



Guia do Desenvolvedor

AWS Data Pipeline



Versão da API 2012-10-29

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

AWS Data Pipeline: Guia do Desenvolvedor

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

| | |
|--|----|
| | ix |
| O que é AWS Data Pipeline? | 1 |
| Migrando cargas de trabalho do AWS Data Pipeline | 2 |
| Migração de cargas de trabalho para AWS Glue | 3 |
| Migração de cargas de trabalho para AWS Step Functions | 4 |
| Migrar workloads para o Amazon MWSA | 5 |
| Mapear conceitos | 6 |
| Amostras | 7 |
| Serviços relacionados | 8 |
| Acessando AWS Data Pipeline | 9 |
| Definição de preço | 10 |
| Tipos de instância com suporte para as atividades de trabalho do pipeline | 10 |
| Instâncias do Amazon EC2 padrão por região da AWS | 11 |
| Outras instâncias do Amazon EC2 compatíveis | 12 |
| Instâncias do Amazon EC2 compatíveis com clusters do Amazon EMR | 13 |
| AWS Data Pipeline Conceitos | 15 |
| Definição de pipeline | 15 |
| Componentes, instâncias e tentativas de pipeline | 17 |
| Executores de tarefas | 18 |
| Nós de dados | 19 |
| Bancos de dados | 20 |
| Atividades | 20 |
| Precondições | 21 |
| Precondições gerenciadas pelo sistema | 22 |
| Precondições gerenciadas pelo usuário | 22 |
| Recursos | 22 |
| Limites de recurso | 23 |
| Plataformas com suporte | 23 |
| Instâncias spot do Amazon EC2 com clusters do Amazon EMR e o AWS Data Pipeline | 24 |
| Ações | 25 |
| Monitoramento proativo de pipelines | 26 |
| Configurar | 27 |
| Inscreva-se para AWS | 27 |
| Inscreva-se para um Conta da AWS | 27 |

| | |
|--|----|
| Criar um usuário com acesso administrativo | 28 |
| Crie funções do IAM AWS Data Pipeline e recursos de pipeline | 29 |
| Permita que as entidades principais do IAM (usuários e grupos) realizem as ações necessárias | 29 |
| Conceder acesso programático | 31 |
| Conceitos básicos do AWS Data Pipeline | 33 |
| Criar o pipeline | 34 |
| Monitorar o pipeline em execução | 35 |
| Visualizar a saída | 36 |
| Excluir o pipeline | 36 |
| Trabalhar com pipelines | 37 |
| Criar um pipeline | 37 |
| Crie pipelines a partir de modelos de Data Pipeline usando a CLI | 38 |
| Visualizar os pipelines | 58 |
| Interpretar códigos de status do pipeline | 58 |
| Interpretar pipeline e estado de integridade do componente | 60 |
| Visualizar as definições do pipeline | 62 |
| Visualizar detalhes da instância do pipeline | 62 |
| Visualizar logs de pipeline | 63 |
| Editar o pipeline | 65 |
| Limitações | 65 |
| Editar um pipeline usando a AWS CLI | 66 |
| Clonar o pipeline | 66 |
| Marcar o pipeline | 67 |
| Desativar o pipeline | 68 |
| Desativar o pipeline usando a AWS CLI | 69 |
| Excluir o pipeline | 69 |
| Preparar dados e tabelas com atividades | 70 |
| Preparação de dados com ShellCommandActivity | 71 |
| Preparação da tabela com Hive e nós de dados compatíveis com preparação | 73 |
| Preparação da tabela com Hive e nós de dados incompatíveis com preparação | 74 |
| Usar recursos em várias regiões | 75 |
| Falhas e novas execuções em cascata | 78 |
| Atividades | 78 |
| Nós de dados e pré-condições | 78 |
| Recursos | 79 |

| | |
|--|-----|
| Reexecutar objetos de falha em cascata | 79 |
| Falha em cascata e preenchimentos | 79 |
| Sintaxe do arquivo de definição do pipeline | 80 |
| Estrutura do arquivo | 80 |
| Campos de pipeline | 81 |
| Campos definidos pelo usuário | 82 |
| Trabalhar com a API | 83 |
| Instalar o SDK da AWS | 83 |
| Fazer uma solicitação HTTP para o AWS Data Pipeline | 84 |
| Segurança | 89 |
| Proteção de dados | 90 |
| Gerenciamento de identidade e acesso | 91 |
| Políticas do IAM para o AWS Data Pipeline | 92 |
| Exemplo de políticas para o AWS Data Pipeline | 97 |
| Funções do IAM | 100 |
| Registro em log e monitoramento | 108 |
| Informações do AWS Data Pipeline no CloudTrail | 108 |
| Noções básicas das entradas dos arquivos de log do AWS Data Pipeline | 109 |
| Resposta a incidentes | 110 |
| Validação de compatibilidade | 111 |
| Resiliência | 111 |
| Segurança da infraestrutura | 111 |
| Análise de vulnerabilidade e configuração no AWS Data Pipeline | 112 |
| Tutoriais | 113 |
| Processe dados usando a Amazon EMR com o Hadoop Streaming | 113 |
| Antes de começar | 114 |
| Usando o CLI | 114 |
| Copiar dados CSV do Amazon S3 para o Amazon S3 | 118 |
| Antes de começar | 120 |
| Uso da CLI | 120 |
| Exportar dados do MySQL para o Amazon S3 | 127 |
| Antes de começar | 128 |
| Uso da CLI | 129 |
| Copiar dados para o Amazon Redshift | 139 |
| Antes de começar: configurar opções COPY | 139 |
| Antes de começar: configurar pipeline, segurança e cluster | 140 |

| | |
|--|-----|
| Uso da CLI | 142 |
| Expressões e funções do pipeline | 152 |
| Tipos de dados simples | 152 |
| DateTime | 152 |
| Numérico | 152 |
| Referências de objeto | 152 |
| Período | 153 |
| Segmento | 153 |
| Expressões | 153 |
| Referenciar campos e objetos | 154 |
| Expressões aninhadas | 155 |
| Listas | 156 |
| Expressão de nó | 156 |
| Avaliação de expressões | 157 |
| Funções matemáticas | 158 |
| Funções de string | 158 |
| Funções de data e hora | 159 |
| Caracteres especiais | 167 |
| Referência de objeto de pipeline | 169 |
| Nós de dados | 170 |
| ynameDBDataNode D | 171 |
| MySqlDataNode | 178 |
| RedshiftDataNode | 186 |
| S3 DataNode | 194 |
| SqlDataNode | 202 |
| Atividades | 210 |
| CopyActivity | 211 |
| EmrActivity | 219 |
| HadoopActivity | 229 |
| HiveActivity | 241 |
| HiveCopyActivity | 251 |
| PigActivity | 261 |
| RedshiftCopyActivity | 276 |
| ShellCommandActivity | 291 |
| SqlActivity | 302 |
| Recursos | 311 |

| | |
|---|-----|
| Ec2Resource | 311 |
| EmrCluster | 322 |
| HttpProxy | 354 |
| Precondições | 357 |
| D ynamoDBData Existe | 358 |
| D ynamoDBTable Existe | 362 |
| Existe | 366 |
| S3 KeyExists | 371 |
| S3 PrefixNotEmpty | 376 |
| ShellCommandPrecondition | 381 |
| Bancos de dados | 386 |
| JdbcDatabase | 386 |
| RdsDatabase | 388 |
| RedshiftDatabase | 391 |
| Formatos de dados | 393 |
| CSVFormato de dados | 393 |
| Formato de dados personalizado | 395 |
| ynamoDBDataFormato D | 397 |
| D ynamoDBExport DataFormat | 400 |
| RegEx Formato de dados | 403 |
| TSVFormato de dados | 405 |
| Ações | 407 |
| SnsAlarm | 407 |
| Encerrar | 409 |
| Schedule | 411 |
| Exemplos | 411 |
| Sintaxe | 416 |
| Utilitários | 418 |
| ShellScriptConfig | 418 |
| EmrConfiguration | 420 |
| Propriedade | 425 |
| Trabalhar com o Task Runner | 428 |
| Executor de tarefas em recursos AWS Data Pipeline gerenciados | 428 |
| Executar trabalho em recursos existentes usando o Task Runner | 430 |
| Instalando o Task Runner | 432 |
| (Opcional) Concedendo acesso ao Task Runner à Amazon RDS | 432 |

| | |
|--|-----|
| Iniciar o Task Runner | 434 |
| Verificando o registro do Task Runner | 435 |
| Threads e pré-condições do Task Runner | 435 |
| Opções de configuração do Task Runner | 436 |
| Usar o Task Runner com um proxy | 438 |
| Task Runner e Custom AMIs | 438 |
| Solução de problemas | 440 |
| Localizar erros em pipelines | 440 |
| Identificar o cluster do Amazon EMR que serve seu pipeline | 441 |
| Interpretar detalhes de status do pipeline | 441 |
| Localizar logs de erro | 443 |
| Logs de pipeline | 443 |
| Logs de trabalho do Hadoop e Amazon EMR | 444 |
| Resolver problemas comuns | 444 |
| Pipeline preso em status pendente | 445 |
| Componente de pipeline preso no status Waiting for Runner | 445 |
| Componente de pipeline preso no status WAITING_ON_DEPENDENCIES | 446 |
| A execução não inicia quando programada | 447 |
| Os componentes do pipeline são executados na ordem errada | 447 |
| O cluster do EMR falha com erro: o token de segurança incluído na solicitação é inválido ... | 448 |
| Permissões insuficientes para acessar recursos | 448 |
| Código de status: 400 Error Code: PipelineNotFoundException | 448 |
| Criar um pipeline provoca um erro de token de segurança | 448 |
| Não é possível ver detalhes do pipeline no console | 449 |
| Erro no código de status do executor remoto: 404, AWS Service: Amazon S3 | 449 |
| Acesso negado – Não autorizado para executar a função datapipeline: | 449 |
| AMIs do Amazon EMR mais antigas podem criar dados falsos em arquivos CSV grandes .. | 450 |
| Aumentar limites do AWS Data Pipeline | 450 |
| Limites | 451 |
| Limites da conta | 451 |
| Limites de chamada do serviço web | 452 |
| Considerações sobre escalabilidade | 454 |
| Recursos da AWS Data Pipeline | 455 |
| Histórico do documento | 457 |

AWS Data Pipeline não está mais disponível para novos clientes. Os clientes existentes do AWS Data Pipeline podem continuar usando o serviço normalmente. [Saiba mais](#)

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.

O que é AWS Data Pipeline?

Note

AWS Data Pipeline o serviço está em modo de manutenção e nenhum novo recurso ou expansão de região está planejado. Para saber mais e descobrir como migrar os workloads existentes, consulte [Migrando cargas de trabalho do AWS Data Pipeline](#).

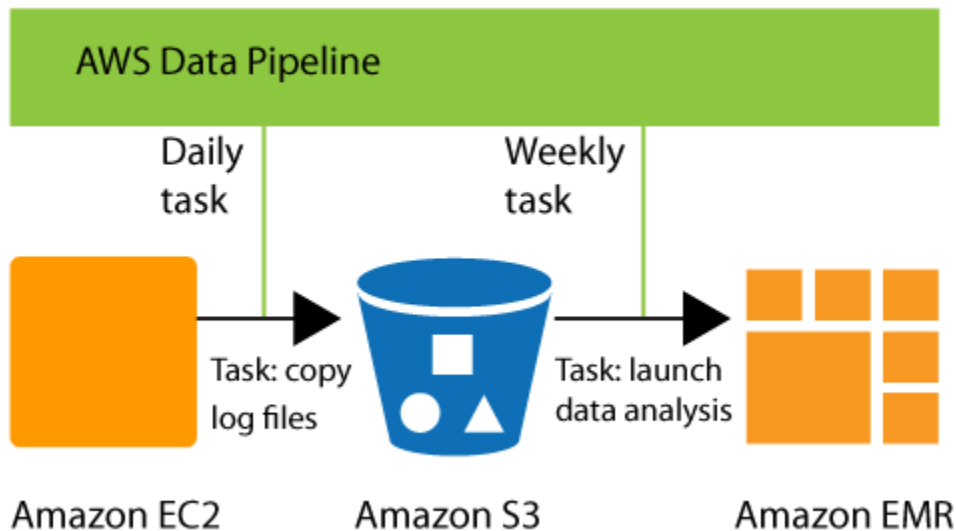
AWS Data Pipeline é um serviço web que você pode usar para automatizar a movimentação e a transformação de dados. Com AWS Data Pipeline, você pode definir fluxos de trabalho orientados por dados, para que as tarefas possam depender da conclusão bem-sucedida das tarefas anteriores. Você define os parâmetros de suas transformações de dados e AWS Data Pipeline aplica a lógica que você configurou.

Os seguintes componentes do AWS Data Pipeline trabalho em conjunto para gerenciar seus dados:

- Uma definição de pipeline especifica a lógica de negócios do seu gerenciamento de dados. Para ter mais informações, consulte [Sintaxe do arquivo de definição do pipeline](#).
- Um pipeline programa e executa tarefas por meio da criação de instâncias do Amazon EC2 para realizar atividades de trabalho definidas. Você faz upload da sua definição de pipeline no pipeline e, em seguida, o ativa. Você pode editar a definição de pipeline para um pipeline em execução e ativá-lo novamente para que essa definição entre em vigor. Você pode desativar o pipeline, modificar uma fonte de dados e, em seguida, ativar o pipeline novamente. Quando não precisar mais do pipeline, você poderá excluí-lo.
- O Task Runner pesquisará tarefas e as executará. Por exemplo, o Task Runner pode copiar arquivos de log para o Amazon S3 e iniciar clusters do Amazon EMR. O Task Runner é instalado e executado automaticamente nos recursos criados pelas suas definições de pipeline. Você pode escrever um aplicativo de execução de tarefas personalizado ou usar o aplicativo Executor de Tarefas fornecido pela AWS Data Pipeline. Para ter mais informações, consulte [Executores de tarefas](#).

