocumentDB como um destino em que as instruções UPDATE não eram replicadas adequadamente em alguns cenários.
Tarefa somente de validação	Tratamento aprimorado de erros para que o recurso de validação de dados falhe adequadamente na tarefa quando a validação de dados está desativada para tarefas somente de validação.
Replicação do Redshift após o término da conexão	Corrigido um problema no destino do Redshift em que a tarefa do DMS não tentava aplicar novamente as alterações no destino quando o destino fosse `ParallelApplyThreads` definido como maior que zero após o término da conexão, o que resultaria em perda de dados.
Replicação de texto para texto médio do MySQL	Corrigido um problema na replicação de tipos de dados de texto médio de MySQL para MySQL com o modo Full-lob.
A tarefa do CDC não está sendo replicada com segredo rotacionado	Corrigido um problema nas tarefas do DMS com a `BatchApplyEnabled` configuração de `true` onde o DMS pararia de replicar dados depois que o Secrets Manager alterasse a senha.
Problema de segmentação do MongoDB/DocumentDB	Corrigido um problema na fonte MongoDB/DocDB em que a segmentação de intervalo não funcionava corretamente quando a coluna da chave primária continha um valor grande.
Validação de dados Oracle de valores numéricos não vinculados	Corrigido um problema no Oracle Target em que o DMS reconhecia um valor do tipo de dados não vinculado `NUMERIC` como a `STRING` durante a validação de dados.
Validação de dados do SQL Server	Corrigido um problema nos endpoints do SQL Server em que a validação de dados do DMS criava uma instrução SQL inválida.

Problema resolvido	Descrição
Segmentação automática do MongoDB	A funcionalidade do particionamento automático de dados foi aprimorada ao migrar documentos em paralelo do MongoDB como origem.
Formato do Amazon S3 Apache Parquet	Corrigido um problema para que os arquivos do Apache Parquet gravados no S3 como destino possam ser visualizados com o Python com o Apache Arrow C++.
PostgreSQL como tratamento da DDL de origem	Corrigido um problema com a origem PostgreSQL em que operações DDL incompatíveis não eram ignoradas adequadam ente.
Erro de dados `timestamp tz` do PostgreSQL	Corrigido um problema com migrações do PostgreSQL para PostgreSQL em que o timestamp com dados de fuso horário não era migrado corretamente com a aplicação em lote ativada durante a CDC.
Falha na validação do Oracle para o PostgreSQL	Corrigido um problema com migrações do Oracle para o PostgreSQ L em que a validação de dados falhava para o tipo de dados NUMERIC(38,30).
Erro de tipo de dados estendido do Oracle	Corrigido um problema com a origem Oracle em que o tipo de dados varchar estendido estava sendo truncado.
Combinação de operadore s de filtro	Corrigido um problema na funcionalidade da filtragem de colunas em que o operador de coluna nula não podia ser combinado com outros tipos de operadores.
Latência de CDC resultant e do registro em log excessivo.	Corrigido um problema com a origem PostgreSQL em que o registro em log excessivo de avisos do plug-in pglogical causava a latência na CDC de origem.
Tratamento de replicaçã o bidirecional do Create Table DDL	Corrigido um problema na replicação bidirecional do PostgreSQL para o PostgreSQL em que a alteração Create Table DDL não era replicada corretamente.

Problema resolvido	Descrição
Falha da CDC ao utilizar filtros	Corrigido um problema no recurso de filtragem em que a replicação da CDC estava falhando.
Validação de nome de host da autoridade de certificação de endpoints Kafka	Aprimorou a funcionalidade dos endpoints Kafka adicionando a opção de desativar a validação de nome de host da autoridade de certificação (`SslEndpointIdentificationAlgorithm`).
Validação do IBM Db2 LUW	Corrigido um problema em que os tipos de dados data, timestamp e hora de origem Db2 LUW não eram tratados adequadamente durante a validação de dados.
Validação do S3	Corrigido um problema com migrações do Db2 LUW para o S3 em que a funcionalidade de validação não estava tratando o tipo de dados timestamp(0) de forma adequada.
Falha na reinicialização da tarefa do DMS	Corrigido um problema com a fonte do PostgreSQL em que AWS DMS a tarefa falhava ao reiniciar e não podia consumir eventos relacionais ao usar o plug-in pglogical.
Validação do tipo de dados HIERARCHY do SQL Server	Corrigido um problema na origem SQL Server em que a validação do tipo de dados HIERARCHY falhava.
Strings de caracteres do SQL Server com caracteres de controle	Corrigido um problema na origem SQL Server em que as strings com caracteres de controle não eram replicadas corretamente.
Redshift com Secrets Manager	Corrigido um problema com o destino Redshift em que o teste do endpoint falhava ao utilizar o Secrets Manager.
Inconsistência na configuração do MySQL ParallelLoadThreads	Corrigido um problema com o destino MySQL em que a configuração de `ParallelLoadThreads` não era retida adequadamente após alterações nas configurações da tarefa.
Erro com o mapeamento de tipos de dados do PostgreSQL para o Oracle	Corrigido um problema com migrações do PostgreSQL para o Oracle em que a tarefa falhava ao replicar do tipo de dados TEXT para o tipo de dados VARCHAR2(2000).

Problema resolvido	Descrição
Validação de dados do Oracle para o PostgreSQL	Corrigido um problema com migrações do Oracle para o PostgreSQL em que a validação de dados relatava falsos positivos quando caracteres NULL eram replicados como caracteres SPACE.
Fonte do SQL Server na AlwaysOn configuração	Corrigido um problema com a fonte do SQL Server na AlwaysOn configuração em que a AWS DMS tarefa falhava quando o nome da réplica não correspondia exatamente ao nome real do servidor.
Falha no teste do endpoint de origem Oracle	Corrigido um problema com a fonte Oracle em que o teste de conexão do AWS DMS endpoint falhava devido a privilégios insuficientes ao recuperar o ID de sessão (SID) do Oracle.
A CDC não seleciona novas tabelas	Corrigido um problema com tarefas somente de CDC em que as tabelas criadas na origem após o início da tarefa não eram replicadas em alguns casos.
Transações abertas no Oracle como origem	Aprimorada a metodologia de tratamento de transações abertas ao iniciar uma tarefa somente de CDC da posição inicial de uma origem Oracle.
Problema de dados ausentes	Corrigido um problema de dados ausentes ao retomar uma tarefa se ela fosse interrompida após a aplicação de alterações em cache (opção `StopTaskCachedChangesApplied` definida como verdadeira). Esse problema pode ocorrer raramente se as alterações em cache AWS DMS persistirem no disco da instância AWS DMS de replicação devido a um alto volume de alterações na origem.
Problema de validação de dados no tipo de dados estendido	Corrigido um problema na validação de dados do PostgreSQL para o Oracle em que a validação falhava para tipos de dados estendidos.
Problema de validação de dados na codificação inconsistente de caracteres	Corrigido um problema na validação de dados do SQL Server para o PostgreSQL em que a validação falhava quando a codificação de caracteres era inconsistente entre a origem e o destino.

Problema resolvido	Descrição
Problema de validação de dados ORA-01455	Corrigido um problema em que um erro ORA-01455 ocorria durante a validação quando um `integer` do PostgreSQL era mapeado para um `number(10)` do Oracle.
Compatibilidade com a `IDENTITY` do SQL Server	Corrigido um problema na replicação de dados do SQL Server para o SQL Server em que a migração de colunas de identidade falhava quando a coluna de destino tinha a propriedade `IDENTITY`.
Problema de conjunto de caracteres com instruções ALTER	Corrigido um problema na replicação do MySQL para o MySQL em que o conjunto de caracteres era AWS DMS alterado para UTF16 ao migrar uma instrução durante o CDC. `ALTER`
Suporte a tipos de dados espaciais do PostgreSQL para o Redshift	Adicionado suporte para o tipo de dados `spatial` ao migrar do PostgreSQL para o Amazon Redshift.
Compressão GZIP de arquivos .parquet	Corrigido um problema em que AWS DMS não era possível gerar arquivos.parquet com compactação GZIP com o S3 como destino.
Migração de origem MongoDB/DocDB	Corrigido um problema em AWS DMS que algumas das partições não eram migradas de uma fonte do MongoDB.
Problema de estatísticas da tabela	Corrigido um problema em que as estatísticas da tabela não eram mostradas quando pelo menos uma das tarefas na instância de replicação continha mais de 1001 tabelas.
Tabela suspensa para as versões 10.1.0 e inferiores do IBM Db2 LUW	Corrigido um problema na origem Db2 LUW em que a migração de tabela era suspensa com o erro `TYPESTRINGUNITS is not valid` quando a versão do banco de dados de origem era 10.1.0 ou inferior.
Problema de particion amento do MongoDB	Corrigido um problema no MongoDB/DocDB em que um ou mais segmentos da partição de origem estavam ausentes.
Problema de particion amento do MongoDB	Corrige um problema em que a segmentação com base em uma coluna com o tipo NumberLong () falha devido a um erro de conversão de tipo.

Problema resolvido	Descrição
Problema de particion amento do MongoDB	Desempenho de segmentação automática melhorado para grandes conjuntos de dados com o MongoDB como origem.
Versão do driver do MongoDB	O driver do MongoDB foi rebaixado para 1.20.0 para continuação da compatibilidade com o MongoDB versões 3.6 e inferior.
Tipo de dados timestamp do Amazon S3 Apache Parquet	Corrigido um problema no alvo de parquete do Amazon S3. AWS DMS agora define o parâmetro de formato `isAdjustedToUTC` `true` para corresponder ao comportamento nas versões anteriores do AWS DMS.
Comando copy do Amazon Redshift como destino	Corrigido um problema no Amazon Redshift como destino em que o comando copy falhava em tabelas grandes ao copiar dados do Amazon S3 para o Amazon Redshift.
Tipos de dados de geometria do PostgreSQL	Corrigido um problema nas migrações do PostgreSQL para o PostgreSQL em que a migração falhava em tipos de dados de geometria grande.
XML no Oracle para o PostgreSQL	Corrigido um problema em que a migração adicionava um espaço extra no XML ao replicar do Oracle para o PostgreSQL.
Atualização do ponto de verificação de destino em mecanismos compatíveis	AWS DMS agora atualiza o ponto de verificação de destino na `awsdms_txn_state` tabela no banco de dados de destino.
Registros do MongoDB/ DocDB enviados para a coleção incorreta	Corrigido um problema no MongoDB/DocDB em que os dados eram enviados para a coleção de destino incorreta.
Seleção de nova tabela de origem Oracle com configuração EscapeCha racter de endpoint	Corrigido um problema na fonte Oracle em que só AWS DMS pegava novas tabelas para replicação quando a tarefa era interrompida e retomada enquanto a configuração do `EscapeCha racter` endpoint estava definida.