Por exemplo, você pode usar AWS Data Pipeline para arquivar os registros do seu servidor web no Amazon Simple Storage Service (Amazon S3) todos os dias e, em seguida, executar um cluster semanal do Amazon EMR (Amazon EMR) sobre esses registros para gerar relatórios de tráfego. AWS Data Pipeline programa as tarefas diárias para copiar dados e a tarefa semanal para iniciar o

cluster do Amazon EMR. AWS Data Pipeline também garante que o Amazon EMR aguarde o upload dos dados do último dia para o Amazon S3 antes de iniciar sua análise, mesmo que haja um atraso imprevisto no upload dos registros.



Conteúdo

- [Migrando cargas de trabalho do AWS Data Pipeline](#)
- [Serviços relacionados](#)
- [Acessando AWS Data Pipeline](#)
- [Definição de preço](#)
- [Tipos de instância com suporte para as atividades de trabalho do pipeline](#)

Migrando cargas de trabalho do AWS Data Pipeline

AWS lançou o AWS Data Pipeline serviço em 2012. Naquela época, os clientes procuravam um serviço que os ajudasse a mover dados de forma confiável entre diferentes fontes de dados usando uma variedade de opções de computação. Agora, existem outros serviços que oferecem aos clientes uma experiência melhor. Por exemplo, você pode usar AWS Glue to para executar e orquestrar aplicativos Apache Spark, Step AWS Functions para ajudar a orquestrar AWS componentes de serviço ou Amazon Managed Workflows for Apache Airflow (Amazon MWAA) para ajudar a gerenciar a orquestração do fluxo de trabalho para o Apache Airflow.

Este tópico explica como AWS Data Pipeline migrar das opções alternativas. A opção escolhida depende de sua workload atual em AWS Data Pipeline. Você pode migrar casos de uso típicos AWS Data Pipeline para AWS Step Functions ou Amazon MWAA. AWS Glue

Migração de cargas de trabalho para AWS Glue

O [AWS Glue](#) é um serviço de integração de dados com tecnologia sem servidor que facilita aos usuários de análise a descoberta, preparação, transferência e integração de dados de várias fontes. Inclui ferramentas para criação, execução de trabalhos e orquestração de fluxos de trabalho. Com AWS Glue, você pode descobrir e se conectar a mais de 70 fontes de dados diversas e gerenciar seus dados em um catálogo de dados centralizado. Você pode criar, executar e monitorar visualmente pipelines de extração, transformação e carregamento (ETL) para carregar dados em seus data lakes. Além disso, é possível pesquisar e consultar imediatamente os dados catalogados usando o Amazon Athena, o Amazon EMR e o Amazon Redshift Spectrum.

Recomendamos migrar sua AWS Data Pipeline carga de trabalho para AWS Glue quando:

- Você estiver procurando um serviço de integração de dados com tecnologia sem servidor que ofereça suporte para várias fontes de dados, interfaces de criação, incluindo editores visuais e notebooks, e recursos avançados de gerenciamento de dados, como qualidade de dados e detecção de dados sensíveis.
- Sua carga de trabalho pode ser migrada para AWS Glue fluxos de trabalho, trabalhos (em Python ou Apache Spark) e rastreadores (por exemplo, seu pipeline existente é construído com base no Apache Spark).
- Você precisar de uma plataforma única que possa lidar com todos os aspectos do seu pipeline de dados, incluindo ingestão, processamento, transferência, testes de integridade e verificações de qualidade.
- Seu pipeline existente foi criado a partir de um modelo predefinido no AWS Data Pipeline console, como a exportação de uma tabela do DynamoDB para o Amazon S3, e você está procurando o mesmo modelo de propósito.
- Seu workload não depender de uma aplicação específica do ecossistema Hadoop, como o Apache Hive.
- Seu workload não exigir orquestração de servidores on-premises.

AWS cobra uma taxa horária, cobrada por segundo, para rastreadores (descoberta de dados) e tarefas de ETL (processamento e carregamento de dados). AWS Glue O Studio é um mecanismo de

orquestração integrado para AWS Glue recursos e é oferecido sem custo adicional. Para saber mais sobre a definição de preço, consulte [Definição de preço da AWS Glue](#).

Migração de cargas de trabalho para AWS Step Functions

AWS O [Step Functions](#) é um serviço de orquestração sem servidor que permite criar fluxos de trabalho para seus aplicativos essenciais aos negócios. Com o Step Functions, você usa um editor visual para criar fluxos de trabalho e integrar-se diretamente a mais de 11.000 ações para mais de 250 AWS serviços, como AWS Lambda, Amazon EMR, DynamoDB e muito mais. Você pode usar o Step Functions para orquestrar pipelines de processamento de dados, lidar com erros e trabalhar com os limites de limitação nos serviços subjacentes. AWS Você pode criar fluxos de trabalho que processam e publicam modelos de aprendizado de máquina, orquestram microsserviços e controlam AWS serviços, como AWS Glue criar fluxos de trabalho de extração, transformação e carregamento (ETL). Você também pode criar fluxos de trabalho automatizados e de longa duração para aplicações que exigem interação humana.

Da mesma forma AWS Data Pipeline, o AWS Step Functions é um serviço totalmente gerenciado fornecido pela AWS. Você não precisará gerenciar a infraestrutura, aplicar patches em workers, gerenciar atualizações da versão do sistema operacional ou similares.

Recomendamos migrar sua AWS Data Pipeline carga de trabalho para o AWS Step Functions quando:

- Você estiver procurando um serviço de orquestração de fluxo de trabalho com tecnologia sem servidor e altamente disponível.
- Você estiver procurando uma solução econômica que faça a cobrança pela granularidade da execução de uma única tarefa.
- Suas cargas de trabalho estão orquestrando tarefas para vários outros AWS serviços, como Amazon EMR, Lambda ou DynamoDB. AWS Glue
- Você está procurando uma solução low-code que venha com um designer drag-and-drop visual para criação de fluxo de trabalho e que não exija o aprendizado de novos conceitos de programação.
- Você está procurando um serviço que forneça integrações com mais de 250 outros AWS serviços cobrindo mais de 11.000 ações out-of-the-box, além de permitir integrações com atividades e não AWS serviços personalizados.

AWS Data Pipeline Tanto o Step Functions quanto o Step Functions usam o formato JSON para definir fluxos de trabalho. Isso permite armazenar seus fluxos de trabalho no controle de origem, gerenciar versões, controlar o acesso e automatizar com CI/CD. O Step Functions está usando uma sintaxe chamada Amazon State Language, que é totalmente baseada em JSON e permite uma transição perfeita entre as representações textuais e visuais do fluxo de trabalho.

Com o Step Functions, você pode escolher a mesma versão do Amazon EMR que você está usando atualmente no AWS Data Pipeline.

Para migrar atividades em recursos AWS Data Pipeline gerenciados, você pode usar a [integração de serviços AWS SDK](#) no Step Functions para automatizar o provisionamento e a limpeza de recursos.

Para migrar atividades em servidores on-premises, instâncias do EC2 gerenciadas pelo usuário ou um cluster do EMR gerenciado pelo usuário, você pode instalar um [agente SSM](#) na instância. Você pode iniciar o comando por meio do [Run Command do AWS Systems Manager](#) a partir do Step Functions. Você também pode iniciar a máquina de estado a partir da programação definida na [Amazon EventBridge](#).

AWS O Step Functions tem dois tipos de fluxos de trabalho: fluxos de trabalho padrão e fluxos de trabalho expressos. Para fluxos de trabalho padrão, a cobrança é efetuada com base no número de transições de estado necessárias para executar sua aplicação. Para fluxos de trabalho expressos, a cobrança é efetuada com base no número de solicitações do seu fluxo de trabalho e na duração. Saiba mais sobre preços em [Definição de preços do AWS Step Functions](#).

Migrar workloads para o Amazon MWAA

[O Amazon MWAA](#) (Managed Workflows for Apache Airflow) é um serviço gerenciado de orquestração para o [Apache Airflow](#) que facilita a configuração e a operação de pipelines de dados na nuvem em grande escala. end-to-end O Apache Airflow é uma ferramenta de código aberto usada para criar, agendar e monitorar por meio de programação sequências de processos e tarefas chamadas de “fluxos de trabalho”. Com o Amazon MWAA, você pode usar o Airflow e a linguagem de programação Python para criar fluxos de trabalho sem precisar gerenciar a infraestrutura subjacente para fins de escalabilidade, disponibilidade e segurança. O Amazon MWAA escala automaticamente sua capacidade de execução de fluxo de trabalho para atender às suas necessidades e é integrado aos serviços de AWS segurança para ajudar a fornecer acesso rápido e seguro aos seus dados.

Da mesma forma AWS Data Pipeline, o Amazon MWAA é um serviço totalmente gerenciado fornecido pela. AWS Embora seja necessário aprender vários novos conceitos específicos desses

serviços, não é necessário gerenciar a infraestrutura, aplicar patches em workers, gerenciar atualizações de versões do sistema operacional ou similares.

Recomendamos migrar suas AWS Data Pipeline cargas de trabalho para o Amazon MWAA quando:

- Você estiver procurando um serviço gerenciado e altamente disponível para orquestrar fluxos de trabalho escritos em Python.
- Você desejar fazer a transição para uma tecnologia de código aberto totalmente gerenciada e amplamente adotada, como o Apache Airflow, para máxima portabilidade.
- Você precisar de uma plataforma única que possa lidar com todos os aspectos do seu pipeline de dados, incluindo ingestão, processamento, transferência, testes de integridade e verificações de qualidade.
- Você estiver procurando um serviço projetado para orquestração de pipeline de dados com recursos como interface avançada para observabilidade, reinicializações para fluxos de trabalho com falha, preenchimentos e novas tentativas de tarefas.
- Você está procurando um serviço que venha com mais de 800 operadores e sensores pré-construídos, AWS abrangendo e não AWS serviços.

Os fluxos de trabalho do Amazon MWAA são definidos como Directed Acyclic Graphs (DAGs – Gráficos acíclicos dirigidos) usando Python, então você também pode tratá-los como código-fonte. A estrutura Python extensível do Airflow permite que você crie fluxos de trabalho conectados a praticamente qualquer tecnologia. Ele vem com uma interface de usuário avançada para visualizar e monitorar fluxos de trabalho e pode ser facilmente integrado aos sistemas de controle de versão para automatizar o processo de CI/CD.

Com o Amazon MWAA, você pode escolher a mesma versão do Amazon EMR que você está usando atualmente na AWS Data Pipeline.

AWS cobranças pelo tempo em que seu ambiente Airflow é executado, além de qualquer escalonamento automático adicional para fornecer mais capacidade de trabalho ou servidor web. Saiba mais sobre preços no [Fluxos de trabalho gerenciados da Amazon para o Apache Airflow](#).

Mapear conceitos

A tabela a seguir contém o mapeamento dos principais conceitos usados pelos serviços. Isso ajudará aqueles familiarizados com o Data Pipeline a entender a terminologia do Step Functions e do MWAA.

| Data Pipeline | União | Step Functions | Amazon MWAA |
|----------------------------------|---|--|---|
| Pipelines | Fluxos de trabalho | Fluxos de trabalho | Direct acylic graphs |
| Definição de pipeline em JSON | Definição de fluxo de trabalho ou esquemas baseados em Python | Amazon State Language em JSON | Baseado em Python |
| Atividades | Trabalhos | Estados e tarefas | Tarefas (operadores e sensores) |
| Instâncias | Execuções de trabalhos | Execuções | Executado por DAG |
| Attempts | Novo attempt | Catchers e retriers | Retries |
| Cronograma do pipeline | Trigger programado | EventBridge Tarefas do agendador | Cron, timetables e data-aware |
| Expressões e funções de pipeline | Biblioteca de esquema | Funções intrínsecas do Step Functions e Lambda AWS | Estrutura Python extensível |

Amostras

As seções a seguir listam exemplos públicos que você pode consultar para AWS Data Pipeline migrar de serviços individuais. Você pode citá-los como exemplos e criar seu próprio pipeline nos serviços individuais atualizando e testando o pipeline com base no seu caso de uso.

AWS Glue amostras

A lista a seguir contém exemplos de implementações para os casos de AWS Data Pipeline uso mais comuns com. AWS Glue

- [Execução de trabalhos do Spark](#)
- [Copiar dados do JDBC para o Amazon S3](#) (incluindo o Amazon Redshift)
- [Copiar dados do Amazon S3 para o JDBC](#) (incluindo o Amazon Redshift)
- [Copiar dados do Amazon S3 para o DynamoDB](#)

- [Importar e exportar dados do Amazon Redshift](#)
- [Acesso a tabelas do DynamoDB entre contas e entre regiões](#)

AWS Exemplos de Step Functions

A lista a seguir contém exemplos de implementações para os AWS Data Pipeline casos de uso mais comuns com Step Functions AWS .

- [Gerenciar um trabalho do Amazon EMR](#)
- [Executar um trabalho de processamento de dados no Amazon EMR Serverless](#)
- [Executar trabalhos do Hie/Pig/Hadoop](#)
- [Consultando grandes conjuntos de dados](#) (Amazon Athena, Amazon S3,) AWS Glue
- [Executar fluxos de trabalho de ETL usando o Amazon Redshift](#)
- [AWS Glue Orquestrando rastreadores](#)

Veja [tutoriais](#) adicionais e [exemplos de projetos](#) para usar o AWS Step Functions.

Amostras do Amazon MWAA

A lista a seguir contém exemplos de implementações para os casos de AWS Data Pipeline uso mais comuns com o Amazon MWAA.

- [Executar um trabalho do Amazon EMR](#)
- [Criar um plug-in personalizado para Apache Hive e Hadoop](#)
- [Copiar dados do Amazon S3 para o Redshift](#)
- [Executar um script Shell em uma instância remota do EC2](#)
- [Orquestrar fluxos de trabalho híbridos \(on-premisses\)](#)

Veja [tutoriais](#) adicionais e [exemplos de projetos](#) para usar o Amazon MWAA.

Serviços relacionados

AWS Data Pipeline funciona com os seguintes serviços para armazenar dados.

- Amazon DynamoDB – Fornece um banco de dados NoSQL totalmente gerenciado com desempenho rápido por um baixo custo. Para obter mais informações, consulte o [Guia do desenvolvedor do Amazon DynamoDB](#).
- Amazon RDS – Fornece um banco de dados relacional totalmente gerenciado que é dimensionado para conjuntos de dados grandes. Para obter mais informações, consulte o [Guia do desenvolvedor para serviços do banco de dados relacional da Amazon](#).
- Amazon Redshift – Fornece um data warehouse rápido, totalmente gerenciado e em escala de petabytes, que torna simples e rentável analisar uma grande quantidade de dados. Para obter mais informações, consulte o [Guia do desenvolvedor do banco de dados do Amazon Redshift](#).
- Amazon S3 – Fornece armazenamento de objetos seguro, durável e altamente escalável. Para obter mais detalhes, consulte o [Guia do usuário do Amazon Simple Storage Service](#).

AWS Data Pipeline trabalha com os seguintes serviços de computação para transformar dados.

- Amazon EC2 – Fornece capacidade de computação redimensionável (literalmente, servidores nos datacenters da Amazon) que você pode usar para criar e hospedar seus sistemas de software. Para obter mais informações, consulte o Guia [do usuário do Amazon EC2](#).
- Amazon EMR – Torna fácil, rápido e econômico distribuir e processar grandes quantidades de dados nos servidores Amazon EC2 por meio de estruturas como o Apache Hadoop ou o Apache Spark. Para obter mais informações, consulte o [Guia do desenvolvedor do Amazon EMR](#).

Acessando AWS Data Pipeline

Você pode criar, acessar e gerenciar seus pipelines usando qualquer uma das seguintes interfaces:

- AWS Management Console — Fornece uma interface web que você pode usar para acessar o AWS Data Pipeline.
- AWS Command Line Interface (AWS CLI) — Fornece comandos para um amplo conjunto de serviços da AWS, inclusive AWS Data Pipeline, e é compatível com Windows, macOS e Linux. Para obter mais informações sobre a instalação do AWS CLI, consulte [AWS Command Line Interface](#). Para obter uma lista de comandos para AWS Data Pipeline, consulte [datapipeline](#).
- AWS SDKs: fornecem APIs específicas da linguagem e cuidam de muitos dos detalhes da conexão, como cálculo de assinaturas, tratamento de novas tentativas de solicitação e tratamento de erros. Para mais informações, consulte [SDKs da AWS](#).

- **API de consulta** — Fornece APIs de baixo nível que você chama usando solicitações HTTPS. Usar a API de consulta é a maneira mais direta para acessar a AWS Data Pipeline, mas exige que seu aplicativo lide com detalhes de baixo nível, como a geração de hash para assinar a solicitação e manuseio de erros. Para obter mais informações, consulte a [Referência da API do AWS Data Pipeline](#).

Definição de preço

Com o Amazon Web Services, você paga somente pelo que usar. Pois AWS Data Pipeline, você paga pelo seu funil com base na frequência com que suas atividades e condições prévias estão programadas para serem executadas e onde elas são executadas. Para obter mais informações, consulte [Preços do AWS Data Pipeline](#).

Se sua conta da AWS tiver menos de 12 meses, você poderá usar o nível gratuito. O nível gratuito inclui três condições e cinco atividades mensais, ambas de baixa frequência, sem qualquer custo. Para obter mais informações, consulte [AWS Free Tier \(Nível gratuito da AWS\)](#).

Tipos de instância com suporte para as atividades de trabalho do pipeline

Quando AWS Data Pipeline executa um pipeline, ele compila os componentes do pipeline para criar um conjunto de instâncias acionáveis do Amazon EC2. Cada instância contém todas as informações para execução de uma tarefa específica. O conjunto completo de instâncias é a lista de tarefas do pipeline. O AWS Data Pipeline entrega as instâncias aos executores de tarefas para processamento.

Instâncias do EC2 acompanham diferentes configurações, que são conhecidas como tipos de instâncias. Cada tipo de instância tem uma capacidade diferente de CPU, entrada/saída e armazenamento. Além de especificar o tipo de instância para uma atividade, você pode escolher diferentes opções de compra. Nem todos os tipos de instâncias estão disponíveis em todas as regiões da AWS. Se um tipo de instância não estiver disponível, o seu pipeline poderá apresentar falha na provisão ou travar no provisionamento. Para obter informações sobre a disponibilidade da instância, consulte a [Página de definição de preço do Amazon EC2](#). Abra o link para a opção de compra da instância e filtre por Region para ver se há algum tipo de instância disponível na região. Para obter mais informações sobre esses tipos de instâncias, famílias e tipos de virtualização, consulte [Instâncias do Amazon EC2](#) e [Matriz de tipo de instância da Amazon Linux AMI](#).

As tabelas a seguir descrevem os tipos de instância AWS Data Pipeline compatíveis. Você pode usar AWS Data Pipeline para iniciar instâncias do Amazon EC2 em qualquer região, incluindo regiões onde não AWS Data Pipeline há suporte. Para obter informações sobre as regiões em que AWS Data Pipeline há suporte, consulte [Regiões e endpoints da AWS](#).

Conteúdo

- [Instâncias do Amazon EC2 padrão por região da AWS](#)
- [Outras instâncias do Amazon EC2 compatíveis](#)
- [Instâncias do Amazon EC2 compatíveis com clusters do Amazon EMR](#)

Instâncias do Amazon EC2 padrão por região da AWS

Se você não especificar um tipo de instância na definição de pipeline, o AWS Data Pipeline executará uma instância por padrão.

A tabela a seguir lista as instâncias do Amazon EC2 que são AWS Data Pipeline usadas por padrão nas regiões em que AWS Data Pipeline há suporte.

| Nome da região | Região | Tipo de instância |
|-----------------------------------|----------------|-------------------|
| Leste dos EUA (Norte da Virgínia) | us-east-1 | m1.small |
| Oeste dos EUA (Oregon) | us-west-2 | m1.small |
| Ásia-Pacífico (Sydney) | ap-southeast-2 | m1.small |
| Ásia-Pacífico (Tóquio) | ap-northeast-1 | m1.small |
| UE (Irlanda) | eu-west-1 | m1.small |

A tabela a seguir lista as instâncias do Amazon EC2 que são AWS Data Pipeline executadas por padrão nas regiões em que não AWS Data Pipeline há suporte.

| Nome da região | Região | Tipo de instância |
|----------------------|-----------|-------------------|
| Leste dos EUA (Ohio) | us-east-2 | t2.small |

| Nome da região | Região | Tipo de instância |
|-------------------------------------|----------------|-------------------|
| Oeste dos EUA (Norte da Califórnia) | us-west-1 | m1.small |
| Ásia-Pacífico (Mumbai) | ap-south-1 | t2.small |
| Ásia-Pacífico (Singapura) | ap-southeast-1 | m1.small |
| Ásia-Pacífico (Seul) | ap-northeast-2 | t2.small |
| Canadá (Central) | ca-central-1 | t2.small |
| UE (Frankfurt) | eu-central-1 | t2.small |
| UE (Londres) | eu-west-2 | t2.small |
| UE (Paris) | eu-west-3 | t2.small |
| América do Sul (São Paulo) | sa-east-1 | m1.small |

Outras instâncias do Amazon EC2 compatíveis

Veja a seguir as instâncias compatíveis, além das instâncias padrão que são criadas se você não especificar um tipo de instância na definição do seu pipeline.

A tabela a seguir lista as instâncias do Amazon EC2 que oferecem AWS Data Pipeline suporte e podem ser criadas, se especificadas.

| Classe de instância | Instance Types (Tipos de instâncias) |
|----------------------------|---|
| Propósito geral | t2.nano t2.micro t2.small t2.medium t2.large |
| Otimizadas para computação | c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c5.xlarge c5.9xlarge c5.2xlarge c5.4xlarge c5.9xlarge c5.18xlarge c5d.xlarge c5d.2xlarge c5d.4xlarge c5d.9xlarge c5d.18xlarge |

| Classe de instância | Instance Types (Tipos de instâncias) |
|------------------------------|--|
| Otimizado para memória | m3.medium m3.large m3.xlarge m3.2xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge m5.xlarge m5.2xlarge m5.4xlarge m5.12xlarge m5.24xlarge m5d.xlarge m5d.2xlarge m5d.4xlarge m5d.12xlarge m5d.24xlarge r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge |
| Otimizada para armazenamento | i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge hs1.8xlarge g2.2xlarge g2.8xlarge d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge |

Instâncias do Amazon EC2 compatíveis com clusters do Amazon EMR

Esta tabela lista as instâncias do Amazon EC2 que oferecem AWS Data Pipeline suporte e podem ser criadas para clusters do Amazon EMR, se especificado. Para obter mais informações, consulte [Tipos de instâncias compatíveis](#) no Guia de gerenciamento do Amazon EMR.

| Classe de instância | Instance Types (Tipos de instâncias) |
|----------------------------|--|
| Propósito geral | m1.small m1.medium m1.large m1.xlarge m3.xlarge m3.2xlarge |
| Otimizadas para computação | c1.medium c1.xlarge c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge cc1.4xlarge cc2.8xlarge c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c5.xlarge c5.9xlarge c5.2xlarge c5.4xlarge c5.9xlarge c5.18xlarge c5d.xlarge c5d.2xlarge c5d.4xlarge c5d.9xlarge c5d.18xlarge |
| Otimizado para memória | m2.xlarge m2.2xlarge m2.4xlarge r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge cr1.8xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge m5.xlarge m5.2xlarge m5.4xlarge m5.12xlarge m5.24xlarge m5d.xlarge m5d.2xlarge m5d.4xlarge m5d.12xlarge m5d.24xlarge |

| Classe de instância | Instance Types (Tipos de instâncias) |
|------------------------------|--|
| | r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge |
| Otimizada para armazenamento | h1.4xlarge hs1.2xlarge hs1.4xlarge hs1.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge |
| Computação acelerada | g2.2xlarge cg1.4xlarge |

AWS Data Pipeline Conceitos

Antes de começar, leia sobre os principais conceitos e componentes do AWS Data Pipeline.

Conteúdo

- [Definição de pipeline](#)
- [Componentes, instâncias e tentativas de pipeline](#)
- [Executores de tarefas](#)
- [Nós de dados](#)
- [Bancos de dados](#)
- [Atividades](#)
- [Precondições](#)
- [Recursos](#)
- [Ações](#)

Definição de pipeline

Uma definição de pipeline é como você comunica sua lógica de negócios com AWS Data Pipeline.

Ela contém as seguintes informações:

- Nomes, locais e formatos das suas fontes de dados
- Atividades que transformam os dados
- A programação dessas atividades
- Recursos que executam suas atividades e precondições
- Precondições que precisam ser atendidas antes que as atividades sejam programadas
- Maneiras de alertar você com atualizações de status à medida que a execução do pipeline prossegue

A partir da definição do pipeline, AWS Data Pipeline determina as tarefas, as agenda e as atribui aos executores de tarefas. Se uma tarefa não for concluída com êxito, AWS Data Pipeline tente novamente a tarefa de acordo com suas instruções e, se necessário, a reatribua a outro executor de tarefas. Se a tarefa falhar repetidamente, você poderá configurar o pipeline para lhe notificar.

Por exemplo, na definição do seu pipeline, você pode especificar que os arquivos de log gerados pelo seu aplicativo sejam arquivados a cada mês, para o ano de 2013, em um bucket do Amazon S3. O AWS Data Pipeline criará 12 tarefas, cada uma copiando os dados correspondentes a um mês, independentemente de o mês conter 30, 31, 28 ou 29 dias.

Você pode criar uma definição de pipeline das seguintes formas:

- Graficamente, usando o console AWS Data Pipeline
- Textualmente gravando um arquivo JSON no formato usado pela interface de linha de comando
- Programaticamente chamando o serviço web com um dos SDKs da AWS ou o a [API do AWS Data Pipeline](#)

Uma definição de pipeline pode conter os seguintes tipos de componentes.

Componentes do pipeline

[Nós de dados](#)

O local dos dados de entrada para uma tarefa ou o local em que os dados de saída serão armazenados.

[Atividades](#)

Uma definição do trabalho a ser realizado em uma programação usando um recurso computacional e nós de dados de entrada e saída.

[Precondições](#)

Uma instrução condicional que precisa ser verdadeira para que uma ação possa ser executada.

[Recursos](#)

O recurso computacional que realiza o trabalho definido por esse pipeline.

[Ações](#)

Uma ação que é acionada quando condições especificadas são atendidas, como a falha de uma atividade.

Para ter mais informações, consulte [Sintaxe do arquivo de definição do pipeline](#).