Problema resolvido	Descrição
Ponto de verificação de recuperação da CDC	Corrigida uma inconsistência no ponto de verificação de recuperação da CDC observada entre o datastore de destino e o console do AWS DMS .
Tarefas somente de validação da CDC	Correção de um problema com tarefas somente de validação da CDC em que a tarefa não falhava mesmo que todas as tabelas da tarefa apresentassem falhas.
Comportamento da validação com problemas de conexão de origem ou de destino	Corrigido um problema com a validação de dados em AWS DMS que suspendia as tabelas na origem ou no destino quando a conexão era interrompida.
Falsos positivos da validação de dados do Oracle para o PostgreSQL	Corrigido um problema com a validação de dados do Oracle para PostgreSQL em AWS DMS que foram relatados falsos positivos. Isso ocorria porque as diferenças na representação de caracteres NULL da origem no destino não eram consideradas com tipos de dados baseados em texto diferentes de VARCHAR.
Truncamento de dados do Oracle para o PostgreSQL	Correção de um problema com o Oracle como origem e o PostgreSQL como destino em que o AWS DMS estava truncando dados de colunas NVARCHAR com a configuração `NLS_NCHAR_CHARACTERSET` do Oracle definida como `AL16UTF16` .
Erro na validação de dados	Corrigido um problema na validação de dados em que um erro `unable to create where filter clause` era gerado quando uma filtragem da origem e uma regra de transformação de adição de coluna estavam em uso.
Tratamento de erros no destino Redshift	Corrigido um problema no Redshift como destino em que o tratamento de erros não funcionava conforme configurado quando a tarefa da CDC tinha a configuração da tarefa `ParallelApplyThreads` definida como um valor maior que zero.

Problema resolvido	Descrição
Falha de comunicação do Oracle como origem	Corrigido um problema no Oracle como origem em que a tarefa permanecia no estado `RUNNING`, mas não conseguia migrar nenhum dado após uma falha de comunicação.
Tabela da CDC suspensa com filtros de coluna	Corrigido um problema com tarefas de carga máxima + CDC em que uma tabela era suspensa durante a fase de CDC quando filtros de coluna eram aplicados.
Falha de caracteres especiais na validação de dados do S3 como destino	Corrigido um problema com a validação de dados de destino do S3 em que a tarefa falhava se o nome da tabela incluísse um caractere especial diferente de um sublinhado.
Falha de carga máxima e de CDC da origem MongoDB	Corrigido um problema com o MongoDB como origem em que uma tarefa de carga máxima + CDC falhava durante o tratamento de eventos em cache ao migrar uma coleção grande.
Problema de atualização com `BatchApplyEnabled` definida como verdadeira	Corrigido um problema em que uma tarefa com a configuração de `BatchApplyEnabled` tarefa definida como verdadeira falhava após a migração da AWS DMS versão 3.4.6 para a 3.5.1 em alguns casos.
AlwaysOn Fonte do SQL Server com agrupamento com distinção entre maiúsculas e minúsculas	Corrigido um problema com o SQL Server AlwaysOn como fonte em que uma tarefa falhava com agrupamento com distinção entre maiúsculas e minúsculas.
Tarefa de origem MySQL suspensa	Corrigido um problema com o MySQL como origem em que uma tarefa era suspensa em vez de falhar quando a origem não estava configurada corretamente.
Falha na tarefa de carga máxima da origem S3	Corrigido um problema com o S3 como fonte em que uma tarefa falhava ao ser retomada após a atualização da AWS DMS versão 3.4.6 ou 3.4.7 para a versão 3.5.1.

Problema resolvido	Descrição
Origem do PostgreSQL com CaptureDDLs definidas como falsas	Corrigido um problema com o PostgreSQL como origem em que as DDLs não eram tratadas adequadamente com a configuração do endpoint do `CaptureDDLs` definida como falsa.
Falha na tarefa de origem Oracle durante retomada	Corrigido um problema com o Oracle como origem em que uma tarefa falhava ao ser retomada devido a dados incorretos no nome da coluna.
Falha na pesquisa de LOB na origem MySQL	Corrigido um problema com o MySQL como origem em que uma pesquisa de LOB falhava quando a configuração da tarefa `ParallelApplyThreads` estava definida como um valor maior que zero.
Erro de LSN ilógico na origem SQL Server	Corrigido um problema com o SQL Server como fonte em que uma tarefa falhava com um `illogical LSN sequencing state error` erro após a atualização da AWS DMS versão 3.4.7 para a versão 3.5.1.
Origem do PostgreSQL com pglogical	Corrigido um problema com o PostgreSQL como origem em que uma tarefa que utilizava o plug-in pglogical falhava quando a tarefa era interrompida, uma tabela era removida das regras de seleção, a tarefa era retomada e eram feitas alterações na tabela removida.
Ponto de verificação de recuperação incorreto do Aurora MySQL	Correção de um problema do Aurora MySQL como origem em que um ponto de verificação de recuperação incorreto era salvo em decorrência de um failover do Aurora ou da parada e início da origem do Aurora.
Falha de tarefa no SQL Server como origem	Correção de um problema do SQL Server como origem em que uma tarefa apresentava falha quando `SafeguardPolicy` era definida como `RELY_ON_SQL_SERVER_REPLICATION_AGENT` .
Conversão de tipo de dados incorreta com o MySQL como destino	Correção de um problema no MySQL como destino em que a replicação de CDC apresentava falha em decorrência da conversão incorreta do tipo de dados na fase de aplicação em lote.

Problema resolvido	Descrição
Falha na tarefa com captureDDLS definido como falso para o PostgreSQL como origem	Correção de um problema no PostgreSQL como origem em que uma tarefa apresentava falha em decorrência do tratamento de uma DDL como DML quando a configuração `CaptureDDLs` do endpoint era definida como `false`.
Falha de coleção vazia do MongoDB	Correção de um problema do MongoDB como origem em que a tarefa apresentava falha em decorrência de uma coleção vazia.
Falha na tarefa de carga máxima no Redshift como destino	Correção de um problema no Redshift como destino em que uma tarefa travava durante a fase de carregamento completo quando a tabela de controle do ponto de verificação de recuperação estava habilitada.
S3 para S3: sem movimentação de dados	Corrigido um problema na replicação de S3 para S3 em AWS DMS que não replicava os dados se não fossem especific adosbucketFolder .
Latência da CDC com `GlueCatalogGeneration` definido como `true`	Correção de um problema no S3 como destino em que ocorria latência excessiva quando `GlueCatalogGeneration` era definido como `true`.
Truncamento de dados no Oracle como destino	Corrigido um problema com o Oracle como destino em que os dados são AWS DMS truncados nas colunas VARCHAR2.
Comportamento do caractere curinga no PostgreSQL	Correção de um problema no PostgreSQL como origem em que o comportamento do curinga “_” nas regras de seleção não estava funcionando conforme documentado.
Problema de cabeçalho WAL vazio no PostgreSQL como origem	Correção de um problema do PostgreSQL como origem em que a tarefa apresentava falha em decorrência de um cabeçalho WAL vazio recebido do slot de replicação.
MySQL ou MariaDB como origem com logs binários compactados	Corrigido um problema no MySQL e no MariaDB como fontes em que uma mensagem de erro adequada não era emitida quando a compressão BINLOG foi detectada. AWS DMS

Problema resolvido	Descrição
Caracteres especiais de validação de dados do S3	Validação de dados aprimorada do S3 para lidar com caracteres especiais em colunas de chave primária e não primária.
Entradas enganosas de log de tarefas com o Redshift como destino	Correção de um problema no Redshift como destino em que entradas enganosas estavam presentes no log de tarefas e relatavam falhas na instrução de aplicação em lote em ATUALIZAÇÕES e EXCLUSÕES.
Falha na tarefa de migração do SQL Server para o S3	Correção de um problema nas migrações do SQL Server para o S3 em que a tarefa apresentava falha ao aplicar alterações armazenadas em cache.
Dados ausentes sobre erros de aplicação em lote	Correção de um problema no recurso de aplicação em lote em que um erro na aplicação de um lote resultava na ausência de dados.

AWS Notas da versão beta do Database Migration Service 3.5.0

> **Important**
>
> AWS DMS 3.5.0 é uma versão beta do mecanismo de instância de replicação. AWS DMS suporta esta versão da mesma forma que todas as versões anteriores. Mas recomendamos que você teste a versão AWS DMS 3.5.0 Beta antes de usá-la para fins de produção.

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos na versão 3.5.0 Beta do AWS Database Migration Service (AWS DMS).

Novo recurso ou aprimoramento	Descrição
Time Travel para Oracle e o Microsoft SQL Server	Agora você pode usar o Time Travel em todas as AWS regiões com endpoints de origem Oracle, Microsoft SQL Server e PostgreSQL compatíveis com DMS e endpoints de destino PostgreSQL e MySQL compatíveis com DMS.

Novo recurso ou aprimoramento	Descrição
Validação do S3	AWS DMS agora oferece suporte à validação de dados replicados nos endpoints de destino do Amazon S3. Para obter informações sobre como validar os dados de destino do Amazon S3, consulte Validação de dados de destino do Amazon S3.
Integração do Glue Catalog	AWS Glue é um serviço que fornece maneiras simples de categorizar dados e consiste em um repositório de metadados conhecido como. AWS Glue Data Catalog Agora você pode integrar e AWS Glue Data Catalog com seu endpoint de destino do Amazon S3 e consultar dados do Amazon S3 por meio de outros serviços, AWS como o Amazon Athena. Para ter mais informações, consulte Utilizar o AWS Glue Data Catalog com um destino do Amazon S3 do AWS DMS.
Aplicação paralela do DocumentDB como destino	Usando o DocumentDB como destino com novas configurações de `ParallelApply*` tarefas, AWS DMS agora suporta um máximo de 5000 registros por segundo durante a replicação do CDC. Para ter mais informações, consulte Utilizar o Amazon DocumentDB como destino para o AWS Database Migration Service.
Registro em log centrado no cliente	Agora você pode examinar e gerenciar os registros de tarefas com mais eficiência com a AWS DMS versão 3.5.0. Para obter informações sobre como visualizar e gerenciar registros de tarefas do AWS DMS, consulteVisualização e gerenciamento dos logs de tarefas do AWS.
Mecanismo SASL_PLAIN para endpoints de destino do Kafka	Agora é possível utilizar a autenticação SASL_PLAIN para suporte aos endpoints de destino do Kafka MSK.
Replicação de transações XA no MySQL	Agora é possível utilizar transações XA na origem MySQL do DMS. Antes do DMS 3.5.0, as alterações de DML aplicadas como parte das transações XA não eram replicadas corretamente.

Novo recurso ou aprimoramento	Descrição
Tipos de dados estendidos do Oracle	AWS DMS agora suporta a replicação de tipos de dados estendidos no Oracle versão 12.2 e superior.
Ambiente Db2 LUW PureScale	AWS DMS agora oferece suporte à replicação de um ambiente Db2 PureScale LUW. Essa funcionalidade só é compatível ao utilizar Iniciar processamento de alterações a partir da opção da posição de alteração de origem.
Origem SQL Server com a opção `READ_COMMITTED_SNAPSHOT`	Ao usar um banco de dados de origem do Microsoft SQL Server com a `READ_COMMITTED_SNAPSHOT` opção definida como`TRUE`, você pode replicar as alterações de DML corretamente definindo o atributo de conexão force DataRow Lookup.

AWS DMS A versão 3.5.0 inclui os seguintes problemas resolvidos:

Problemas resolvidos na AWS DMS versão 3.5.0 lançada em 17 de março de 2023

Tópico	Resolução
Oracle: comparação de caso especial de string que foi convertida de numérica	Corrigido um problema na origem Oracle em que as regras de filtragem não funcionavam conforme o esperado em uma coluna numérica quando existia a transformação do tipo de dados para string para a mesma coluna.
Aprimoramentos do SQL Server AG on-premises	Maior eficiência do tratamento de conexões com a fonte do SQL Server na AlwaysOnconfiguração, eliminando conexões desnecessárias com réplicas que não são usadas pelo DMS.
Conversão interna do SQL Server HIERARCHYID	Corrigido um problema com a origem SQL Server em que o tipo de dados HIERARCHYID era replicado como VARCHAR(250) em vez de HIERARCHYID para o destino SQL Server.

Tópico	Resolução	
Correção da tarefa de movimentação de destino S3	Corrigido um problema ao mover uma tarefa com destino S3 que demorava muito tempo, parecia congelada ou nunca era concluída.	
Mecanismo SASL Plain do Kafka	Introduzido suporte ao método de autenticação SASL Plain para o endpoint de destino do Kafka MSK.	
Falha na carga/aplicação paralela devido ao parâmetro _type com o Opensearch 2.x	Corrigido um problema no destino Opensearch 2.x em que a carga paralela ou a aplicação paralela falhava devido à falta de suporte para o parâmetro _type.	
Compatibilidade com o filtro de mapeamento de tabelas com operadores mistos	Removida uma limitação em que somente um filtro podia ser aplicado em uma coluna.	
Endpoints do S3, do Kinesis e do Kafka: a migração de colunas de lob baseadas em alter na fase CDC	Corrigido um problema nos destinos do Kinesis, do Kafka e do S3 em que os dados em colunas LOB adicionadas durante a CDC não eram replicados.	
Atualização do driver do MongoDB	O driver do MongoDB foi atualizado para a versão v1.23.2.	
Atualização do driver do Kafka	O driver do Kafka foi atualizado de 1.5.3 para 1.9.2.	
A configuração do endpoint S3 não estava funcionando corretamente	Corrigido um problema no destino S3 em que a configuração do endpoint `AddTrailingPaddingCharacter` não funcionava quando os dados continham o caractere especificado como delimitador para o destino S3.	

Tópico	Resolução	
A tarefa de destino do Kinesis falhava	Corrigido um problema no destino Kinesis em que uma tarefa falhava quando o valor de PK estava vazio e a depuração detalhada estava ativada.	
Quando os nomes das colunas dos destinos S3 eram movidos por uma posição	Corrigido um problema em um destino S3 em que os nomes das colunas eram movidos por uma posição quando `AddColumnName` estava definido como `true` e `TimestampColumnName` estava definido como `""`.	
Log de avisos aprimorado para truncamento de LOB	Log de avisos aprimorado para truncamento de LOB na origem SQL Server para incluir a instrução select usada para recuperar o LOB.	
Adição de erro fatal para evitar falhas na tarefa do DMS quando a senha da TDE está incorreta.	Introduzida uma mensagem de erro significativa e eliminado o problema de falha na tarefa em situações em que a tarefa do DMS estava falhando sem mensagem de erro devido à senha incorreta da TDE para o Oracle como origem.	
Permite a migração do DDL do PostgreSQL CTAS (Criar tabela conforme selecionado) durante a CDC.	Removidas as limitações do DMS de não poder replicar DDLs do PostgreSQL CTAS (criar tabela conforme selecionado) durante a CDC.	
Falha da tarefa de correção de pg_logical quando as colunas da tabela são descartadas na CDC.	Corrigido um problema na origem PostgreSQL com destino S3 em que as colunas eram desalinhadas no destino quando o suporte para LOBs estava desativado e LOBs estavam presentes.	

Tópico	Resolução	
Correção de vazamento de memória no tratamento de conexões do MySQL	Corrigido um problema na origem MySQL em que o consumo de memória da tarefa aumentava continuamente.	
Configuração do endpoint de origem Oracle: `ConvertTimestampWithZoneToUTC`	Defina esse atributo como `true` para converter o valor do timestamp das colunas 'TIMESTAMP WITH TIME ZONE' e 'TIMESTAMP WITH LOCAL TIME ZONE' em UTC. Por padrão, o valor desse atributo é 'falso' e os dados são replicados utilizando o fuso horário do banco de dados de origem.	
Origem Oracle: `DataTruncationErrorPolicy` para SUSPEND_TABLE não funciona	Corrigido um problema na origem Oracle com destino S3 em que as tabelas não eram suspensas enquanto a configuração da tarefa `DataTruncationErrorPolicy` estava definida como SUSPEND_TABLE.	
Falha no SQL Server em um esquema/tabela longa ao criar a cláusula de consulta	Corrigido um problema na origem SQL Server em que a tarefa falhava ou deixava de responder quando a regra de seleção continha uma lista de tabelas separadas por vírgula.	
Autenticação do Secret Manager com o endpoint do MongoDB	Corrigido um problema nos endpoints do MongoDB e do DocumentDB em que a autenticação baseada no Secret Manager não estava funcionando.	

Tópico	Resolução	
DMS truncando os dados durante a CDC para uma coluna varchar de vários bytes quando NLS_NCHAR _CHARACTERSET está definido como UTF8	Corrigido um problema na origem Oracle com destino Oracle em que os dados estavam sendo truncados para colunas VARCHAR de vários bytes com NLS_NCHAR _CHARACTERSET definido como UTF8.	
`filterTra nsactions 0fUser` ECA para Oracle LogMiner	Foi adicionado um Atributo de Conexão Extra (ECA) `filterTransactions0fUser` para permitir que o DMS ignore transações de um usuário especificado ao replicar do Oracle usando. LogMiner	
Erro recuperável de configuração do SQL Server quando o lsn está ausente do backup	Corrigido um problema no SQL Server em que uma tarefa não falhava com LSN ausente.	

AWS Notas de versão do Database Migration Service 3.4.7

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos na versão 3.4.7 do AWS Database Migration Service (AWS DMS).

Novo recurso ou aprimoramento	Descrição
Compatibilidade com o Babelfish como destino	AWS DMS agora suporta Babelfish como alvo. Usando AWS DMS, agora você pode migrar dados ativos de qualquer fonte AWS DMS suportada para um Babelfish, com o mínimo de tempo de inatividade.

Novo recurso ou aprimoramento	Descrição
	Para ter mais informações, consulte Utilizar o Babelfish como destino do AWS Database Migration Service.
Compatibilidade com bancos de dados IBM Db2 z/OS como origem somente para carga máxima	AWS DMS agora suporta bancos de dados IBM Db2 z/OS como fonte. Usando AWS DMS, agora você pode realizar migrações ao vivo de mainframes Db2 para qualquer AWS DMS destino compatível. Para ter mais informações, consulte Utilizar o bancos de dados IBM Db2 for z/OS como origem do AWS DMS.
Compatibilidade com a réplica de leitura do SQL Server como origem	AWS DMS agora oferece suporte à réplica de leitura do SQL Server como fonte. Usando AWS DMS, agora você pode realizar migrações ao vivo da réplica de leitura do SQL Server para qualquer destino AWS DMS compatível. Para ter mais informações, consulte Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS.
Eventos do Support EventBridge DMS	AWS DMS suporta o gerenciamento de assinaturas de eventos usando EventBridge para eventos do DMS. Para ter mais informações, consulte Como trabalhar com eventos e notificações do Amazon EventBridge no AWS Database Migration Service.

Novo recurso ou aprimoramento	Descrição
Compatibilidade com endpoints de origem e de destino da VPC	AWS DMS agora oferece suporte a endpoints Amazon Virtual Private Cloud (VPC) como fontes e destinos. AWS DMS agora podem se conectar a qualquer AWS serviço com VPC endpoints quando rotas explicitamente definidas para os serviços são definidas em sua VPC. AWS DMS **Note** As atualizações para AWS DMS as versões 3.4.7 e superiores exigem que você primeiro configure para AWS DMS usar VPC endpoints ou usar rotas públicas. Esse requisito se aplica aos endpoints de origem e destino do Amazon S3, Amazon Kinesis AWS Secrets Manager Data Streams, Amazon DynamoDB, Amazon Redshift e Amazon Service. OpenSearch Para ter mais informações, consulte Configurar endpoints da VPC como endpoints de origem e de destino do AWS.
Nova versão do PostgreSQL	O PostgreSQL versão 14.x agora é compatível como origem e como destino.
Compatibilidade com o Aurora Sem Servidor v2 como destino	AWS DMS agora oferece suporte ao Aurora Serverless v2 como destino. Usando AWS DMS, agora você pode realizar migrações ao vivo para o Aurora Serverless v2. Para obter informações sobre AWS DMS alvos compatíveis, consulteDestinos para a migração de dados.

Novo recurso ou aprimoramento	Descrição
Novas versões do IBM Db2 for LUW	AWS DMS agora suporta as versões 11.5.6 e 11.5.7 do IBM Db2 for LUW como fonte. Usando AWS DMS, agora você pode realizar migrações ao vivo das versões mais recentes do IBM DB2 for LUW. Para obter informações sobre AWS DMS fontes, consulteOrigens para a migração de dados. Para obter informações sobre AWS DMS alvos compatíveis, consulteDestinos para a migração de dados.

AWS DMS 3.4.7 inclui o seguinte comportamento novo ou alterado e problemas resolvidos:

- Agora é possível utilizar um formato de data da definição da tabela para analisar uma string de dados em um objeto de data ao utilizar o Amazon S3 como origem.
- Novos contadores de estatísticas de tabela agora estão disponíveis: `AppliedInserts`, `AppliedDdls`, `AppliedDeletes` e `AppliedUpdates.`
- Agora você pode escolher o tipo de mapeamento padrão ao usar OpenSearch como destino.
- A nova configuração do endpoint `TrimSpaceInChar` para origens Oracle, PostgreSQL e SQLServer permite especificar se você deseja cortar dados nos tipos de dados CHAR e NCHAR.
- A nova configuração do endpoint `ExpectedBucketOwner` do Amazon S3 evita o corte de dados ao utilizar o S3 como origem ou destino.
- Para o RDS SQL Server, o Azure SQL Server e o SQL Server autogerenciado, o DMS agora fornece configuração automática do MS-CDC em todas as tabelas selecionadas para uma tarefa de migração com ou sem uma CHAVE PRIMÁRIA ou com um índice exclusivo considerando a prioridade de habilitação de MS-REPLICATION em tabelas autogerenciadas do SQL Server com CHAVE PRIMÁRIA.
- Adicionado suporte para replicação de operações DDL de partição e subpartição do Oracle durante migrações homogêneas do Oracle.
- Corrigido um problema em que uma tarefa de validação de dados falhava com uma chave primária composta ao utilizar o Oracle como origem e como destino.
- Corrigido um problema ao converter corretamente um tipo de caractere variável em um booleano enquanto a coluna de destino era pré-criada como booleana ao utilizar o Redshift como destino.