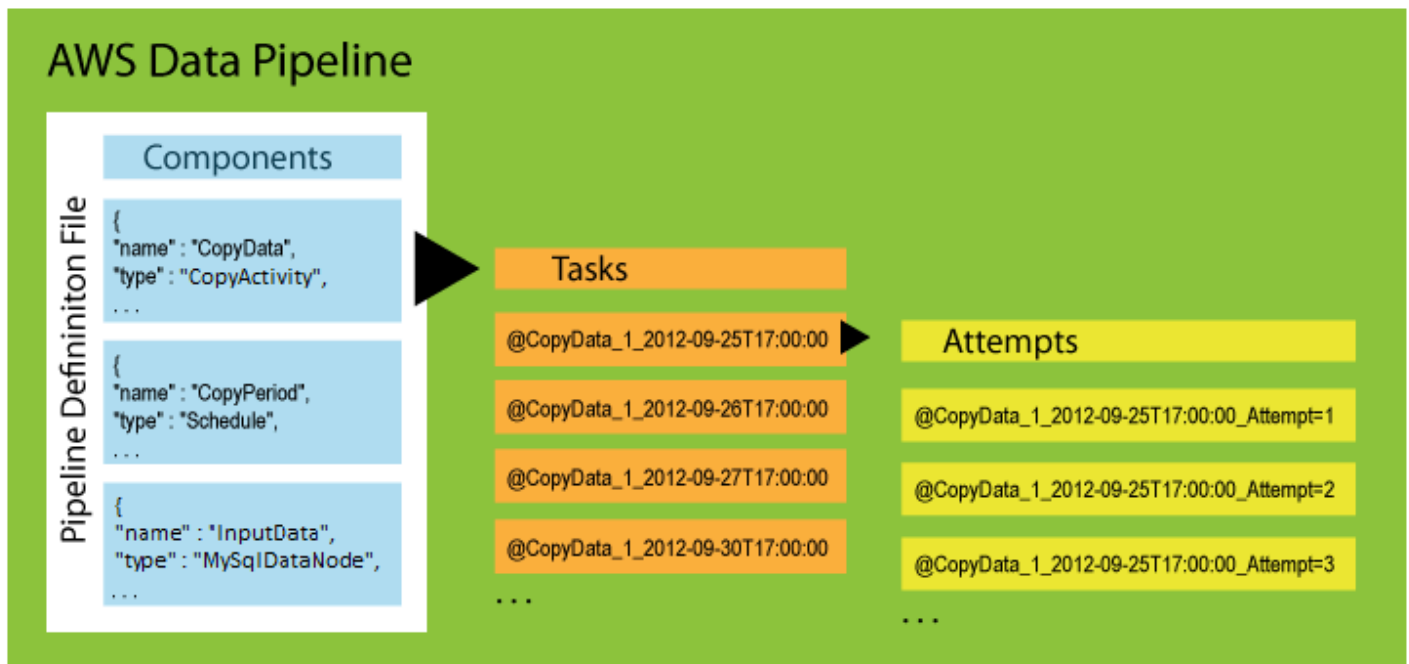
Componentes, instâncias e tentativas de pipeline

Existem três tipos de itens associados a um pipeline programado:

- **Componentes do pipeline** – Os componentes do pipeline representam a lógica de negócios do pipeline e são representados pelas diferentes seções de uma definição de pipeline. Os componentes do pipeline especificam fontes de dados, atividades, programação e precondições do fluxo de trabalho. Eles podem herdar propriedades dos componentes principais. As relações entre os componentes são definidas por referência. Os componentes do pipeline definem as regras de gerenciamento de dados.
- **Instâncias** — Quando AWS Data Pipeline executa um pipeline, ele compila os componentes do pipeline para criar um conjunto de instâncias acionáveis. Cada instância contém todas as informações para execução de uma tarefa específica. O conjunto completo de instâncias é a lista de tarefas do pipeline. AWS Data Pipeline distribui as instâncias para os executores de tarefas processarem.
- **Attempts** – Para fornecer um gerenciamento de dados eficiente, o AWS Data Pipeline tenta executar novamente uma operação com falha. Ele continua fazendo as tentativas até que a tarefa atinja o número máximo de tentativas permitidas. Os objetos de tentativa acompanham as tentativas, os resultados e as falhas, se aplicável. Essencialmente, é a instância com um contador. AWS Data Pipeline executa novas tentativas usando os mesmos recursos das tentativas anteriores, como clusters do Amazon EMR e instâncias EC2.

Note

Repetir tarefas com falhas é parte importante de uma estratégia de tolerância a falhas, e as definições de do AWS Data Pipeline fornecem condições e limites para controlar as tentativas. No entanto, muitas tentativas podem atrasar a detecção de uma falha irreversível, pois o AWS Data Pipeline não relata a falha até que todas as tentativas especificadas tenham se esgotado. Novas tentativas podem incorrer em cobranças adicionais se estiverem sendo executadas em recursos da AWS. Como resultado, considere cuidadosamente quando é apropriado exceder as configurações AWS Data Pipeline padrão que você usa para controlar novas tentativas e configurações relacionadas.

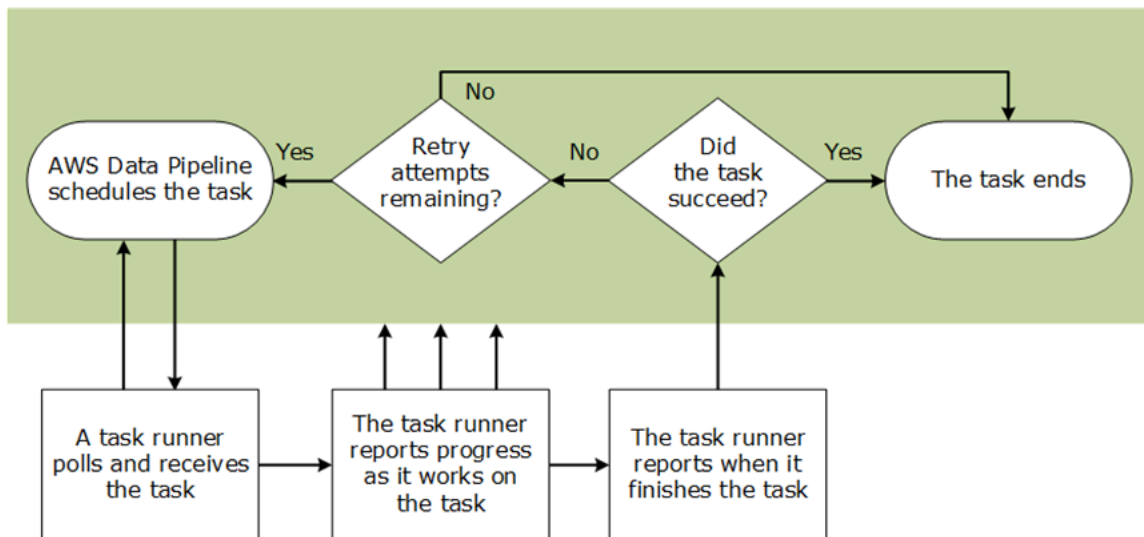


Executores de tarefas

Um executor de tarefas é um aplicativo que pesquisa tarefas e, AWS Data Pipeline em seguida, executa essas tarefas.

O Task Runner é uma implementação padrão de um executor de tarefas fornecido pelo AWS Data Pipeline. Quando o Task Runner é instalado e configurado, ele pesquisa AWS Data Pipeline as tarefas associadas aos pipelines que você ativou. Quando uma tarefa é atribuída ao Task Runner, ele a executa e informa seu status para o AWS Data Pipeline.

O diagrama a seguir ilustra como AWS Data Pipeline um executor de tarefas interage para processar uma tarefa agendada. Uma tarefa é uma unidade de trabalho discreta que o AWS Data Pipeline serviço compartilha com um executor de tarefas. Ela se difere de um pipeline, que é uma definição geral de atividades e recursos que geralmente produzem várias tarefas.



Existem duas maneiras de usar o Task Runner para processar seu pipeline:

- AWS Data Pipeline instala o Task Runner para você em recursos que são lançados e gerenciados pelo serviço AWS Data Pipeline web.
- É possível instalar o Task Runner em um recurso computacional que você gerencia, como uma instância do EC2 prolongada ou um servidor on-premises.

Para obter mais informações sobre como trabalhar com o Task Runner, consulte [Trabalhar com o Task Runner](#).

Nós de dados

Em AWS Data Pipeline, um nó de dados define a localização e o tipo de dados que uma atividade de pipeline usa como entrada ou saída. AWS Data Pipeline suporta os seguintes tipos de nós de dados:

[ynameDBDataNode D](#)

Uma tabela do DynamoDB que contém dados para utilização do [HiveActivity](#) ou [EmrActivity](#).

[SqlDataNode](#)

Uma tabela do SQL e uma consulta de banco de dados que representa os dados a serem usados por uma atividade de pipeline.

Note

Anteriormente, `MySqlDataNode` foi usado. Use `SqlDataNode` em vez disso.

[RedshiftDataNode](#)

Uma tabela do Amazon Redshift que contém dados para utilização do [RedshiftCopyActivity](#).

[S3 DataNode](#)

Um local do Amazon S3 que contém um ou mais arquivos a serem usados por uma atividade de pipeline.

Bancos de dados

AWS Data Pipeline suporta os seguintes tipos de bancos de dados:

[JdbcDatabase](#)

Um banco de dados JDBC.

[RdsDatabase](#)

Um bancos de dados do Amazon RDS.

[RedshiftDatabase](#)

Um banco de dados do Amazon Redshift.

Atividades

Em AWS Data Pipeline, uma atividade é um componente do pipeline que define o trabalho a ser executado. AWS Data Pipeline fornece várias atividades pré-empacotadas que acomodam cenários comuns, como mover dados de um local para outro, executar consultas do Hive e assim por diante. As atividades são extensíveis. Assim, você pode executar seus próprios scripts personalizados para oferecer suporte a infinitas combinações.

AWS Data Pipeline suporta os seguintes tipos de atividades:

[CopyActivity](#)

Copia dados de um local para outro.

[EmrActivity](#)

Executa o cluster do Amazon EMR.

[HiveActivity](#)

Executa uma consulta do Hive em um cluster do Amazon EMR.

[HiveCopyActivity](#)

Executa uma consulta do Hive em um cluster do Amazon EMR com suporte para filtragem avançada de dados, além de suporte a [S3 DataNode](#) e a [ynamoDBDataNodo D](#).

[PigActivity](#)

Executa um script do Pig em um cluster do Amazon EMR.

[RedshiftCopyActivity](#)

Copia dados entre as tabelas do Amazon Redshift.

[ShellCommandActivity](#)

Executa um comando shell UNIX/Linux personalizado como uma atividade.

[SqlActivity](#)

Executa uma consulta SQL em um banco de dados.

Algumas atividades contam com suporte especial para preparação de dados e tabelas de banco de dados. Para ter mais informações, consulte [Preparar dados e tabelas com atividades de pipeline](#).

Precondições

Em AWS Data Pipeline, uma pré-condição é um componente do pipeline que contém declarações condicionais que devem ser verdadeiras para que uma atividade possa ser executada. Por exemplo, uma condição prévia pode verificar se os dados de origem estão presentes antes que uma atividade do pipeline tente copiá-los. AWS Data Pipeline fornece várias pré-condições predefinidas que acomodam cenários comuns, como a existência de uma tabela de banco de dados, a presença de uma chave do Amazon S3 e assim por diante. No entanto, as precondições são extensíveis e permitem que você execute seus próprios scripts personalizados para oferecer suporte a combinações infinitas.

Existem dois tipos de condições: as gerenciadas pelo sistema e as gerenciadas pelo usuário. As pré-condições gerenciadas pelo sistema são executadas pelo serviço AWS Data Pipeline web em seu nome e não exigem um recurso computacional. As condições gerenciadas pelo usuário são executadas apenas no recurso computacional que você especifica por meio do campo `runsOn` ou `workerGroup`. O recurso `workerGroup` é derivado da atividade que usa a condição.

Condições gerenciadas pelo sistema

[DynamoDBData Existe](#)

Verifica se os dados existem em uma tabela específica do DynamoDB.

[DynamoDBTable Existe](#)

Verifica se uma tabela do DynamoDB existe.

[S3 KeyExists](#)

Verifica se uma chave do Amazon S3 existe.

[S3 PrefixNotEmpty](#)

Verifica se um prefixo do Amazon S3 está vazio.

Condições gerenciadas pelo usuário

[Existe](#)

Verifica se um nó de dados existe.

[ShellCommandPrecondition](#)

Executa um comando shell do Unix/Linux como uma condição.

Recursos

Em AWS Data Pipeline, um recurso é o recurso computacional que executa o trabalho que uma atividade de pipeline especifica. AWS Data Pipeline suporta os seguintes tipos de recursos:

[Ec2Resource](#)

Uma instância do EC2 que executa o trabalho definido por uma atividade de pipeline.

[EmrCluster](#)

Um cluster do Amazon EMR que executa o trabalho definido por uma atividade de pipeline, como [EmrActivity](#).

Os recursos podem ser executados na mesma região do seu conjunto de dados de trabalho, mesmo que ela seja diferente da região do AWS Data Pipeline. Para ter mais informações, consulte [Usar um pipeline com recursos em várias regiões](#).

Limites de recurso

AWS Data Pipeline é dimensionado para acomodar um grande número de tarefas simultâneas e você pode configurá-lo para criar automaticamente os recursos necessários para lidar com grandes cargas de trabalho. Esses recursos criados automaticamente são controlados por você e contam para os limites de recursos da sua conta da AWS. Por exemplo, se você configurar AWS Data Pipeline para criar automaticamente um cluster Amazon EMR de 20 nós para processar dados e sua conta da AWS tiver um limite de instância do EC2 definido como 20, você poderá inadvertidamente esgotar seus recursos de preenchimento disponíveis. Por isso, considere essas restrições de recursos no seu projeto ou aumente os limites da sua conta. Para obter mais informações sobre limites de serviço, consulte [Limites de serviço da AWS](#) na Referência geral da AWS.

Note

O limite é de uma instância por objeto de componente `Ec2Resource`.

Plataformas com suporte

Os pipelines podem iniciar seus recursos nas seguintes plataformas:

EC2-Classic

Seus recursos são executados em uma única rede simples que você compartilha com outros clientes.

EC2-VPC

Seus recursos são executados em uma nuvem privada virtual (VPC) que é isolada logicamente para sua conta da AWS.

Sua conta da AWS pode iniciar recursos em ambas as plataformas ou somente na plataforma EC2-VPC, dependendo da região. Para obter mais informações, consulte [Plataformas suportadas](#) no Guia do usuário do Amazon EC2.

Se a sua conta da AWS oferecer suporte somente à EC2-VPC, criaremos uma VPC padrão para você em cada região da AWS. Por padrão, iniciamos seus recursos em uma sub-rede padrão da sua VPC padrão. Se preferir, você pode criar uma VPC não padrão e especificar uma das suas sub-redes ao configurar seus recursos. Assim, iniciaremos seus recursos na sub-rede especificada da VPC não padrão.