- Corrigido um problema que causava o truncamento de tipos de dados `varchar` migrados como `varchar(255)` devido a um problema conhecido de ODBC ao utilizar o PostgreSQL como destino.
- Corrigido um problema em que a dica paralela para a operação DELETE não era respeitada com `BatchApplyEnabled` definido como `true` e `BatchApplyPreserveTransaction` definido como `false` ao utilizar o Oracle como destino.
- A nova configuração do endpoint `AddTrailingPaddingCharacter` para um Amazon S3 adiciona preenchimento aos dados da string ao utilizar o S3 como destino.
- A nova configuração da tarefa `max_statement_timeout_seconds` estende o tempo limite padrão das consultas do endpoint. Essa configuração é usada atualmente por consultas de metadados do endpoint do MySQL.
- Ao utilizar o PostgreSQL como destino, foi corrigido um problema em que uma tarefa da CDC não estava utilizando adequadamente as configurações da tarefa de tratamento de erros.
- Corrigido um problema em que o DMS não conseguia identificar corretamente o modo Redis para uma instância do Redis Enterprise.
- Suporte estendido do atributo de conexão extra (ECA) `includeOpForFullLoad` para o formato parquet do destino S3.
- Introduzida uma nova configuração `migrateBooleanAsBoolean` do endpoint do PostgreSQL. Quando essa configuração estiver definida como `true` para uma migração do PostgreSQL para o Redshift, um booleano será migrado como varchar(1). Quando definido como `false`, um booleano será migrado como varchar(15), que é o comportamento padrão.
- Corrigido um problema na migração ao utilizar a origem SQL Server com o tipo de dados `datetime`. Essa correção soluciona o problema de inserção de `Null` quando a precisão é em milissegundos.
- Corrigido um problema na migração da origem PostgreSQL com PGLOGICAL ao utilizar pglogical e remover um campo da tabela de origem durante a fase de CDC, em que, após o campo removido, o valor não era migrado para a tabela de destino.
- Corrigido um problema na migração de loopback do SQL Server com a replicação bidirecional obtendo registros repetidos.
- Adicionada uma nova ECA `mapBooleanAsBoolean` para o PostgreSQL como origem. Usando esse atributo de conexão extra, você pode substituir o mapeamento padrão do tipo de dados de um booleano do PostgresSQL para um tipo de dados booleano. RedShift
- Corrigido um problema na migração ao utilizar o SQL Server como origem que aborda o ALTER DECIMAL/NUMERIC SCALE que não replicava para os destinos.

- Problema de conexão corrigido com o SQL Server 2005.
- Desde 17 de outubro de 2022, o DMS 3.4.7 é compatível com as classes de instância do Amazon EC2 de 6ª geração para instâncias de replicação.
- Desde 25 de novembro de 2022, com o DMS 3.4.7, é possível converter esquemas de banco de dados e objetos de código utilizando o DMS Schema Conversion e descobrir bancos de dados em seu ambiente de rede que são bons candidatos à migração utilizando o DMS Fleet Advisor.
- Em 25 de novembro de 2022, o DMS Studio foi descontinuado.
- Desde 31 de janeiro de 2023, o DMS Schema Conversion é compatível com o Aurora MySQL e com o Aurora PostgreSQL como provedores de dados de destino.
- Desde 6 de março de 2023, é possível gerar recomendações de destino de tamanho certo para os bancos de dados de origem com o DMS Fleet Advisor.
- A partir de 6 de março de 2023, AWS DMS oferece suporte à política AWS gerenciada que permite a publicação de pontos de dados métricos na Amazon CloudWatch.

Problemas resolvidos na versão de manutenção do DMS 3.4.7, datada de 5 de maio de 2023

Tópico	Resolução	
Falha na tarefa de origem PostgreSQL	Corrigido um problema na origem PostgreSQL em que as tarefas falhavam ao exceder o número máximo de operações de DDL permitidas em um único evento.	
Falsos positivos da validação de dados de origem do PostgreSQL	Corrigido um problema na origem do PostgreSQL com destino do Oracle em que a conversão incorreta do campo de timestamp resultava em erros falsos positivos na validação de dados.	
Tratamento de erros de origem MySQL	Corrigido um problema em uma origem MySQL em que a tarefa do DMS não falhava quando o próximo log de BIN estava indisponível.	
Log ROTATE_EVENT da origem MySQL	Log aprimorado para a origem MySQL relacionado a ROTATE_EVENT: incluído o nome do log de BIN que está sendo lido.	

Tópico	Resolução
Problema de tempo limite da validação de dados	Corrigido um problema no recurso de validação de dados em que a configuração do endpoint `executeTimeout` não estava sendo respeitada para consultas relacionadas à validação de dados.
Problema de carga máxima paralela do destino PostgreSQL	Corrigido um problema no destino do PostgreSQL em que a carga máxima segmentada (paralela) falhava devido ao erro de “conexão inativa”.
Problema de movimentação de tarefas do DMS	Corrigido um problema no destino S3 em que uma tarefa de movimentação do DMS demorava muito ou nunca era concluída.
Problema de registro duplicado na origem PostgreSQL	Corrigido um problema na origem PostgreSQL em que uma tarefa do DMS emitia erros relacionados a duplicatas no destino após a parada e a retomada de uma tarefa.
Falsos positivos da validação de dados no destino do Oracle	Corrigido um problema no destino do Oracle em que a validação de dados relatava erros de falsos positivos devido ao fuso horário replicado incorretamente para campos de carimbo de timestamp.

Problemas resolvidos na versão de manutenção do DMS 3.4.7, datada de 22 de fevereiro de 2023

Tópico	Resolução
Réplicas do SQL Server AG como origem	Foi adicionado suporte para a fonte do SQL Server na AlwaysOnconfiguração em que a porta TCP do ouvinte diferia da porta TCP de réplica.
Perda de dados com o Amazon Redshift como destino	Corrigido um problema no destino Redshift em que, em alguns casos raros, a reinicialização inesperada

Tópico	Resolução	
	do Redshift poderia ter utilizado a falta de dados no destino.	
Compatibilidade com a proteção da origem do SQL Server	Corrigido um problema na origem SQL Server em que a tarefa do DMS poderia falhar com um erro indicando a incapacidade de ler os backups do log de transaçõe s quando a configuração do endpoint `"SafeguardPolicy": "EXCLUSIVE_AUTOMATIC_TRUNCATION"` era especificada.	
Falha na tarefa de validação de dados no Oracle como origem	Corrigido um problema na origem Oracle em que a tarefa do DMS poderia falhar na validação de dados devido a valores de chave primária identificados incorretamente.	
Problema de dados de imagem antes do Kinesis	Corrigido um problema nos destinos de streaming (Kinesis, Kafka) em que a configuração da tarefa `"EnableBeforeImage"` funcionava somente para tipos de dados de caracteres.	
Arquivos de log do Time Travel	Corrigido um problema no recurso Time Travel em que o DMS criava arquivos de log do Time Travel de zero bytes quando a origem estava ociosa.	

Problemas resolvidos na versão de manutenção do DMS 3.4.7, datada de 16 de dezembro de 2022

Tópico	Resolução	
BatchApplyAtivado	Corrigido um problema de registro excessivo quando BatchApplyEnabled definido como True.	
Nova configuração de endpoint do MongoDB — Tempo limite FullLoad NoCursor	A FullLoadNoCursorTimeout configuração do endpoint do MongoDB NoCursorTimeout especifica o cursor de carga total. NoCursorTimeout é uma configuração de conexão do MongoDB que impede que o servidor feche o cursor se estiver ocioso.	

Tópico	Resolução	
MongoDB: perfil de filtro para segmentação de coluna única	O novo perfil de filtro melhora o desempenho da migração de bancos de dados MongoDB utilizando uma única coluna para segmentação.	
MongoDB para Redshift	Corrigido um problema na migração do MongoDB para o Redshift em que se a coleção do MongoDB tivesse o tipo de dados binário, o DMS não estava criando a tabela de destino no Redshift.	
Novo atributo de conexão SocketTimeout MongoDB MS	O novo atributo de conexão extra SocketTimeout MongoDB MS configura o tempo limite de conexão para clientes MongoDB em unidades de milissegundos. Se o valor for menor ou igual a zero, o padrão do cliente MongoDB será utilizado.	
Corrigido o problema que fazia com que uma tarefa do Amazon Kinesis falhasse	Corrigido um problema no tratamento de valores nulos ao migrar para o Amazon Kinesis Data Streams como destino se uma chave primária não estivesse presente na tabela.	
Compatibilidade com a validação de dados do Oracle NULL PK/UK	Removida a limitação de que a validação de dados de valores NULL PK/UK não é compatível.	
Oracle para Amazon S3	Corrigido um problema ao migrar do Oracle para o Amazon S3 em que alguns registros eram migrados incorretamente como NULL.	
Oracle Standby	Adicionada a capacidade do DMS de tratar transações abertas ao utilizar o Oracle Standby como origem.	

Tópico	Resolução
Migração de Oracle para Oracle com o tipo de dados espaciais SDO_GEOMETRY	Corrigido um problema em que a tarefa falhava se a tabela tivesse uma coluna SDO_GEOMETRY presente no DDL ao migrar do Oracle para o Oracle.
Oracle como origem	Ao utilizar o Oracle como origem, foi corrigido um problema em que o DMS ocasionalmente ignorava um número de sequência do redo log do Oracle.
Oracle como origem: arquivamento ausente/online de redo logs	Corrigido um problema que fazia com que a tarefa do DMS falhasse quando os logs de arquivamento estavam ausentes ao utilizar o Oracle como origem.
Corrigido: o DMS ocasionalmente ignora o redo log do Oracle Standby	Ao utilizar o Oracle como origem, foi corrigido um problema em que o DMS ocasionalmente ignorava um número de sequência do redo log do Oracle.
Corrigido: os tipos de dados espaciais do Oracle para Oracle não são replicados durante a CDC	Ao replicar do Oracle para o Oracle, foi corrigido um problema em que os tipos de dados espaciais não estavam sendo replicados durante a CDC.
Oracle como destino	Ao utilizar o Oracle como destino, foi corrigido um problema em que a aplicação do destino falhava com um erro ORA-01747.
Amazon S3: perda de dados da tabela de recarga corrigido	Ao utilizar o Amazon S3 como destino, foi corrigido um problema em que uma operação de recarregamento de tabela não estava gerando arquivos de CDC.

Tópico	Resolução
Corrigido: inicializ ação de contexto SQL Server Always On, caso o servidor primário fosse a origem	Ao usar o SQL Server Always On como fonte, corrigiu um problema para não inicializar Grupos de Disponibi lidade (AG) se a fonte for primária e AlwaysOnS haredSyncedBackupIsEnabled estiver definida como verdadeira.
Configuração do endpoint do SQL Server atualizada	Quando um endpoint de origem é o SQL Server Always On Availability Group e é uma réplica secundária, foi corrigido um problema em que a tarefa de replicação falhava se estivesse AlwaysOnSharedSynchedBackup sIsEnabled definida como True.
PostgreSQL como origem	Corrigido um problema em que o CDC não consegue migrar as operações de exclusão/atualização na fonte do PostgreSQL, que foi introduzida na versão 3.4.7 no suporte ao Boolean. mapBooleanAs

AWS Notas de versão do Database Migration Service 3.4.6

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos na versão 3.4.6 do AWS Database Migration Service (AWS DMS).

Novo recurso ou aprimoramento	Descrição
AWS DMS Viagem no tempo	AWS DMS apresenta o Time Travel, um recurso que concede aos clientes flexibilidade em seus recursos de registro e aprimora sua experiência de solução de problemas. Com o Time Travel, você pode armazenar e criptografar AWS DMS registros usando o Amazon S3 e visualizar, baixar e ofuscar os registros dentro de um determinado período de tempo.
Compatibilidade com a instância gerenciada do	AWS DMS agora oferece suporte à Instância Gerenciada SQL do Microsoft Azure como fonte. Usando AWS DMS, agora você

Novo recurso ou aprimoramento	Descrição
Microsoft Azure SQL como origem	pode realizar migrações ao vivo da Instância Gerenciada SQL do Microsoft Azure para qualquer destino AWS DMS compatível. Para obter informações sobre AWS DMS fontes, consulteOrigens para a migração de dados. Para obter informações sobre AWS DMS alvos compatíveis, consulteDestinos para a migração de dados.
Compatibilidade com o Google Cloud SQL para MySQL como origem	AWS DMS agora é compatível com o Google Cloud SQL para MySQL como fonte. Usando AWS DMS, agora você pode realizar migrações ao vivo do Google Cloud SQL para MySQL para AWS DMS qualquer destino compatível. Para obter informações sobre AWS DMS fontes, consulteOrigens para a migração de dados. Para obter informações sobre AWS DMS alvos compatíveis, consulteDestinos para a migração de dados.
Compatibilidade com a carga paralela para dados particionados no S3	AWS DMS agora oferece suporte ao carregamento paralelo de dados particionados para o Amazon S3, melhorando os tempos de carregamento para migrar dados particionados dos dados de origem do mecanismo de banco de dados compatível para o Amazon S3. Esse recurso cria subpastas do Amazon S3 para cada partição da tabela na origem do banco de dados, permitindo que o AWS DMS execute processos paralelos para preencher cada subpasta.