Ao iniciar uma instância em uma VPC, você precisa especificar um security group criado especificamente para essa VPC. Não é possível especificar um security group criado para o EC2-Classic ao executar uma instância em uma VPC. Além disso, é necessário usar o ID do security group (e não o nome dele) para identificá-lo em uma VPC.

Instâncias spot do Amazon EC2 com clusters do Amazon EMR e o AWS Data Pipeline

Os pipelines podem usar as instâncias spot do Amazon EC2 para os nós de tarefa nos seus recursos de cluster do Amazon EMR. Por padrão, os pipelines usam instâncias sob demanda. As instâncias spot permitem que você use instâncias excedentes do EC2 e execute-as. O modelo de definição de preço da instância spot complementa os modelos de instâncias reservadas e sob demanda fornecendo potencialmente a opção mais econômica para obter capacidade computacional, dependendo do seu aplicativo. Para obter mais informações, consulte a página do produto [Instâncias spot do Amazon EC2](#).

Quando você usa Instâncias Spot, AWS Data Pipeline envia o preço máximo de sua Instância Spot para o Amazon EMR quando seu cluster é lançado. Ele aloca automaticamente o trabalho do cluster ao número de nós de tarefas da Instância Spot que você define usando o `taskInstanceCount` campo. AWS Data Pipeline limita as instâncias spot dos nós de tarefas para garantir que os nós principais sob demanda estejam disponíveis para executar seu pipeline.

Você pode editar uma instância de recurso de pipeline com falha ou concluída para adicionar instâncias spot. Quando o pipeline reiniciar o cluster, ele usará instâncias spot para os nós de tarefa.

Considerações sobre as instâncias spot

Quando você usa instâncias spot com AWS Data Pipeline, as seguintes considerações se aplicam:

- Suas instâncias spot poderão ser encerradas quando o preço de instância spot for maior do que o preço máximo da instância, ou por motivos de capacidade do Amazon EC2. No entanto, você não perde seus dados porque AWS Data Pipeline emprega clusters com nós principais que são sempre instâncias sob demanda e não estão sujeitos à rescisão.
- As instâncias spot podem levar mais tempo para ser iniciadas, pois elas atendem à capacidade de forma assíncrona. Portanto, um pipeline de instância spot pode ser executado mais lentamente do que um pipeline de Instância sob demanda equivalente.
- Seu cluster poderá não ser executado se você não receber suas instâncias spot, por exemplo, quando o preço máximo é muito baixo.

Ações

AWS Data Pipeline ações são etapas que um componente do pipeline executa quando certos eventos ocorrem, como sucesso, falha ou atividades atrasadas. O campo de evento de uma atividade refere-se a uma ação, como uma referência a `snsAlarm` no campo `onLateAction` de `EmrActivity`.

AWS Data Pipeline depende das notificações do Amazon SNS como a principal forma de indicar o status dos pipelines e seus componentes de forma autônoma. Para obter mais informações, consulte [Amazon SNS](#). Além das notificações do SNS, você pode usar o AWS Data Pipeline console e a CLI para obter informações sobre o status do pipeline.

AWS Data Pipeline suporta as seguintes ações:

[SnsAlarm](#)

Uma ação que envia uma notificação do SNS para um tópico com base nos eventos `onSuccess`, `OnFail` e `onLateAction`.

[Encerrar](#)

Uma ação que aciona o cancelamento de atividades, recursos ou nós de dados pendentes ou não concluídos. Não é possível encerrar ações que incluem `onSuccess`, `OnFail` ou `onLateAction`.

Monitoramento proativo de pipelines

A melhor maneira de detectar problemas é monitorar seus pipelines de forma proativa desde o início. Você pode configurar os componentes do pipeline para informá-lo sobre determinadas situações ou eventos, como quando um componente do pipeline falha ou não começa na hora de início programada. AWS Data Pipeline facilita a configuração de notificações fornecendo campos de eventos em componentes do pipeline que você pode associar às notificações do Amazon SNS, como `onSuccessOnFail`, e `onLateAction`

Configurando para AWS Data Pipeline

Antes de usar AWS Data Pipeline pela primeira vez, conclua as tarefas a seguir.

Tarefas

- [Inscreva-se para AWS](#)
- [Crie funções do IAM AWS Data Pipeline e recursos de pipeline](#)
- [Permita que as entidades principais do IAM \(usuários e grupos\) realizem as ações necessárias](#)
- [Conceder acesso programático](#)

Depois de concluir essas tarefas, você pode começar a usar AWS Data Pipeline. Para ver um tutorial básico, consulte [Conceitos básicos do AWS Data Pipeline](#).

Inscreva-se para AWS

Quando você se inscreve na Amazon Web Services (AWS), sua conta da AWS é automaticamente cadastrada em todos os serviços na AWS, inclusive AWS Data Pipeline. Você será cobrado apenas pelos serviços que usar. Para obter mais informações sobre taxas AWS Data Pipeline de uso, consulte [AWS Data Pipeline](#).

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 AWS é 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 Console](#) como 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 .

Crie funções do IAM AWS Data Pipeline e recursos de pipeline

AWS Data Pipeline requer funções do IAM que determinam as permissões para realizar ações e acessar AWS recursos. A função de pipeline determina as permissões que AWS Data Pipeline tem, e uma função de recurso determina as permissões que os aplicativos executados em recursos de pipeline, como instâncias do EC2, têm. Você deve especificar essas funções ao criar um pipeline. Mesmo que você não especifique uma função personalizada e use as funções padrão `DataPipelineDefaultRole` e `DataPipelineDefaultResourceRole`, você deve criar primeiro as funções e anexar as políticas de permissões. Para ter mais informações, consulte [Funções do IAM para o AWS Data Pipeline](#).

Permita que as entidades principais do IAM (usuários e grupos) realizem as ações necessárias

Para trabalhar com um pipeline, uma entidade principal do IAM (um usuário ou grupo) em sua conta deve ter permissão para realizar as [AWS Data Pipeline ações](#) necessárias, assim como as ações para outros serviços, conforme definido pelo seu pipeline.

Para simplificar as permissões, a política `AWSDatapipeline_FullAccessgerenciada` está disponível para você anexar aos diretores do IAM. Essa política gerenciada permite que o diretor execute todas as ações que um usuário exige e a `iam:PassRole` ação nas funções padrão usadas AWS Data Pipeline quando uma função personalizada não é especificada.

É altamente recomendável que você avalie cuidadosamente essa política gerenciada e restrinja as permissões somente àquelas que seus usuários precisam. Se necessário, use essa política como ponto de partida e, em seguida, remova as permissões para criar uma política de permissões em linha mais restritiva que você possa anexar às entidades principais do IAM. Para obter mais informações e exemplos de políticas de permissões, consulte [Exemplo de políticas para o AWS Data Pipeline](#).

Uma declaração de política semelhante ao exemplo a seguir deve ser incluída em uma política anexada a qualquer entidade principal do IAM que usa o pipeline. Essa declaração permite que a entidade principal do IAM execute a ação de `PassRole` nas funções usadas pelo pipeline. Se você não usar funções padrão, substitua *MyPipelineRole* e *MyResourceRole* pelas funções personalizadas que você criar.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "iam:PassRole",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::*:role/MyPipelineRole",
        "arn:aws:iam::*:role/MyResourceRole"
      ]
    }
  ]
}
```

O procedimento a seguir demonstra como criar um grupo do IAM, anexar a política `AWSDDataPipeline_FullAccess` gerenciada ao grupo e, em seguida, adicionar usuários ao grupo. Você pode usar esse procedimento para qualquer política em linha.

Para criar um grupo de usuários **DataPipelineDevelopers** e anexar a `AWSDDataPipeline_FullAccess` política

1. Abra o console IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Grupos, Criar novo grupo.
3. Insira um Nome do grupo, por exemplo, **DataPipelineDevelopers**, e selecione Próxima etapa.
4. Insira **AWSDDataPipeline_FullAccess** para Filtro e, em seguida, selecione-o na lista.

5. Selecione Next Step (Próxima etapa) e, em seguida, Create Group (Criar grupo).
6. Adicione usuários ao grupo:
 - a. Selecione o grupo que você criou na lista de grupos.
 - b. Escolha Group Actions (Ações de grupo) e Add Users to Group (Adicionar usuários ao grupo).
 - c. Selecione os usuários que você deseja adicionar a partir da lista e, em seguida, selecione Adicionar usuários ao grupo.

Conceder acesso programático

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do AWS Management Console. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

| Qual usuário precisa de acesso programático? | Para | Por |
|---|--|--|
| Identificação da força de trabalho (Usuários gerenciados no Centro de Identidade do IAM) | Use credenciais temporárias para assinar solicitações programáticas para AWS SDKs ou APIs. AWS CLI AWS | Siga as instruções da interface que deseja utilizar. <ul style="list-style-type: none"> • Para o AWS CLI, consulte Configurando o AWS CLI para uso AWS IAM Identity Center no Guia do AWS Command Line Interface usuário. • Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de AWS SDKs e ferramentas. |

| Qual usuário precisa de acesso programático? | Para | Por |
|--|--|---|
| IAM | Use credenciais temporárias para assinar solicitações programáticas para AWS SDKs ou APIs. AWS CLI AWS | Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM. |
| IAM | (Não recomendado) Use credenciais de longo prazo para assinar solicitações programáticas para AWS SDKs AWS CLI ou APIs. AWS | Siga as instruções da interface que deseja utilizar. <ul style="list-style-type: none">• Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário.• Para AWS SDKs e ferramentas, consulte Autenticar usando credenciais de longo prazo no Guia de referência de AWS SDKs e ferramentas.• Para AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM. |

Conceitos básicos do AWS Data Pipeline

O AWS Data Pipeline ajuda você a sequenciar, programar, executar e gerenciar cargas de trabalho de processamento de dados recorrentes de forma confiável e econômica. Este serviço facilita o desenvolvimento de atividades de extração, transformação e carregamento (ETL) usando dados estruturados e não estruturados, tanto no local como na nuvem, com base na sua lógica de negócios.

Para usar o AWS Data Pipeline, basta criar uma definição de pipeline que especifique a lógica de negócios do processamento dos seus dados. Uma definição típica de pipeline consiste em [atividades](#) que definem o trabalho a ser realizado e os [nós de dados](#) que definem o local e o tipo de dados de entrada e saída.

Neste tutorial, você executará um script de comando shell que conta o número de solicitações GET nos logs do servidor web Apache. Este pipeline é executado a cada 15 minutos por uma hora e grava a saída no Amazon S3 em todas as iterações.

Pré-requisitos

Antes de começar, conclua as tarefas em [Configurando para AWS Data Pipeline](#).

Objetos de pipeline

O pipeline usa os seguintes objetos:

[ShellCommandActivity](#)

Lê o arquivo de log de entrada e conta o número de erros.

[S3 DataNode](#) (entrada)

O bucket do S3 que contém o arquivo de log de entrada.

[S3 DataNode](#) (saída)

O bucket do S3 para saída.

[Ec2Resource](#)

O recurso de computação que o AWS Data Pipeline usa para executar a atividade.

Se você tiver uma grande quantidade de dados do arquivo de log, poderá configurar seu pipeline para usar um cluster do EMR para processar os arquivos em vez de uma instância do EC2.

Schedule

Define que a atividade é realizada a cada 15 minutos e dura uma hora.

Tarefas

- [Criar o pipeline](#)
- [Monitorar o pipeline em execução](#)
- [Visualizar a saída](#)
- [Excluir o pipeline](#)

Criar o pipeline

A maneira mais rápida de começar a usar o AWS Data Pipeline é por meio de uma definição de pipeline chamada de modelo.

Para criar o pipeline

1. Abra o console do AWS Data Pipeline em <https://console.aws.amazon.com/datapipeline/>.
2. Na barra de navegação, selecione uma região. Selecione qualquer região que estiver disponível para você, independentemente do seu local. Muitos recursos da AWS são específicos de uma região, mas o AWS Data Pipeline permite que você use os recursos de regiões diferentes da região do pipeline.
3. A primeira tela que você vê dependerá de você ter criado ou não um pipeline na região atual.
 - a. Se ainda não tiver criado um pipeline nessa região, o console exibe uma tela introdutória. Selecione Get started now.
 - b. Se você já criou um pipeline nessa região, o console exibirá uma página que lista seus pipelines para a região. Escolha Create new pipeline (Criar um novo pipeline).
4. Em Nome, insira um nome para seu pipeline.
5. (Opcional) Em Descrição, insira uma descrição para seu pipeline.
6. Em Source, selecione Build using a template e, em seguida, o modelo Getting Started using ShellCommandActivity.
7. Na seção Parameters, que abriu quando você selecionou o modelo, deixe S3 input folder e Shell command to run com seus respectivos valores padrão. Clique no ícone de pasta ao lado de S3 output folder, selecione um dos seus buckets ou pastas e, em seguida, clique em Select.

8. Em Schedule, deixe os valores padrão. Quando você ativa o pipeline, ele é iniciado e continua sendo executado a cada 15 minutos durante uma hora.

Se preferir, você pode selecionar Run once on pipeline activation.

9. Em Configuração do pipeline, deixe o registro de log ativado. Escolha o ícone da pasta na localização do S3 para registros, selecione um dos seus buckets ou pastas e, em seguida, escolha Selecionar.

Se preferir, você poderá desabilitar o registro de log.

10. Em Segurança/aceso, mantenha a seleção perfil do IAM como Padrão.
11. Clique em Activate.

Se preferir, você pode selecionar Editar no Architect para modificar esse pipeline. Por exemplo, você pode adicionar pré-condições.

Monitorar o pipeline em execução

Após ativar o pipeline, você será levado à página Execution details na qual poderá monitorar o progresso do pipeline.

Para monitorar o progresso do seu pipeline

1. Clique em Update ou pressione F5 para atualizar o status exibido.

Tip

Se não houver execuções listadas, certifique-se que as opções Start (in UTC) e End (in UTC) abrangem o início e o término programado do pipeline. Em seguida, clique em Update.

2. Quando o status de cada objeto no pipeline for FINISHED, o pipeline concluiu com êxito as tarefas programadas.
3. Se o pipeline não for concluído com êxito, verifique se há algum problema nas configurações do pipeline. Para obter mais informações sobre a solução de problemas de execuções de instâncias com falha ou incompletas do pipeline, consulte [Resolver problemas comuns](#).

Visualizar a saída

Abra o console do Amazon S3 e navegue até seu bucket. Se você executou seu pipeline a cada 15 minutos durante uma hora, verá quatro subpastas com os horários registrados. Cada subpasta contém a saída em um arquivo chamado `output.txt`. Como executamos o script no mesmo arquivo de entrada todas as vezes, os arquivos de saída serão idênticos.

Excluir o pipeline

Para parar de incorrer em cobranças, exclua seu pipeline. A exclusão do pipeline exclui a definição do pipeline e todos os objetos associados.

Para excluir seu pipeline

1. Na página Listar Pipelines, selecione o pipeline.
2. Clique em Ações e selecione Excluir.
3. Quando a confirmação for solicitada, escolha Delete (Excluir).

Se você já concluiu este tutorial, exclua as pastas de saída do seu bucket do Amazon S3.

Trabalhar com pipelines

Você pode administrar, criar e modificar pipelines usando a interface de linha de comando (CLI) ou o SDK do AWS. As seções a seguir apresentam os conceitos fundamentais do AWS Data Pipeline e mostram como trabalhar com pipelines.

Important

Antes de começar, consulte [Configurando para AWS Data Pipeline](#).

Índice

- [Criar um pipeline](#)
- [Visualizar os pipelines](#)
- [Editar o pipeline](#)
- [Clonar o pipeline](#)
- [Marcar o pipeline](#)
- [Desativar o pipeline](#)
- [Excluir o pipeline](#)
- [Preparar dados e tabelas com atividades de pipeline](#)
- [Usar um pipeline com recursos em várias regiões](#)
- [Falhas e novas execuções em cascata](#)
- [Sintaxe do arquivo de definição do pipeline](#)
- [Trabalhar com a API](#)

Criar um pipeline

O AWS Data Pipeline oferece várias maneiras de criar pipelines:

- Use a AWS Command Line Interface (CLI) com um modelo fornecido para a sua comodidade. Para obter mais informações, consulte [Crie pipelines a partir de modelos de Data Pipeline usando a CLI](#).

- Use a AWS Command Line Interface (CLI) com um arquivo de definição de pipeline em formato JSON.
- Use um AWS SDK com uma API específica do idioma. Para obter mais informações, consulte [Trabalhar com a API](#).

Crie pipelines a partir de modelos de Data Pipeline usando a CLI

O Data Pipeline oferece diversas definições pré-configuradas de pipeline, conhecidas como modelos. Os modelos podem ser usados para começar a trabalhar com o AWS Data Pipeline rapidamente. Esses modelos estão disponíveis em um bucket público no local do Amazon S3: `s3://datapipeline-us-east-1/templates/`. Esses modelos predefinidos são criados para obter casos de uso específicos e podem ser usados para criar pipelines. Você pode usar `aws s3 ls --recursive "s3://datapipeline-us-east-1/templates/"` para listar todos os modelos disponíveis.

Crie um pipeline a partir de um modelo usando a CLI

Suponha que você queira criar um pipeline que exporte uma tabela do DynamoDB para o Amazon S3. O modelo a ser usado neste caso pode ser encontrado em: `s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json`.

Para baixar o modelo em JSON e criar um pipeline usando a CLI

1. Faça o download do modelo usando a CLI do `aws s3 cp` ou `cURL`. Por exemplo:

```
aws s3 cp "s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json" <destination directory>
```

2. Faça as alterações no modelo baixado conforme necessário. Por exemplo, para usar a versão mais recente do EMR, altere o campo `releaseLabel` no objeto `EmrClusterForBackup`, altere os tipos de instância principal e central e altere os valores padrão dos parâmetros no modelo.
3. Criar um pipeline usando a CLI da `create-pipeline`. Por exemplo:

```
aws datapipeline create-pipeline --name my-ddb-backup-pipeline --unique-id my-ddb-backup-pipeline --region ap-northeast-1
```

4. Anote a ID do pipeline criado.

5. Use `put-pipeline-definition` para fazer o upload da definição. Forneça valores dos parâmetros cujos valores padrão você deseja substituir usando a opção `--parameter-values`.

Para obter mais informações sobre os modelos, consulte [Escolher um modelo](#).

Escolher um modelo

Os modelos a seguir estão disponíveis para download no bucket do Amazon S3: `s3://datapipeline-us-east-1/templates/`.

Modelos

- [Conceitos básicos de uso do ShellCommandActivity](#)
- [Execute os comandos da CLI da AWS.](#)
- [Exportar tabela do DynamoDB para o S3](#)
- [Importar dados de backup do DynamoDB a partir do S3](#)
- [Executar trabalho em um cluster do Amazon EMR](#)
- [Cópia completa da tabela MySQL do Amazon RDS para o Amazon S3](#)
- [Cópia incremental da tabela MySQL do Amazon RDS para o Amazon S3](#)
- [Carregar dados do S3 para uma tabela MySQL do Amazon RDS](#)
- [Cópia total da tabela MySQL do Amazon RDS para o Amazon Redshift](#)
- [Cópia incremental da tabela MySQL do Amazon RDS para o Amazon Redshift](#)
- [Carregar dados do Amazon S3 para o Amazon Redshift](#)

Conceitos básicos de uso do ShellCommandActivity

O modelo `Getting Started using ShellCommandActivity` executa um script de comando shell para contar o número de solicitações GET em um arquivo de log. A saída é gravada em um local do Amazon S3 com marca de tempo em todas as execuções programadas do pipeline.

O modelo usa os seguintes objetos de pipeline:

- `ShellCommandActivity`
- `S3InputNode`

- S3OutputNode
- Ec2Resource

Execute os comandos da CLI da AWS.

Este modelo executa um comando da AWS CLI especificado pelo usuário em intervalos programados.

Exportar tabela do DynamoDB para o S3

O modelo Export DynamoDB table to S3 programa um cluster do Amazon EMR a fim de exportar dados de uma tabela do DynamoDB para um bucket do Amazon S3. Esse modelo usa um cluster do Amazon EMR, que é dimensionado proporcionalmente ao valor do throughput disponível para a tabela do DynamoDB. Embora possa aumentar IOPs em uma tabela, você pode incorrer em custos adicionais ao importar e exportar. Anteriormente, a exportação usava HiveActivity, mas agora usa MapReduce nativo.

O modelo usa os seguintes objetos de pipeline:

- [EmrActivity](#)
- [EmrCluster](#)
- [ynamoDBDataNode D](#)
- [S3 DataNode](#)

Importar dados de backup do DynamoDB a partir do S3

O modelo Import DynamoDB backup data from S3 programa um cluster do Amazon EMR para de carregar um backup do DynamoDB criado anteriormente no Amazon S3 para uma tabela do DynamoDB. Os itens existentes na tabela do DynamoDB são atualizados com os dados de backup, e os novos itens são adicionados à tabela. Esse modelo usa um cluster do Amazon EMR, que é dimensionado proporcionalmente ao valor do throughput disponível para a tabela do DynamoDB. Embora possa aumentar IOPs em uma tabela, você pode incorrer em custos adicionais ao importar e exportar. Anteriormente, a importação usava HiveActivity, mas agora usa MapReduce nativo.

O modelo usa os seguintes objetos de pipeline:

- [EmrActivity](#)

- [EmrCluster](#)
- [ynamoDBDataNodo D](#)
- [S3 DataNode](#)
- [S3 PrefixNotEmpty](#)

Executar trabalho em um cluster do Amazon EMR

O modelo Run Job on an Elastic MapReduce Cluster inicia um cluster do Amazon EMR com base nos parâmetros fornecidos e começa a executar etapas com base na programação especificada. Assim que o trabalho for concluído, o cluster do EMR será encerrado. As ações de bootstrap opcionais podem ser especificadas para instalar um software adicional ou alterar a configuração do aplicativo no cluster.

O modelo usa os seguintes objetos de pipeline:

- [EmrActivity](#)
- [EmrCluster](#)

Cópia completa da tabela MySQL do Amazon RDS para o Amazon S3

O modelo Full Copy of RDS MySQL Table to S3 copia uma tabela inteira MySQL do Amazon RDS e armazena a saída em um local do Amazon S3. A saída é armazenada como um arquivo CSV em uma subpasta com marca de tempo no local especificado do Amazon S3.

O modelo usa os seguintes objetos de pipeline:

- [CopyActivity](#)
- [Ec2Resource](#)
- [SqlDataNode](#)
- [S3 DataNode](#)

Cópia incremental da tabela MySQL do Amazon RDS para o Amazon S3

O modelo Incremental Copy of RDS MySQL Table to S3 cria uma cópia incremental dos dados de uma tabela MySQL do Amazon RDS e armazena a saída em uma local do Amazon S3. A tabela MySQL do Amazon RDS deve ter uma coluna Última modificação.

Este modelo copia alterações feitas na tabela entre intervalos programados começando na hora inicial programada. Como o tipo de programação é de série temporal, se uma cópia tiver sido programada para uma determinada hora, o AWS Data Pipeline copiará as linhas da tabela que têm um marca de tempo para Última modificação que esteja na hora. As exclusões físicas feitas na tabela não são copiadas. A saída é gravada em uma subpasta com marca de tempo no local do Amazon S3 em todas as execuções programadas.

O modelo usa os seguintes objetos de pipeline:

- [CopyActivity](#)
- [Ec2Resource](#)
- [SqlDataNode](#)
- [S3 DataNode](#)

Carregar dados do S3 para uma tabela MySQL do Amazon RDS

O modelo Load S3 Data into RDS MySQL Table programa uma instância do Amazon EC2 a fim de copiar o arquivo CSV do caminho do arquivo do Amazon S3 especificado abaixo para uma tabela MYSQL do Amazon RDS. O arquivo CSV não deve ter uma linha de cabeçalho. O modelo atualiza entradas existentes na tabela MySQL do Amazon RDS com aquelas nos dados do Amazon S3 e adiciona novas entradas dos dados do Amazon S3 à tabela MySQL do Amazon RDS. Você pode carregar os dados em uma tabela existente ou fornecer uma consulta SQL para criar uma nova tabela.

O modelo usa os seguintes objetos de pipeline:

- [CopyActivity](#)
- [Ec2Resource](#)
- [SqlDataNode](#)
- [S3 DataNode](#)

Modelos do Amazon RDS para o Amazon Redshift

Estes dois modelos copiam tabelas MySQL do Amazon RDS para o Amazon Redshift usando um script de conversão, que cria uma tabela do Amazon Redshift usando o esquema da tabela de origem com as seguintes ressalvas:

- Se uma chave de distribuição não for especificada, a primeira chave primária da tabela do Amazon RDS será definida como a chave de distribuição.
- Você não pode ignorar uma coluna presente na tabela MySQL do Amazon RDS ao fazer uma cópia para o Amazon Redshift.
- (Opcional) Você pode fornecer um MySQL do Amazon RDS para o mapeamento dos tipos de dados da coluna do Amazon Redshift como um dos parâmetros no modelo. Se isso for especificado, o script o usará para criar a tabela do Amazon Redshift.

Se o modo de inserção do Amazon Redshift de `Overwrite_Existing` estiver sendo usado:

- Se uma chave de distribuição não for fornecida, será usada uma chave primária na tabela MySQL do Amazon RDS.
- Se houver chaves primárias compostas na tabela, a primeira será usada como a chave de distribuição, se a chave de distribuição não for fornecida. Somente a primeira chave composta é definida como a chave primária na tabela do Amazon Redshift.
- Se uma chave de distribuição não for fornecida e não houver chave primária na tabela MySQL do Amazon RDS, ocorrerá falha na operação de cópia.

Para obter mais informações sobre o Amazon Redshift, consulte os seguintes tópicos:

- [Amazon Redshift cluster](#) (Cluster do Amazon Redshift)
- [COPY](#) do Amazon Redshift
- [Estilos de distribuição](#) e [exemplos](#) DISTKEY
- [Chaves de classificação](#)

A seguinte tabela descreve como o script converte os tipos de dados:

Conversões de tipo de dados entre MySQL e Amazon Redshift

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|----------------------------|----------------------------------|--|
| TINYINT, TINYINT (size) | SMALLINT | MySQL: de -128 a 127. O número máximo de dígitos pode ser especificado entre parênteses. |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|--|----------------------------------|--|
| | | <p>Amazon Redshift: INT2. Número inteiro de dois bytes assinado</p> |
| <p>TINYINT UNSIGNED, TINYINT (size) UNSIGNED</p> | <p>SMALLINT</p> | <p>MySQL: de 0 a 255 UNSIGNED. O número máximo de dígitos pode ser especificado entre parênteses.</p> <p>Amazon Redshift: INT2. Número inteiro de dois bytes assinado</p> |
| <p>SMALLINT, SMALLINT(size)</p> | <p>SMALLINT</p> | <p>MySQL: de -32768 a 32767 normal. O número máximo de dígitos pode ser especificado entre parênteses.</p> <p>Amazon Redshift: INT2. Número inteiro de dois bytes assinado</p> |
| <p>SMALLINT UNSIGNED, SMALLINT(size) UNSIGNED,</p> | <p>INTEGER</p> | <p>MySQL: de 0 a 65535 UNSIGNED*. O número máximo de dígitos pode ser especificado entre parênteses</p> <p>Amazon Redshift: INT4. Número inteiro de quatro bytes assinado</p> |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|--|----------------------------------|---|
| MEDIUMINT, MEDIUMINT(size) | INTEGER | MySQL: de 388608 a 8388607. O número máximo de dígitos pode ser especificado entre parênteses Amazon Redshift: INT4. Número inteiro de quatro bytes assinado |
| MEDIUMINT UNSIGNED, MEDIUMINT(size) UNSIGNED | INTEGER | MySQL: de 0 a 16777215. O número máximo de dígitos pode ser especificado entre parênteses Amazon Redshift: INT4. Número inteiro de quatro bytes assinado |
| INT, INT(size) | INTEGER | MySQL: de 147483648 a 2147483647 Amazon Redshift: INT4. Número inteiro de quatro bytes assinado |
| INT UNSIGNED, INT(size) UNSIGNED | BIGINT | MySQL: de 0 a 4294967295 Amazon Redshift: INT8. Número inteiro de oito bytes assinado |
| BIGINT BIGINT(size) | BIGINT | Amazon Redshift: INT8. Número inteiro de oito bytes assinado |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|--|----------------------------------|--|
| BIGINT UNSIGNED BIGINT(size) UNSIGNED | VARCHAR(20*4) | MySQL: de 0 a 18446744073709551615 Amazon Redshift: Sem equivalente nativo. Por isso, usando char array. |