Novo recurso ou aprimoramento	Descrição
Suporte a vários tópicos de destino do Apache Kafka em uma única tarefa	AWS DMS agora suporta alvos multitópicos do Apache Kafka com uma única tarefa. Utilizando o AWS DMS, agora é possível replicar vários esquemas de um único banco de dados para diferentes tópicos de destino do Apache Kafka utilizando a mesma tarefa. Isso elimina a necessidade de criar várias tarefas separadas em situações em que muitas tabelas do mesmo banco de dados de origem precisam ser migradas para diferentes tópicos de destino do Kafka.

Os problemas resolvidos na AWS DMS versão 3.4.6 incluem o seguinte:

- Corrigido um problema em que as colunas de instruções UPDATE eram preenchidas em colunas incorretas se a coluna de chave primária não fosse a primeira coluna ao utilizar o Amazon S3 como destino com o formato CSV.
- Corrigido um problema em que AWS DMS as tarefas podiam falhar ao usar o plug-in pglogical com `NULL` valores em `BYTEA` colunas no modo LOB limitado ao usar o PostgreSQL como fonte.
- Corrigido um problema em que AWS DMS as tarefas podiam falhar quando um grande número de tabelas de origem era excluído ao usar o PostgreSQL como fonte.
- O particionamento de pastas com base em datas do Amazon S3 foi aprimorado com a introdução de uma nova configuração `DatePartitionTimezone` do Amazon S3 para permitir o particionamento em datas não UTC.
- Compatibilidade com o mapeamento entre os tipos de dados `TIMESTAMP WITH TIME ZONE` das origens para `TIMESTAMPTZ` ao utilizar o Redshift como destino
- Desempenho da CDC melhorado para tarefas sem regras de seleção de caracteres curinga ao utilizar o MongoDB ou o Amazon DocumentDB como origem.
- Corrigido um problema em que nomes de esquema com caractere curinga de sublinhado e tamanho menor que 8 não eram capturados por tarefas do AWS DMS ao utilizar o Db2 LUW como origem.
- Corrigido um problema em que AWS DMS as instâncias ficavam sem memória em um grande volume de dados ao usar o OpenSearch Serviço como destino.

- Desempenho melhorado da validação de dados tornando-se compatível com somente as tarefas de validação de carga máxima.
- Corrigido um problema em que AWS DMS as tarefas não eram retomadas após um failover forçado ao usar o Sybase como fonte.
- Corrigido um problema em que o aviso era AWS DMS enviado `Invalid BC timestamp was encountered in column` incorretamente.

Os problemas resolvidos no DMS 3.4.6 incluem os seguintes:

- Corrigido o problema de falha de uma tarefa quando o modo de aplicação em massa estava ativado ao utilizar o Oracle como origem e destino.
- Corrigido um problema de forma que uma tarefa de carga máxima utilize corretamente a configuração do endpoint `ExecuteTimeout` com o PostgreSQL como origem.
- Corrigido um problema com a migração de colunas do tipo de dados Array quando a tarefa é definida no modo LOB limitado ao utilizar o PostgreSQL como origem.
- Corrigido um problema com a migração de timestamps com fuso horário antes de 01-01-1970 ao utilizar o PostgreSQL como origem.
- Corrigido um problema em que o DMS estava tratando uma string vazia como nula durante a replicação ao utilizar o SQL Server como origem e como destino.
- Corrigido um problema para honrar as configurações do endpoint de tempo limite de leitura e gravação da sessão ao utilizar a origem/destino MySQL.
- Corrigido um problema em que uma tarefa do DMS CDC estava baixando arquivos relacionados à carga máxima ao utilizar o Amazon S3 como origem.
- Corrigido um problema de falha no log quando `CdcInsertsAndUpdates` e `PreserveTransactions` estavam configurados como `true` ao utilizar o Amazon S3 como destino.
- Corrigido um problema em que uma tarefa travava quando o recurso ParallelApply * estava ativado, mas algumas tabelas não tinham uma chave primária padrão ao usar o Amazon Kinesis Data Streams como fonte.
- Corrigido um problema em que não era dado um erro devido a um erro StreamArn ao usar o Amazon Kinesis Data Streams como fonte.
- Corrigido um problema em que um valor de chave primária como uma string vazia fazia com que uma tarefa falhasse ao OpenSearch ser usada como destino.
- Corrigido um problema em que muito espaço em disco era utilizado pela validação de dados.

Problemas resolvidos na versão de manutenção do DMS 3.4.6, datada de 13 de dezembro de 2022

Tópico	Resolução
Driver odbc do SAP ASE	Corrigido um problema no SAP ASE como origem para que o driver ODBC seja compatível com conjuntos de caracteres.
Erro de chave primária de data e hora do SQL Server para pesquisa de LOB	Corrigido um problema no SQL Server como origem em que a pesquisa de LOB não estava funcionando corretamente, quando a chave primária tinha um tipo de dados de data e hora, com precisão em milissegundos.
SQL Server para Redshift: “datetime offset' mapeado para “timestamptz”	Para migrações do SQL Server para o Redshift, o mapeamento foi aprimorado para que o formato “datetimeoffset” do SQL Server seja mapeado para o formato “timestamptz” do Redshift.
Validação de dados - SkipLobColumns é verdade	Corrigido um problema em que a tarefa do DMS trava quando SkipLobColumns é True, há um LOB na origem, a chave primária está na última coluna e uma diferença de dados é detectada pela validação.
Validação de dados com o MySQL como origem	Corrigido um problema no MySQL como origem com a validação de dados ativada, em que ocorria uma falha na tarefa do DMS ao utilizar uma tabela que tinha uma chave exclusiva composta com valores nulos.
MySQL como origem	Corrigido um problema no MySQL como origem, em que uma tabela era suspensa com o erro de estouro quando as colunas eram alteradas para adicionar precisão.
Atualização do driver ODBC do MySQL para 8.0.23	Corrigido um problema no MySQL como origem, em que o agrupamento “utf8mb4_0900_bin” era incompatí vel com o driver do MySQL utilizado pelo DMS.
MySQL: suporte às alterações de DDL	Introduziu uma nova configuração de endpoint MySQL skipTableSuspension ForPartitionDdl para permitir que

Tópico	Resolução	
para tabelas particion adas	o usuário pule a suspensão da tabela para alterações de DDL de partição durante o CDC, para que o DMS agora possa suportar alterações de DDL em tabelas MySQL particionadas.	
Migração do MongoDB para o Redshift	Corrigido um problema em migrações do MongoDB para o Redshift, em que o DMS não criava a tabela de destino no Redshift se a coleção do MongoDB tivesse o tipo de dados binário.	
Destino Redshift: Segfault do Time Travel em aplicação em massa	Corrigido um problema no Redshift como destino, em que a tarefa do DMS falhava quando definida como verdadeira. BatchApplyEnabled	
Redshift como destino	Corrigido um problema no Redshift como destino, em que, com a carga paralela definida como type=part itions-auto, os segmentos paralelos estavam gravando arquivos CSV em massa no mesmo diretório da tabela e interferindo um com o outro.	
Redshift como destino	Corrigido um problema no Redshift como destino, em que, durante a CDC, a coluna de destino era do tipo booleano, enquanto a origem era do tipo caractere variável.	
Redshift como destino	O log de tarefas foi aprimorado para identificar uma alteração de DDL que não consegue ser replicada para o Redshift como destino.	
Validação de dados com o PostgreSQL	Corrigido um problema na validação com o PostgreSQ L, em que a validação falhava quando tipos de dados booleanos estavam presentes.	
PostgreSQL como origem	Foi corrigido um problema com o PostgreSQL como fonte, para que a carga total usasse o campo em ExecuteTimeout Atributos de conexão extra.	

Tópico	Resolução	
PostgreSQL como origem	Corrigido um problema no PostgreSQL como origem, de forma que uma tarefa falhe ao ler LSNs maiores do que a tarefa solicitada, retome o LSN por mais de 60 minutos para indicar que há um problema com o slot de replicação que está sendo utilizado.	
PostgreSQL como origem: timestamptz antes de 01-01-1970	Corrigido um problema no PostgreSQL como origem, em que os timestamptz anteriores a 01-01-1970 não eram migrados corretamente durante o CDC.	
PostgreSQL como origem	Corrigido um problema no PostgreSQL como origem, em que o DMS truncava os valores de tipos de dados de vários caracteres durante a CDC.	
PostgreSQL como origem: retomada de tarefa interrompida	Corrigido um problema no PostgreSQL como origem em que a retomada de repetição de uma tarefa interrompida anteriormente perdia uma ou mais transações durante a CDC.	
Amazon S3 como destino	Corrigido um problema no S3 como destino, em que o cabeçalho do arquivo CSV resultante estava desligado em uma coluna quando AddColumnName era verdadeiro e TimestampColumnName era “”.	
Amazon S3 como origem: comportamento de utilização da memória na fase de carga máxima da tarefa	Corrigido um problema no S3 como origem, em que uma tarefa do DMS em carga máxima só liberava a memória utilizada depois que a tabela inteira era carregada no banco de dados de destino.	
Amazon S3 como destino: operação de recarga da tabela	Corrigido um problema no S3 como destino, em que uma operação de recarregamento de tabela não gerava arquivos de CDC.	

AWS Notas de versão do Database Migration Service 3.4.5

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos na versão 3.4.5 do AWS Database Migration Service (AWS DMS).