| FLOAT FLOAT(size,d) FLOAT(size,d) UNSIGNED | REAL | O número máximo de dígitos pode ser especificado no parâmetro size. O número máximo de dígitos à direita da casa decimal é especificado no parâmetro d. Amazon Redshift: FLOAT4 |
| DOUBLE(size,d) | DOUBLE PRECISION | O número máximo de dígitos pode ser especificado no parâmetro size. O número máximo de dígitos à direita da casa decimal é especificado no parâmetro d. Amazon Redshift: FLOAT8 |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|---------------------|----------------------------------|---|
| DECIMAL(size,d) | DECIMAL(size,d) | <p>Um DOUBLE armazenado como uma string, o que possibilita uma casa decimal fixa. O número máximo de dígitos pode ser especificado no parâmetro size. O número máximo de dígitos à direita da casa decimal é especificado no parâmetro d.</p> <p>Amazon Redshift: sem equivalente nativo.</p> |
| CHAR(size) | VARCHAR(size*4) | <p>Mantém uma string de tamanho fixo, que pode conter letras, números e caracteres especiais. O tamanho fixo é especificado como o parâmetro entre parênteses. É possível armazenar até 255 caracteres.</p> <p>Lado direito preenchido com espaços.</p> <p>Amazon Redshift: o tipo de dados CHAR não dá suporte a caracteres multibyte, logo, VARCHAR é usado.</p> <p>O número máximo de bytes por caractere é 4 de acordo com RFC3629, o que limita a tabela de caracteres U+10FFFF.</p> |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|-------------------------------|----------------------------------|--|
| VARCHAR(size) | VARCHAR(size*4) | É possível armazenar até 255 caracteres. VARCHAR não dá suporte aos seguintes pontos de código UTF-8 inválidos: 0xD800 - 0xDFFF, (Sequências de bytes: ED A0 80 - ED BF BF), 0xFDD0 - 0xFDEF, 0xFFFE e 0xFFFF, (Sequências de bytes: EF B7 90 - EF B7 AF, EF BF BE, and EF BF BF) |
| TINYTEXT | VARCHAR(255*4) | Mantém uma string com um tamanho máximo de 255 caracteres |
| TEXT | VARCHAR(máximo) | Mantém uma string com um tamanho máximo de 65.535 caracteres. |
| MEDIUMTEXT | VARCHAR(máximo) | De 0 a 16.777.215 caracteres |
| LONGTEXT | VARCHAR(máximo) | De 0 a 4.294.967.295 caracteres |
| BOOLEAN BOOL TINYINT(1) | BOOLEAN | MySQL: esses tipos são sinônimos para TINYINT (1) . Um valor zero é considerado falso. Valores diferentes de zero são considerados verdadeiros. |
| BINARY[(M)] | varchar(255) | M é de 0 a 255 bytes, FIXED |
| VARBINARY(M) | VARCHAR(máximo) | 0 a 65,535 bytes |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|---------------------|----------------------------------|--|
| TINYBLOB | VARCHAR (255) | 0 a 255 bytes |
| BLOB | VARCHAR(máximo) | 0 a 65,535 bytes |
| MEDIUMBLOB | VARCHAR(máximo) | 0 a 16,777,215 bytes |
| LOB | VARCHAR(máximo) | 0 a 4,294,967,295 bytes |
| ENUM | VARCHAR(255*2) | O limite não está no tamanho da string enum literal, e sim na definição de tabela para o número de valores enum. |
| SET | VARCHAR(255*2) | Como enum. |
| DATE | DATE | (YYYY-MM-DD) De "1000-01-01" a "9999-12-31" |
| TIME | VARCHAR(10*4) | (hh:mm:ss) De "-838:59:59" a "838:59:59" |
| DATETIME | TIMESTAMP | (YYYY-MM-DD hh:mm:ss) De 1000-01-01 00:00:00" a "9999-12-31 23:59:59" |
| TIMESTAMP | TIMESTAMP | (YYYYMMDDhhmmss) De 19700101000000 a 2037+ |
| YEAR | VARCHAR(4*4) | (YYYY) 1900 – 2155 |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|---|--|---|
| Coluna SERIAL | <p>Geração de ID/Este atributo não é necessário para um data warehouse OLAP após a cópia da coluna.</p> <p>A palavra-chave SERIAL não é adicionada durante a conversão.</p> | <p>Na verdade, SERIAL é uma entidade chamada SEQUENCE. Ela existe de maneira independente no restante da tabela.</p> <p>Coluna GENERATED BY DEFAULT</p> <p>equivale a:</p> <p>Nome CREATE SEQUENCE; tabela CREATE TABLE (coluna INTEGER NOT NULL DEFAULT nextval(name));</p> |
| Coluna BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE | <p>Geração de ID/Este atributo não é necessário para um data warehouse OLAP após a cópia da coluna.</p> <p>Dessa forma, a palavra-chave SERIAL não é adicionada durante a conversão.</p> | <p>Na verdade, SERIAL é uma entidade chamada SEQUENCE. Ela existe de maneira independente no restante da tabela.</p> <p>Coluna GENERATED BY DEFAULT</p> <p>equivale a:</p> <p>Nome CREATE SEQUENCE; tabela CREATE TABLE (coluna INTEGER NOT NULL DEFAULT nextval(name));</p> |

| Tipo de dados MySQL | Tipo de dados do Amazon Redshift | Observações |
|---------------------|--|--|
| ZEROFILL | A palavra-chave ZEROFILL não é adicionada durante a conversão. | INT UNSIGNED ZEROFILL NOT NULL ZEROFILL preenche o valor exibido do campo com zeros até a exibição da largura especificada na definição da coluna. Os valores maiores que a largura de exibição não são truncados. O uso de ZEROFILL também implica UNSIGNED. |

Cópia total da tabela MySQL do Amazon RDS para o Amazon Redshift

O modelo Full copy of Amazon RDS MySQL table to Amazon Redshift copia toda a tabela MySQL do Amazon RDS para uma tabela Amazon Redshift ao preparar dados em uma pasta do Amazon S3. A pasta de preparação do Amazon S3 deve estar na mesma região que o cluster do Amazon Redshift. Uma tabela do Amazon Redshift será criada com o mesmo esquema da tabela de origem do MySQL do Amazon RDS, se ainda não existir. Forneça qualquer substituição do tipo de dados da coluna MySQL do Amazon RDS para Amazon Redshift que você gostaria de aplicar durante a criação da tabela do Amazon Redshift.

O modelo usa os seguintes objetos de pipeline:

- [CopyActivity](#)
- [RedshiftCopyActivity](#)
- [S3 DataNode](#)
- [SqlDataNode](#)
- [RedshiftDataNode](#)
- [RedshiftDatabase](#)

Cópia incremental da tabela MySQL do Amazon RDS para o Amazon Redshift

O modelo Incremental copy of Amazon RDS MySQL table to Amazon Redshift copia dados de uma tabela MySQL do Amazon RDS para uma tabela do Amazon Redshift preparando dados em uma pasta do Amazon S3.

A pasta de preparação do Amazon S3 deve estar na mesma região que o cluster do Amazon Redshift.

O AWS Data Pipeline usa um script de conversão para criar uma tabela do Amazon Redshift com o mesmo esquema da tabela de origem do MySQL do Amazon RDS, se ela ainda não existir. Você deve fornecer qualquer substituição do tipo de dados da coluna MySQL do Amazon RDS para Amazon Redshift que você gostaria de aplicar durante a criação da tabela do Amazon Redshift.

Este modelo copia alterações feitas na tabela MySQL do Amazon RDS entre intervalos programados começando na hora inicial programada. As exclusões físicas feitas na tabela MySQL do Amazon RDS não são copiadas. Você deve fornecer o nome da coluna que armazena o valor da hora da modificação mais recente.

Ao usar o modelo padrão para criar pipelines para cópias incrementais do Amazon RDS, será criada uma atividade com o nome padrão RDSToS3CopyActivity. Você pode renomeá-la.

O modelo usa os seguintes objetos de pipeline:

- [CopyActivity](#)
- [RedshiftCopyActivity](#)
- [S3 DataNode](#)
- [SqlDataNode](#)
- [RedshiftDataNode](#)
- [RedshiftDatabase](#)

Carregar dados do Amazon S3 para o Amazon Redshift

O modelo Load data from S3 into Redshift copia dados de uma tabela do Amazon S3 para uma tabela do Amazon Redshift. Você pode carregar os dados em uma tabela existente ou fornecer uma consulta SQL para criar a tabela.

Os dados são copiados com base nas opções COPY do Amazon Redshift. A tabela do Amazon Redshift deve ter o mesmo esquema que os dados no Amazon S3. Para opções COPY, consulte [COPY](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

O modelo usa os seguintes objetos de pipeline:

- [CopyActivity](#)
- [RedshiftCopyActivity](#)
- [S3 DataNode](#)
- [RedshiftDataNode](#)
- [RedshiftDatabase](#)
- [Ec2Resource](#)

Criar um pipeline usando modelos parametrizados

Você pode usar um modelo parametrizado para personalizar uma definição de pipeline. Isso permite criar uma definição de pipeline comum, mas fornecer parâmetros diferentes quando você adiciona a definição de pipeline a um novo pipeline.

Índice

- [Adicionar myVariables à definição de pipeline](#)
- [Definir objetos de parâmetro](#)
- [Definir valores de parâmetro](#)
- [Enviar a definição de pipeline](#)

Adicionar myVariables à definição de pipeline

Ao criar o arquivo de definição de pipeline, especifique variáveis usando a seguinte sintaxe: `#{myVariable}`. É necessário que a variável seja prefixada por `my`. Por exemplo, o arquivo de definição de pipeline a seguir, `pipeline-definition.json`, inclui as seguintes variáveis: `myShellCmd`, `myS3InputLoc` e `myS3OutputLoc`.

Note

Uma definição de pipeline tem um limite máximo de 50 parâmetros.

```

{
  "objects": [
    {
      "id": "ShellCommandActivityObj",
      "input": {
        "ref": "S3InputLocation"
      },
      "name": "ShellCommandActivityObj",
      "runsOn": {
        "ref": "EC2ResourceObj"
      },
      "command": "#{myShellCmd}",
      "output": {
        "ref": "S3OutputLocation"
      },
      "type": "ShellCommandActivity",
      "stage": "true"
    },
    {
      "id": "Default",
      "scheduleType": "CRON",
      "failureAndRerunMode": "CASCADE",
      "schedule": {
        "ref": "Schedule_15mins"
      },
      "name": "Default",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "id": "S3InputLocation",
      "name": "S3InputLocation",
      "directoryPath": "#{myS3InputLoc}",
      "type": "S3DataNode"
    },
    {
      "id": "S3OutputLocation",
      "name": "S3OutputLocation",
      "directoryPath": "#{myS3OutputLoc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-mm-ss')}",
      "type": "S3DataNode"
    }
  ]
}

```

```

    "id": "Schedule_15mins",
    "occurrences": "4",
    "name": "Every 15 minutes",
    "startAt": "FIRST_ACTIVATION_DATE_TIME",
    "type": "Schedule",
    "period": "15 Minutes"
  },
  {
    "terminateAfter": "20 Minutes",
    "id": "EC2ResourceObj",
    "name": "EC2ResourceObj",
    "instanceType": "t1.micro",
    "type": "Ec2Resource"
  }
]
}

```

Definir objetos de parâmetro

Você pode criar um arquivo à parte com objetos de parâmetro que determinem as variáveis na definição de pipeline. Por exemplo, o arquivo JSON a seguir, `parameters.json`, contém objetos de parâmetro para as variáveis `myShellCmd`, `myS3InputLoc` e `myS3OutputLoc` da definição do pipeline de exemplo acima.