Novo recurso ou aprimoramento	Descrição
Compatibilidade com o Redis como destino	AWS DMS agora oferece suporte ao Redis como alvo. Usando AWS DMS, agora você pode migrar dados ativos de qualquer fonte AWS DMS compatível para um armazenamento de dados do Redis, com o mínimo de tempo de inatividade. Para obter informaçõ es sobre AWS DMS alvos, consulteDestinos para a migração de dados.
Compatibilidade com os MongoDB 4.2 e 4.4 como origens	AWS DMS agora suporta MongoDB 4.2 e 4.4 como fontes. Usando AWS DMS, agora você pode migrar dados dos clusters MongoDB 4.2 e 4.4 para AWS DMS qualquer destino compatível, incluindo o Amazon DocumentDB (com compatibilidade com o MongoDB), com tempo de inatividade mínimo. Para obter informações sobre AWS DMS fontes, consulteOrigens para a migração de dados.
Suporte a vários bancos de dados ao utilizar o MongoDB como origem	AWS DMS agora suporta a migração de vários bancos de dados em uma tarefa usando o MongoDB como fonte. Usando AWS DMS, agora você pode agrupar vários bancos de dados de um cluster MongoDB e migrá-los usando uma tarefa de migração de banco de dados. Você pode migrar para qualquer destino AWS DMS compatível, incluindo o Amazon DocumentDB (com compatibilidade com o MongoDB), com o mínimo de tempo de inatividade.
Compatibilidade com a segmentação automátic a ao utilizar o MongoDB ou o Amazon DocumentD B (compatível com MongoDB) como origem	AWS DMS agora oferece suporte à segmentação automática usando o MongoDB ou o Amazon DocumentDB como fonte. Usando AWS DMS, você pode configurar tarefas de migração de banco de dados para segmentar automaticamente a coleção de um cluster MongoDB ou DocumentDB. Em seguida, você pode migrar os segmentos paralelamente para qualquer destino AWS DMS

Novo recurso ou aprimoramento	Descrição
	compatível, incluindo o Amazon DocumentDB, com o mínimo de tempo de inatividade.
Melhoria no desempenho de carga máxima do Amazon Redshift	AWS DMS agora suporta o uso de threads paralelos ao usar o Amazon Redshift como destino durante a carga total. Ao aproveitar as configurações de tarefas multisegmentadas de carga total, você pode melhorar o desempenho da sua migração inicial de qualquer fonte AWS DMS compatível para o Amazon Redshift. Para obter informações sobre AWS DMS alvos, consulteDestinos para a migração de dados.

Os problemas resolvidos na AWS DMS versão 3.4.5 incluem o seguinte:

- Corrigido um problema em que os dados podiam estar ausentes ou duplicados após a retomada ao utilizar o PostgreSQL como origem com alta simultaneidade de transações.
- Corrigido um problema em que as tarefas de migração do banco de dados falhavam com o erro Não foi possível encontrar o id da relação... ao utilizar o PostgreSQL como origem, com o plug-in pglogical ativado.
- Corrigido um problema em que as colunas `VARCHAR` não eram replicadas corretamente ao utilizar o PostgreSQL como origem e o Oracle como destino.
- Corrigido um problema em que as operações de exclusão não eram capturadas corretamente quando a chave primária não era a primeira coluna na definição da tabela ao utilizar o PostgreSQL como origem.
- Corrigido um problema em que as tarefas de migração do banco de dados perdiam as atualizações de LOB em uma configuração especial de metadados ao utilizar o MySQL como origem.
- Corrigido um problema em que as colunas `TIMESTAMP` eram tratadas como `DATETIME` no modo LOB completo ao utilizar o MySQL versão 8 como origem.
- Corrigido um problema em que as tarefas de migração do banco de dados falhavam ao analisar registros de `NULL DATETIME` ao utilizar o MySQL 5.6.4 e superior como origem.

- Corrigido um problema em que as tarefas de migração do banco de dados ficavam paralisadas após encontrar um erro de Saída de Thread ao utilizar o Amazon Redshift como destino com aplicação paralela.
- Corrigido um problema em que os dados poderiam ser perdidos quando as tarefas de migração do banco de dados se desconectavam com um endpoint de destino do Amazon Redshift durante a aplicação em lote da CDC.
- Desempenho da carga máxima melhorado por meio de chamadas de `ACCEPTINVCHARS` ao utilizar o Amazon Redshift como destino.
- Corrigido um problema em que registros duplicados eram replicados ao reverter do modo one-by-one para o modo de aplicação paralela usando o Amazon Redshift como destino.
- Corrigido um problema em que as tarefas de migração do banco de dados não trocavam a propriedade do objeto do Amazon S3 para o proprietário do bucket com `cannedAclForObjects=bucket_owner_full_control` ao utilizar o Amazon S3 como destino.
- AWS DMS Aprimorado ao oferecer suporte a vários destinos de arquivamento com o ECA `additionalArchivedLogDestId` ao usar o Oracle como fonte.
- Corrigido um problema em que as tarefas de migração do banco de dados falhavam com erro `OCI_INVALID_HANDLE` ao atualizar uma coluna de LOB no modo LOB completo.
- Corrigido um problema em que as colunas `NVARCHAR2` não eram migradas adequadamente durante a CDC ao utilizar o Oracle como origem.
- AWS DMS Aprimorado com a habilitação `SafeguardPolicy` ao usar o RDS para SQL Server como fonte.
- Corrigido um problema em que as tarefas de migração do banco de dados relatavam o erro `rdsadmin` ao utilizar uma origem SQL Server que não era do RDS.
- Corrigido um problema em que a validação de dados falhava com o UUID como chave primária em uma configuração de partição ao utilizar o SQL Server como origem.
- Corrigido um problema em que as tarefas de carga máxima mais CDC podiam falhar se o LSN necessário não pudesse ser encontrado no log do banco de dados ao utilizar o Db2 LUW como origem.
- AWS DMS Aprimorado ao oferecer suporte a carimbos de data/hora personalizados do CDC ao usar o MongoDB como fonte.
- Corrigido um problema em que as tarefas de migração do banco de dados travavam ao serem interrompidas, ao utilizar o MongoDB como origem, quando o driver do MongoDB era ativado por erros de `endSessions`.

- Corrigido um problema em que AWS DMS não era possível atualizar campos não primários ao usar o DynamoDB como destino
- Corrigido um problema em que a validação de dados relatava incompatibilidades de falsos positivos nas colunas `CLOB` e `NCLOB`.
- Corrigido um problema em que a validação de dados falhava em registros de somente espaço em branco ao utilizar o Oracle como origem.
- Corrigido um problema em que as tarefas de migração do banco de dados falhavam ao truncar uma tabela particionada.
- Corrigido um problema em que as tarefas de migração do banco de dados falhavam ao criar a tabela de controle `awsdms_apply_exceptions`.
- Suporte estendido do plug-in de autenticação da `caching_sha2_password` ao utilizar o MySQL versão 8.

AWS Notas de versão do Database Migration Service 3.4.4

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos no AWS DMS versão 3.4.4.

Novo recurso ou aprimoramento	Descrição
Compatibilidade com a criptografia TLS e à autenticação TLS ou SASL ao utilizar o Kafka como destino	AWS DMS agora oferece suporte à criptografia TLS e à autentica ção TLS ou SASL usando o Amazon MSK e o cluster Kafka local como destino. Para obter mais informações sobre como utilizar a criptografia e a autenticação para endpoints do Kafka, consulte Conectar-se ao Kafka utilizando Transport Layer Security (TLS).

Os problemas resolvidos na AWS DMS versão 3.4.4 incluem o seguinte:

- AWS DMS Registro aprimorado de falhas de tarefas ao usar endpoints Oracle.
- AWS DMS A execução aprimorada de tarefas continua sendo processada quando os endpoints de origem da Oracle trocam de função após o failover do Oracle Data Guard.
- O tratamento de erros aprimorado trata o ORA—12561 como um erro recuperável ao utilizar endpoints do Oracle.

- Corrigido um problema em que as colunas `EMPTY_BLOB()` e `EMPTY_CLOB()` eram migradas como nulas ao utilizar o Oracle como origem.
- Corrigido um problema em que AWS DMS as tarefas não atualizavam os registros após adicionar alterações no DDL da coluna ao usar o SQL Server como fonte.
- PostgreSQL aprimorado como origem da migração tornando-se compatível com o tipo de dados `TIMESTAMP WITH TIME ZONE`.
- Corrigido um problema em que a configuração `afterConnectScript` não funcionava durante uma carga máxima ao utilizar o PostgreSQL como destino.
- Introduzida uma nova configuração `mapUnboundedNumericAsString` para tratar melhor o tipo de data `NUMERIC` sem precisão e escala ao utilizar endpoints do PostgreSQL.
- Corrigido um problema em que AWS DMS as tarefas falhavam com “0 linhas afetadas” após interromper e retomar a tarefa ao usar o PostgreSQL como fonte.
- Corrigido um problema em que AWS DMS não era possível migrar o tipo de `TIMESTAMP` dados com o `BC` sufixo ao usar o PostgreSQL como fonte.
- Corrigido um problema em que AWS DMS não era possível migrar o `TIMESTAMP` valor “±infinity” ao usar o PostgreSQL como fonte.
- Corrigido um problema em que strings vazias eram tratadas como `NULL` ao utilizar o S3 como origem com a configuração `csvNullValue` definida como outros valores.
- O atributo de conexão extra `timestampColumnName` foi aprimorado em uma carga máxima com a CDC para ser classificável durante a CDC ao utilizar o S3 como destino.
- Aprimorado o tratamento de tipos de dados binário em formato hexadecimal, como `BYTE`, `BINARY` e `BLOB` ao utilizar o S3 como origem.
- Corrigido um problema em que os registros excluídos eram migrados com caracteres especiais ao utilizar o S3 como destino.
- Corrigido um problema de tratamento de valores de chaves vazios ao utilizar o Amazon DocumentDB (compatível com MongoDB) como destino.
- Corrigido um problema em que AWS DMS falhas na replicação `NumberDecimal` ou `Decimal128` nas colunas ao usar o MongoDB ou o Amazon DocumentDB (com compatibilidade com o MongoDB) como fonte.
- Corrigido um problema que permitia que as tarefas da CDC fossem repetidas quando houvesse um failover no MongoDB ou no Amazon DocumentDB (compatível com MongoDB) como origem.
- Foi adicionada uma opção para remover o prefixo hexadecimal “0x” `RAW` dos valores do tipo de dados ao usar Kinesis, Kafka ou como destino. OpenSearch

- Corrigido um problema em que a validação falhava em colunas de caracteres de tamanho fixo ao utilizar o Db2 LUW como origem.
- Corrigido um problema em que a validação falhava quando somente o tipo de dados de origem ou o tipo de dados de destino era `FLOAT` ou `DOUBLE`.
- Corrigido um problema em que a validação falhava em caracteres `NULL` ao utilizar o Oracle como origem.
- Corrigido um problema em que a validação falhava em colunas XML ao utilizar o Oracle como origem.
- Corrigido um problema em que AWS DMS as tarefas falhavam quando havia colunas anuláveis em chaves compostas usando o MySQL como fonte.
- Corrigido um problema em que AWS DMS não era possível validar as `UNIQUEIDENTIFIER` colunas dos endpoints de origem do SQL Server e as colunas UUID dos endpoints de destino do PostgreSQL.
- Corrigido um problema em que uma tarefa da CDC não utiliza uma definição de tabela de origem atualizada após ela ser modificada.
- AWS DMS Failover aprimorado para tratar falhas de tarefas causadas por um nome de usuário ou senha inválidos como erros recuperáveis.
- Corrigido um problema em que AWS DMS as tarefas falhavam devido à falta de LSNs ao usar o RDS para SQL Server como fonte.

AWS Notas de versão do Database Migration Service 3.4.3

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos no AWS DMS versão 3.4.3.

Novo recurso ou aprimoramento	Descrição
Nova versão do Amazon DocumentDB	O Amazon DocumentDB versão 4.0 agora é compatível como origem.
Nova versão do MariaDB	O MariaDB versão 10.4 agora é compatível como origem e como destino.

Novo recurso ou aprimoramento	Descrição
Support para AWS Secrets Manager integração	É possível armazenar os detalhes da conexão do banco de dados (credenciais do usuário) dos endpoints compatíveis com segurança no AWS Secrets Manager. Em seguida, você pode enviar o segredo correspondente em vez de credenciais de texto sem formatação para criar ou AWS DMS modificar um endpoint. AWS DMS em seguida, se conecta aos bancos de dados do endpoint usando o segredo. Para obter mais informações sobre a criação de segredos para AWS DMS endpoints, consulteUtilizar segredos para acessar endpoints do AWS Database Migration Service.
Opções maiores para instâncias de replicação C5 e R5	Agora é possível criar os seguintes tamanhos maiores de instância de replicação: tamanhos C5 de até 96 vCPUs e 192 GiB de memória e tamanhos R5 de até 96 vCPUs e 768 GiB de memória.
Melhoria no desempenho do Amazon Redshift	AWS DMS agora oferece suporte à aplicação paralela ao usar o Redshift como alvo para melhorar o desempenho da replicação contínua. Para ter mais informações, consulte Configurações de tarefas de vários threads para o Amazon Redshift.

Os problemas resolvidos na AWS DMS versão 3.4.3 incluem o seguinte:

- Corrigido um problema em que o timestamp de confirmação se tornava “01-01-1970 00:00:00” para eventos adiados ao utilizar o Db2 LUW como origem.
- Corrigido um problema em que AWS DMS as tarefas falhavam com uma `NVARCHAR` coluna como chave primária ao usar o SQL Server como fonte com o modo LOB completo.
- Corrigido um problema de registros ausentes durante a fase de alterações em cache ao utilizar o SQL Server como origem.
- Corrigido um problema em que os registros eram ignorados após a retomada AWS DMS das tarefas ao usar o RDS para SQL Server como fonte.
- Corrigido um problema em que o componente AWS DMS de registro ASSERTION gera registros grandes para o SQL Server.
- Corrigido um problema em que a validação de dados falhava durante a fase de CDC devido ao estouro da análise de colunas ao utilizar o MySQL como origem.

- Corrigido um problema em que AWS DMS as tarefas travavam devido a uma falha de segmentação durante a validação de dados ao usar o PostgreSQL como destino.
- Corrigido um problema em que a validação de dados falhava no tipo de dados DOUBLE durante a CDC ao utilizar o PostgreSQL como origem e como destino.
- Corrigido um problema em que os registros inseridos pelo comando copy não eram replicados corretamente ao utilizar o PostgreSQL como origem e o Redshift como destino.
- Corrigido um problema de perda de dados durante a fase de alterações armazenadas em cache ao utilizar o PostgreSQL como origem.
- Corrigido um problema que poderia causar a perda de dados ou duplicatas de registros ao utilizar o PostgreSQL como origem.
- Corrigido um problema em que esquemas com letras maiúsculas e minúsculas não migravam com pglogical ao utilizar o PostgreSQL como origem.
- Corrigido um problema em que a última mensagem de falha não continha o erro ORA ao utilizar o Oracle como origem.
- Corrigido um problema em que AWS DMS as tarefas falhavam ao criar instruções UPDATE ao usar o Oracle como destino.
- Corrigido um problema em que AWS DMS as tarefas não replicavam dados ao usar o Oracle 12.2 como fonte com a configuração ASM e Pluggable Database.
- Análise de registros foi aprimorada preservando as aspas para ser compatível com a RFC 4180 ao utilizar o S3 como origem.
- O tratamento de `timestampColumnName` foi aprimorado para que a coluna de carga máxima seja classificável com a da CDC.
- Ao introduzir uma nova configuração de endpoint`MessageMaxBytes`, corrigiu um problema em que AWS DMS as tarefas falhavam quando havia elementos LOB maiores que 1 MB.
- Corrigido um problema em que AWS DMS as tarefas travavam devido a uma falha de segmentação ao usar o Redshift como alvo.
- Aprimorado o registro de erros em log para a conexão de teste do Redshift.
- Corrigido um problema em AWS DMS que não transferia todos os documentos do MongoDB para o DocumentDB durante o carregamento total.
- Corrigido um problema em que AWS DMS as tarefas relatavam um erro fatal quando nenhuma tabela era incluída nas regras de mapeamento de tabelas.
- Corrigido um problema em que os esquemas e as tabelas criados antes de reiniciar as tarefas do AWS DMS não eram replicados para o destino ao utilizar o MySQL como origem.

- Corrigido um problema em que o escape do caractere curinga [_] não funcionava na regra de exclusão ao utilizar o MySQL como origem.
- Corrigido um problema em que a coluna do tipo de dados UNSIGNED BIGINT não era replicada corretamente ao utilizar o MySQL como origem.

AWS Notas de versão do Database Migration Service 3.4.2

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos no AWS DMS versão 3.4.2.

Novo recurso ou aprimoramento	Descrição
Support para conectar de forma privada sua Amazon Virtual Private Cloud (Amazon VPC) AWS ao Database Migration Service (DMS) sem exigir um gateway de internet, dispositivo NAT, conexão VPN ou conexão. AWS Direct Connect	Agora você pode se conectar e acessar a AWS DMS partir da sua Amazon VPC por meio de um endpoint de interface VPC criado por você. Esse endpoint de interface permite que você isole toda a atividade de rede da sua instância de AWS DMS replicação na infraestrutura de rede da Amazon. Ao incluir uma referência a esse endpoint de interface em todas as chamadas de API para AWS DMS usar o AWS CLI ou um SDK, você garante que todas as AWS DMS atividades permaneçam invisíveis para a Internet pública. Para ter mais informações, consulte Segurança da infraestrutura no AWS Database Migration Service. **Note** Esse recurso está disponível usando todas as versões de AWS DMS mecanismo compatíveis.
Particionamento de pastas baseado em data da CDC ao utilizar o Amazon S3 como destino	AWS DMS agora oferece suporte ao particionamento de pastas baseado em data ao replicar dados usando o S3 como destino. Para ter mais informações, consulte Utilizar o particionamento de pastas com base em data.

Os problemas resolvidos na AWS DMS versão 3.4.2 incluem o seguinte:

- Adicionada uma opção `STATUPDATE` ao executar uma migração ao utilizar o Redshift como destino.
- Tarefas de validação aprimoradas com a introdução de uma nova configuração. `ValidQueryCdcDelaySecond` atrasa a primeira consulta de validação nos endpoints de origem e de destino para ajudar a reduzir a contenção de recursos quando a latência da migração é alta.
- Corrigido um problema em que AWS DMS demorava muito para iniciar as tarefas de validação.
- Corrigido um problema em que registros vazios eram gerados ao iniciar ou interromper tarefas de replicação ao utilizar o S3 como destino.
- Corrigido um problema em que as tarefas ficavam paralisadas após a conclusão de uma carga máxima.
- Corrigido um problema em que as tarefas ficavam paralisadas quando uma tabela de origem apresentava erros de dados ao utilizar o S3 como origem.
- Corrigido um problema em que as tarefas ficavam paralisadas ao serem iniciadas quando a conta de usuário do endpoint de origem estava desativada.
- Corrigido um problema em que as tarefas ficavam paralisadas ao utilizar o PostgreSQL como origem com `REPLICA IDENTITY FULL`.
- Corrigido um problema em que as tarefas perdiam transações ao utilizar o PostgreSQL como origem com o plug-in pglogical.
- Corrigido um problema em AWS DMS que não excluía arquivos de origem compactados ao usar o Redshift como destino.
- Corrigido um problema em que as tarefas de validação relatavam falsos negativos ao utilizar o MySQL como origem e como destino com o tipo de dados `BIGINT UNSIGNED`.
- Corrigido um problema em que as tarefas de validação relatavam falsos positivos ao utilizar o SQL Server como origem com uma coluna de chave primária como tipo CHAR.
- Corrigido um problema em AWS DMS que não limpa objetos de destino ao usar `start-replication` para iniciar tarefas de replicação usando o S3 como destino.
- Foram corrigidos vários problemas na validação de dados ao utilizar o Db2 como origem.
- Corrigido um problema em que as tarefas de validação ficavam paralisadas ao utilizar o SQL Server como origem com a coluna VARCHAR como chave primária.
- Adicionado suporte para o tipo de dados TIMESTAMP WITH TIMEZONE ao utilizar o PostgreSQL como origem

AWS Notas da versão beta do Database Migration Service 3.4.1

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos no AWS DMS versão 3.4.1 Beta.

Novo recurso ou aprimoramento	Descrição
Nova versão do MongoDB	O MongoDB versão 4.0 agora é compatível como origem.
Compatibilidade com o TLS 1.2 para SQL Server	AWS DMS agora oferece suporte ao TLS 1.2 para endpoints do SQL Server.

Os problemas resolvidos na versão AWS DMS 3.4.1 Beta incluem o seguinte:

- Suporte aprimorado ao Oracle 19c TDE.
- Suporte aprimorado ao conjunto de caracteres utf8mb4 e ao tipo de dados de identidade ao utilizar o Redshift como destino.
- Replicação aprimorada da falha da tarefa ao utilizar o MySQL como origem e sem o log binário.
- Compatibilidade com a validação de dados aprimorada em vários tipos de dados e conjuntos de caracteres.
- O tratamento do valor nulo foi aprimorado com a nova configuração `IncludeNullAndEmpty` de endpoint ao utilizar o Kinesis e o Kafka como destino.
- O registro em log e o tratamento de erros foram aprimorados ao utilizar o Kafka como destino.
- O deslocamento do horário de verão foi aprimorado ao utilizar o SQL Server como origem.
- Corrigido um problema em que as tarefas de replicação tentavam criar tabelas existentes para o Oracle como destino.
- Corrigido um problema em que as tarefas de replicação ficavam paralisadas depois que a conexão com o banco de dados era encerrada ao utilizar o Oracle como origem.
- Corrigido um problema em que as tarefas de replicação falhavam na detecção e se reconectavam ao novo primário ao utilizar o SQL Server como origem com a configuração AlwaysOn.
- Corrigido um problema em que as tarefas de replicação não adicionavam uma coluna `"OP"` para a coluna `"D"` sob certas condições do S3 como destino.

AWS Notas da versão beta do Database Migration Service 3.4.0

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos no AWS DMS versão 3.4.0.

Novo recurso ou aprimoramento	Descrição
Nova versão do MySQL	AWS DMS agora oferece suporte ao MySQL versão 8.0 como fonte, exceto quando a carga útil da transação é compactada.
Compatibilidade com o TLS 1.2 para MySQL	AWS DMS agora suporta TLS 1.2 para endpoints MySQL.
Nova versão do MariaDB	AWS DMS agora oferece suporte ao MariaDB versão 10.3.13 como fonte.
Sem SysAdmin acesso a fontes autogerenciadas do Microsoft SQL Server	AWS DMS agora oferece suporte ao acesso de não SysAdmin usuários a endpoints de origem do SQL Server no local e hospedados no EC2. **Note** No momento, esse recurso está no modo Beta. Se você quiser experimentar, entre em contato com o AWS suporte para obter mais informações.
Tarefas da CDC e tabelas de origem do Oracle criadas utilizando `CREATE TABLE AS`	AWS DMS agora suporta tarefas de carga completa e somente CDC e CDC executadas em tabelas de origem Oracle criadas usando a instrução. `CREATE TABLE AS`

Os problemas resolvidos na AWS DMS versão 3.4.0 incluem o seguinte:

- Avaliações das tarefas de pré-migração aprimoradas. Para ter mais informações, consulte Ativar e trabalhar com avaliações de pré-migração de uma tarefa.
- Validação de dados para tipos de dados flutuantes, reais e duplos aprimorada.

- O Amazon Redshift como destino foi aprimorado para tratar melhor este erro: “A chave especificada não existe”.
- Suporta configurações de tarefas de carregamento de CDC multiencadeadas, incluindo`ParallelApplyThreads`,, e`ParallelApplyBufferSize`, `ParallelApplyQueuesPerThread` para o Amazon OpenSearch Service (OpenSearch Service) como destino.
- OpenSearch Serviço aprimorado como alvo ao oferecer suporte ao uso de chaves primárias compostas.
- Corrigido um problema em que a conexão de teste falhava ao utilizar o PostgreSQL como origem e a senha continha caracteres especiais.
- Corrigido um problema com a utilização do SQL Server como origem quando algumas colunas `VARCHAR` são truncadas.
- Corrigido um problema em AWS DMS que não fechava transações abertas ao usar o SQL Server do Amazon RDS como fonte. Isso poderá resultar em perda de dados se o parâmetro do intervalo de pesquisa for definido incorretamente. Para obter mais informações sobre como configurar um valor de intervalo de sondagem recomendado, consulte Usando um banco de dados Microsoft SQL Server como fonte para AWS DMS.
- Corrigido um problema no Oracle Standby como origem em que as tarefas da CDC paravam inesperadamente ao utilizar o Binary Reader.
- Corrigido um problema no IBM DB2 for LUW em que a tarefa falhava com a mensagem “O literal numérico 0 não é válido porque seu valor está fora do intervalo”.
- Corrigido um problema na migração do PostgreSQL para o PostgreSQL quando uma nova coluna era adicionada à origem do PostgreSQL, e a coluna era criada com um tipo de dados diferente do tipo de dados do qual a coluna foi originalmente criada na origem.
- Corrigido um problema com uma origem do MySQL em que a tarefa de migração parava inesperadamente quando não era possível buscar logs binários.
- Corrigido um problema relacionado a um destino Oracle quando `BatchApply` estava sendo utilizado.
- Corrigido um problema no MySQL e no MariaDB ao migrar o tipo de dados `TIME`.
- Corrigido um problema em uma origem de IBM DB2 LUW em que a migração de tabelas com LOBs falhava quando as tabelas não tinham uma chave primária ou uma chave exclusiva.

AWS Notas de versão do Database Migration Service 3.3.4

Os problemas resolvidos na AWS DMS versão 3.3.4 incluem o seguinte:

- Corrigido um problema em que as transações eram descartadas ou duplicadas ao utilizar o PostgreSQL como origem.
- Compatibilidade com a utilização de cifrão ($) em nomes de esquemas aprimorada.
- Corrigido um problema em que as instâncias de replicação não fechavam as transações abertas ao utilizar o SQL Server do RDS como origem.
- Corrigido um problema em que a conexão de teste falhava ao utilizar o PostgreSQL como origem e a senha continha caracteres especiais.
- O Amazon Redshift como destino foi aprimorado para tratar melhor este erro: “A chave especificada não existe”.
- Compatibilidade com a validação de dados aprimorada em vários tipos de dados e conjuntos de caracteres.
- Corrigido um problema em que as tarefas de replicação tentavam criar tabelas existentes para o Oracle como destino.
- Corrigido um problema em que as tarefas de replicação não adicionavam um `"OP"` à coluna `"D"` sob certas condições do Amazon S3 como destino.

AWS Notas de versão do Database Migration Service 3.3.3

A tabela a seguir mostra os novos recursos e aprimoramentos introduzidos no AWS DMS versão 3.3.3.

Novo recurso ou aprimoramento	Descrição
Nova versão do PostgreSQL	O PostgreSQL versão 12 agora tem suporte como origem e destino.
Support para chave primária composta com Amazon OpenSearch Service como destino	A partir do AWS DMS 3.3.3, o uso de uma chave primária composta é suportado pelos OpenSearch destinos de serviço.

Novo recurso ou aprimoramento	Descrição
Compatibilidade com os tipos de dados estendidos do Oracle	Os tipos de dados estendidos Oracle para origem e destinos Oracle agora são compatíveis.
Aumento do número de AWS DMS recursos por conta	O limite do número de AWS DMS recursos que você pode criar aumentou. Para ter mais informações, consulte Cotas para o AWS Database Migration Service.

Os problemas resolvidos na AWS DMS versão 3.3.3 incluem o seguinte:

- Corrigido um problema em que uma tarefa falhava ao utilizar uma instrução de atualização específica com a aplicação paralela no Amazon Kinesis.
- Corrigido um problema em que uma tarefa falhava na instrução ALTER TABLE com o Amazon S3 como destino.
- Corrigido um problema em que os valores em colunas de polígono eram truncados o utilizar o Microsoft SQL Server como origem.
- Correção de um problema no conversor Unicode de JA16SJISTILDE e JA16EUCTILDE no uso do Oracle como uma origem.
- Correção de um problema em que as colunas MEDIUMTEXT e LONGTEXT falhavam ao migrar do MySQL para o formato CSV (valor separado por vírgula) do S3.
- Correção de um problema em que colunas boolianas eram transformadas em tipos incorretos com a saída do Apache Parquet.
- Correção de um problema com colunas varchar estendidas no Oracle.
- Correção de um problema em que as tarefas de validação de dados falhavam devido a determinadas combinações de carimbo de data e hora.
- Correção de um problema com a replicação de linguagem de definição de dados Sybase (DDL).
- Correção de um problema envolvendo uma origem do Oracle Real Application Clusters (RAC) falhando com o Oracle Binary Reader.
- Correção de um problema com a validação para destinos do Oracle com maiúsculas e minúsculas de nomes de esquema.
- Correção de um problema com a validação das versões 9.7 e 10 do IBM Db2.

- Correção de um problema para uma tarefa que não parava duas vezes com `StopTaskCachedChangesApplied` e `StopTaskCachedChangesNotApplied` ativados.

Histórico do documento

A tabela a seguir descreve as alterações importantes na documentação do guia do usuário do AWS Database Migration Service após janeiro de 2018.

É possível se inscrever em um feed RSS para ser notificado sobre atualizações nessa documentação. Para obter mais detalhes sobre as liberações de versões do AWS DMS, consulte AWS Notas de versão do DMS.

Alteração	Descrição	Data
O AWS DMS adicionou compatibilidade com o RDS IBM DB2 como destino.	O AWS DMS agora aceita o uso do Amazon RDS IBM DB2 como destino.	4 de dezembro de 2023
O AWS DMS adicionou compatibilidade com o Timestream como destino.	O AWS DMS agora é compatível com o Timestream como destino.	17 de novembro de 2023
O AWS DMS adicionou compatibilidade com a validação de dados de destino do Redshift.	O AWS DMS agora aceita validação de dados em destinos do Redshift.	14 de novembro de 2023
O AWS DMS adicionou compatibilidade com quatro novos tipos de endpoints	Agora, o AWS DMS é compatível com a utilizaçã o dos bancos de dados Microsoft Azure para PostgreSQL, Microsoft Azure para MySQL, OCI MySQL Heatwave e Google Cloud para PostgreSQL como origem.	26 de outubro de 2023
O AWS DMS adicionou compatibilidade com um novo perfil vinculado a serviço da AWS	O AWS DMS agora é compatível com o perfil vinculado a serviço da AWS, `AWSServiceRoleForD`	22 de maio de 2023

	`MSServerless` , que permite que o AWS DMS crie e gerencie recursos em seu nome, como publicar pontos de dados de métricas no Amazon CloudWatch.	
O AWS DMS adicionou compatibilidade com uma nova política gerenciada pela AWS	O AWS DMS agora é compatível com a política gerenciada pela AWS que permite publicar logs de replicação com tecnologia sem servidor no CloudWatch Logs.	22 de maio de 2023
O AWS DMS adicionou compatibilidade com uma nova política gerenciada pela AWS	O AWS DMS agora é compatível com a política gerenciada pela AWS que permite publicar pontos de dados de métricas no Amazon CloudWatch. Além disso, o AWS DMS começou a rastrear as alterações em políticas gerenciadas pela AWS.	6 de março de 2023
Compatibilidade com endpoints de origem e de destino da VPC	O AWS DMS agora é compatível com endpoints da nuvem privada virtual (VPC) como origens e destinos. O AWS DMS agora pode se conectar a qualquer serviço da AWS com endpoints da VPC quando rotas explicitamente definidas para os serviços são definidas na VPC do AWS DMS.	30 de junho de 2022

Compatibilidade com a réplica de leitura do SQL Server como origem	O AWS DMS agora é compatível com a réplica de leitura do SQL Server como origem. utilizando o AWS DMS, agora é possível executar migrações em tempo real da réplica de leitura do SQL Server para qualquer destino compatível com o AWS DMS.	30 de junho de 2022
Compatibilidade com bancos de dados IBM Db2 z/OS como origem somente para carga máxima	O AWS DMS agora é compatível com bancos de dados IBM Db2 z/OS como origem. Utilizando o AWS DMS, agora é possível executar migrações em tempo real de mainframes do Db2 para qualquer destino compatível com o AWS DMS.	30 de junho de 2022
Compatibilidade com eventos do DMS no EventBridge	O AWS DMS é compatíve l com o gerenciamento de assinaturas de eventos ao utilizar o EventBridge para eventos do DMS.	30 de junho de 2022
Compatibilidade com o Babelfish como destino	O AWS DMS agora é compatível com o Babelfish como destino. Utilizando o AWS DMS, agora é possível migrar dados em tempo real de qualquer origem compatíve l com o AWS DMS para o Babelfish, com tempo de inatividade mínimo.	30 de junho de 2022

Compatibilidade com o Aurora Sem Servidor v2 como destino	O AWS DMS agora é compatível com o Aurora Sem Servidor v2 como destino. Utilizando o AWS DMS, agora é possível executar migrações em tempo real para o Aurora Sem Servidor v2.	30 de junho de 2022
Tutorial de conceitos básicos	Uma atualização do Tutorial de conceitos básicos do AWS DMS. O tutorial utiliza um banco de dados MySQL como origem e um banco de dados PostgreSQL como destino.	20 de maio de 2021
Compatibilidade com o Amazon Neptune como destino	Adicionada compatibilidade com o Amazon Neptune como destino para a migração de dados.	1 de junho de 2020
Compatibilidade com o Apache Kafka como destino	Adição de suporte para Apache Kafka como um destino para migração de dados.	20 de março de 2020
Conteúdo de segurança atualizado	Atualização e padronização do conteúdo de segurança como resposta às solicitações dos clientes.	20 de dezembro de 2019
Migração com o AWS Snowball Edge	Adição de suporte para usar o AWS Snowball Edge para migrar bancos de dados grandes.	24 de janeiro de 2019

Compatibilidade com o Amazon DocumentDB (compatível com MongoDB) como destino	Compatibilidade com o Amazon DocumentDB (compatível com MongoDB) como destino adicionada.	9 de janeiro de 2019
Compatibilidade com o Amazon OpenSearch Service e o Amazon Kinesis Data Streams como destinos	Compatibilidade adicionada para o OpenSearch Service e o Kinesis Data Streams como destinos para a migração de dados.	15 de novembro de 2018
Compatibilidade com o início nativo da CDC	Adição de suporte para pontos de início nativo ao usar captura de dados de alteração (CDC).	28 de junho de 2018
Compatibilidade com o Db2 LUW	Adição de suporte para Db2 LUW da IBM como origem para a migração de dados.	26 de abril de 2018
Compatibilidade com o SQL Server como destino	Adição de suporte para Amazon RDS para Microsoft SQL como origem.	6 de fevereiro de 2018

AWS Glossário

Para obter a terminologia mais recente da AWS, consulte o glossário da AWS na Referência do Glossário da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.