```

{
  "parameters": [
    {
      "id": "myShellCmd",
      "description": "Shell command to run",
      "type": "String",
      "default": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* > ${OUTPUT1_STAGING_DIR}/output.txt"
    },
    {
      "id": "myS3InputLoc",
      "description": "S3 input location",
      "type": "AWS::S3::ObjectKey",
      "default": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/data"
    },
    {
      "id": "myS3OutputLoc",
      "description": "S3 output location",
      "type": "AWS::S3::ObjectKey"
    }
  ]
}

```



```

    }
  ]
}
```

Note

Você poderia adicionar esses objetos diretamente ao arquivo de definição do pipeline, em vez de usar um arquivo à parte.

A tabela a seguir descreve os atributos dos objetos de parâmetro.

Atributos de parâmetro

| Atributo | Type | Descrição |
|-----------|---|--|
| id | Segmento | O identificador exclusivo do parâmetro. Para mascarar o valor enquanto ele é digitado ou exibido, adicione um asterisco (*) como um prefixo. Por exemplo, *myVariable —. Isso também criptografa o valor antes que ele seja armazenado pelo AWS Data Pipeline. |
| descrição | Segmento | Uma descrição do parâmetro. |
| tipo | String, Integer, Double ou AWS::S3::ObjectKey | O tipo de parâmetro que define o intervalo permitido de valores de entrada e regras de validação. O padrão é String. |
| opcional | Booleano | Indica se o parâmetro é opcional ou obrigatório. O padrão é false. |

| Atributo | Type | Descrição |
|---------------|------------------|---|
| allowedValues | Lista de strings | Enumera todos os valores permitidos para o parâmetro. |
| padrão | Segmento | O valor padrão do parâmetro . Se você especificar um valor para esse parâmetro usando valores de parâmetro, ele substituirá o valor padrão. |
| isArray | Booleano | Indica se o parâmetro é uma matriz. |

Definir valores de parâmetro

Você pode criar um arquivo à parte para definir as variáveis usando valores de parâmetro. Por exemplo, o arquivo JSON a seguir, `file://values.json`, contém o valor da variável `myS3OutputLoc` da definição do pipeline de exemplo acima.

```
{
  "values":
  {
    "myS3OutputLoc": "myOutputLocation"
  }
}
```

Enviar a definição de pipeline

Ao enviar a definição de pipeline, você pode especificar parâmetros, objetos de parâmetro e valores de parâmetro. Por exemplo, você pode usar o comando da <https://docs.aws.amazon.com/cli/latest/reference/datapipeline/put-pipeline-definition.html> `put-pipeline-definition` AWS CLI da seguinte maneira:

```
$ aws datapipeline put-pipeline-definition --pipeline-id id --pipeline-definition
file://pipeline-definition.json \
--parameter-objects file://parameters.json --parameter-values-uri file://values.json
```

Note

Uma definição de pipeline tem um limite máximo de 50 parâmetros. O tamanho do arquivo para `parameter-values-uri` tem um limite máximo de 15 KB.

Visualizar os pipelines

Você pode visualizar os pipelines usando o console ou a Interface de linha de comando (CLI).

Para visualizar os pipelines usando a AWS CLI

- Use o seguinte comando [list-pipelines](#) para listar os pipelines:

```
aws datapipeline list-pipelines
```

Interpretar códigos de status do pipeline

Os níveis de status exibidos no console do AWS Data Pipeline e na CLI indicam a condição de um pipeline e seus componentes. O status do pipeline é simplesmente uma visão geral de um pipeline. Para mais informações, veja o status dos componentes individuais do pipeline.

Um pipeline terá um status SCHEDULED se estiver pronto (a validação de definição do pipeline aprovada), realizando um trabalho no momento ou tiver concluído a realização do trabalho. Um pipeline terá um status PENDING se não estiver ativado ou não conseguir realizar o trabalho (por exemplo, a validação de definição do pipeline com falha).

Um pipeline será considerado inativo se o status for PENDING, INACTIVE ou FINISHED. Os pipelines inativos incorrem em uma cobrança (para obter mais informações, consulte [Definição de preço](#)).

Códigos de status

ACTIVATING

O componente ou recurso está sendo iniciado, como uma instância do EC2.

CANCELED

O componente foi cancelado por um usuário ou AWS Data Pipeline antes de ser executado. Isso pode acontecer automaticamente quando ocorre uma falha em um componente ou recurso diferente do qual esse componente depende.

CASCADE_FAILED

O componente ou recurso foi cancelado como em resposta a uma falha em cascata de uma de suas dependências, mas o componente provavelmente não era a fonte original da falha.

DEACTIVATING

O pipeline está sendo desativado.

FAILED

O componente ou recurso encontrou um erro e parou de funcionar. Quando há falha de um componente ou recurso, isso pode causar cancelamentos e falhas em cascata para outros componentes que dependem dele.

FINISHED

O componente concluiu o trabalho atribuído.

INACTIVE

O pipeline foi desativado.

PAUSED

O componente foi pausado e, no momento, não está executando seu trabalho.

PENDING

O pipeline está pronto para ser ativado pela primeira vez.

RUNNING

O recurso está sendo executado e pronto para receber trabalho.

SCHEDULED

O recurso está programado para ser executado.

SHUTTING_DOWN

O recurso está sendo encerrado após a conclusão bem-sucedida do trabalho.

SKIPPED

O componente pulou os intervalos de execução após a ativação do pipeline usando uma marca de tempo posterior à programação atual.

TIMEDOUT

O recurso excedeu o limite de `terminateAfter` e foi interrompido pelo AWS Data Pipeline. Depois que o recurso atinge esse status, AWS Data Pipeline ignora os valores de `actionOnResourceFailure`, `retryDelay` e `retryTimeout` para esse recurso. Esse status só é aplicável aos recursos.

VALIDATING

A definição de pipeline está sendo validada pelo AWS Data Pipeline.

WAITING_FOR_RUNNER

O componente está aguardando que o operador do cliente recupere um item de trabalho. O relacionamento entre componente e operador do cliente é controlado pelos campos `runsOn` ou `workerGroup` definidos por esse componente.

WAITING_ON_DEPENDENCIES

O componente está verificando se as pré-condições padrão e configuração pelo usuário foram atendidas antes de realizar seu trabalho.

Interpretar pipeline e estado de integridade do componente

Cada pipeline e componente dentro desse pipeline retorna um status de integridade de `HEALTHY`, `ERROR`, `"-"`, `No Completed Executions` ou `No Health Information Available`. Um pipeline só terá um estado de integridade depois que um componente de pipeline tiver concluído a primeira execução ou se houver falha nas condições do componente. O status de integridade de componentes agrega ao status de integridade de um pipeline porque os estados de erro são visíveis quando você os detalhes da execução do pipeline primeiro.

Estados de integridade do pipeline

HEALTHY

O status de integridade agregado de todos os componentes é `HEALTHY`. Isso significa que pelo menos um componente deve ter sido concluído com êxito. Você pode clicar no status `HEALTHY`

para ver a instância do componente do pipeline concluído mais recentemente na página Detalhes da execução.

ERROR

Pelo menos um componente no pipeline apresenta um status de integridade ERROR. Você pode clicar no status ERROR para ver a instância do componente do pipeline com falha mais recente na página Execution Details.

No Completed Executions ou No Health Information Available.

Nenhum status de integridade foi relatado para o pipeline.

Note

Embora os componentes atualizem o status de integridade quase imediatamente, pode levar até cinco minutos para o status de integridade do pipeline ser atualizado.

Estados de integridade do componente

HEALTHY

Um componente (Activity ou DataNode) terá um status de integridade HEALTHY se tiver concluído uma execução bem-sucedida na qual tenha sido marcado com um status FINISHED ou MARK_FINISHED. Você pode clicar no nome do componente ou no status HEALTHY para ver as instâncias do componente do pipeline concluído mais recentemente na página Detalhes da execução.

ERROR

Ocorreu um erro no nível do componente ou uma das condições falhou. Os status FAILED, TIMEOUT ou CANCELED disparam esse erro. Você pode clicar no nome do componente ou no status ERROR para ver a instância do componente do pipeline com falha mais recente na página Execution Details.

No Completed Executions ou No Health Information Available

Nenhum status de integridade foi relatado para o componente.

Visualizar as definições do pipeline

Use a interface da linha de comando (CLI) para visualizar a definição do pipeline. A CLI imprime uma linha de definição de pipeline em formato JSON. Para obter informações sobre a sintaxe e o uso de arquivos de definição de pipeline, consulte [Sintaxe do arquivo de definição do pipeline](#).

Ao usar a CLI, é uma boa ideia recuperar a definição do pipeline antes de enviar modificações, porque é possível que outro usuário ou processo tenha alterado a definição do pipeline depois que você trabalhou nele mais recentemente. Fazendo download de uma cópia da definição atual e o usando como a base para as modificações, você pode ter a certeza de que está trabalhando com a definição de pipeline mais recente. Também é uma boa ideia recuperar a definição do pipeline novamente depois de modificá-lo, de maneira que você possa garantir que a atualização tenha sido bem-sucedida.

Se estiver usando a CLI, você poderá ter duas versões diferentes do pipeline. A versão `active` é o pipeline em execução no momento. A versão `latest` é uma cópia criada quando você edita um pipeline em execução. Quando você carrega o pipeline editado, ele se torna a versão `active`, e a versão `active` anterior deixa de estar disponível.

Para obter uma definição de pipeline usando a AWS CLI

Para obter a definição de pipeline completa, use o comando [get-pipeline-definition](#) a seguir. A definição de pipeline é impressa na saída padrão (stdout).

O exemplo a seguir obtém a definição do pipeline especificado.

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE
```

Para recuperar uma versão específica de um pipeline, use a opção `--version`. O exemplo a seguir recupera a versão `active` do pipeline especificado.

```
aws datapipeline get-pipeline-definition --version active --id df-00627471S0VYZEXAMPLE
```

Visualizar detalhes da instância do pipeline

Você pode monitorar o progresso do pipeline. Para obter mais informações sobre o status da instância, consulte [Interpretar detalhes de status do pipeline](#). Para obter mais informações sobre a solução de problemas de execuções de instâncias com falha ou incompletas do pipeline, consulte [Resolver problemas comuns](#).

Para monitorar o progresso de um pipeline usando a AWS CLI

Para recuperar os detalhes da instância do pipeline, como um histórico das vezes em que um pipeline foi executado, use o comando [list-runs](#). Esse comando permite filtrar a lista de execuções retornadas com base no status atual ou no intervalo de datas em que elas foram iniciadas. Filtrar os resultados é útil porque, dependendo da idade do pipeline e da programação, o histórico de execuções pode ser grande.

O exemplo a seguir recupera informações de todas as execuções.

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE
```

O exemplo a seguir recupera informações de todas as execuções concluídas.

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE --status finished
```

O exemplo a seguir recupera informações de todas as execuções iniciadas no período especificado.

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE --start-interval  
"2013-09-02", "2013-09-11"
```

Visualizar logs de pipeline

Há suporte para o registro em log no nível do pipeline na criação do pipeline especificando um local do Amazon S3 no console ou com um `pipelineLogUri` no objeto padrão em SDK/CLI. A estrutura do diretório de cada pipeline nesse URI é como a seguinte:

```
pipelineId  
  -componentName  
    -instanceId  
      -attemptId
```

Para pipeline, `df-00123456ABC7DEF8HIJK`, a estrutura do diretório é semelhante a:

```
df-00123456ABC7DEF8HIJK  
  -ActivityId_fXNzc  
    -@ActivityId_fXNzc_2014-05-01T00:00:00  
      -@ActivityId_fXNzc_2014-05-01T00:00:00_Attempt=1
```


Para `ShellCommandActivity`, logs de `stderr` e `stdout` associados a essas atividades são armazenados no diretório de cada tentativa.

Para recursos como `EmrCluster`, em que um `emrLogUri` é definido, esse valor tem precedência. Caso contrário, os recursos (inclusive logs do `TaskRunner` desses recursos) seguirão a estrutura do registro em log do pipeline acima.

Para visualizar os logs de uma determinada execução de pipeline:

1. Recupere o `ObjectId` ao chamar `query-objects` para obter a ID exata do objeto. Por exemplo:

```
aws datapipeline query-objects --pipeline-id <pipeline-id> --sphere ATTEMPT --region ap-northeast-1
```

`query-objects` é uma CLI paginada e pode retornar um token de paginação se houver mais execuções para o `pipeline-id` determinado. Você pode usar o token para passar por todas as tentativas até encontrar o objeto esperado. Por exemplo, um `ObjectID` retornado teria a seguinte aparência: `@TableBackupActivity_2023-05-020T18:05:18_Attempt=1`.

2. Usando o `ObjectID`, recupere a localização do log usando:

```
aws datapipeline describe-objects --pipeline-id <pipeline-id> --object-ids <object-id> --query "pipelineObjects[].fields[?key=='@logLocation'].stringValue"
```

Mensagem de erro de uma atividade com falha

Para receber a mensagem de erro, primeiro obtenha o `ObjectID` usando `query-objects`.

Depois de recuperar o `ObjectID` com falha, use a CLI de `describe-objects` para obter a mensagem de erro real.

```
aws datapipeline describe-objects --region ap-northeast-1 --pipeline-id <pipeline-id> --object-ids <object-id> --query "pipelineObjects[].fields[?key=='errorMessage'].stringValue"
```

Cancelar, executar novamente ou marcar como concluído um objeto

Use a CLI de `set-status` para cancelar um objeto em execução, executar novamente um objeto com falha ou marcar um objeto em execução como `Concluído`.

Primeiro, obtenha o ID do objeto usando a CLI de `query-objects`. Por exemplo:

```
aws datapipeline query-objects --pipeline-id <pipeline-id> --sphere INSTANCE --region ap-northeast-1
```

Use a CLI de `set-status` para alterar o status do objeto desejado. Por exemplo:

```
aws datapipeline set-status --pipeline-id <pipeline-id> --region ap-northeast-1 --status TRY_CANCEL --object-ids <object-id>
```

Editar o pipeline

Para alterar algum aspecto de um dos pipelines, você poderá atualizar a definição do pipeline. Depois de alterar um pipeline em execução, você deverá reativar o pipeline para que as alterações entrem em vigor. Além disso, você pode reexecutar um ou mais componentes do pipeline.

Índice

- [Limitações](#)
- [Editar um pipeline usando a AWS CLI](#)

Limitações

Enquanto o pipeline estiver no estado `PENDING` e não estiver ativado, você não poderá fazer alterações nele. Depois de ativar um pipeline, você poderá editá-lo com as restrições a seguir. As alterações feitas por você se aplicarão a novas execuções dos objetos do pipeline depois de salvá-las e reativar o pipeline.

- Você não pode remover um objeto
- Você não pode alterar o período de programação de um objeto existente
- Você não pode adicionar, excluir nem modificar campos de referência em um objeto existente
- Você não pode fazer referência a um objeto existente em um campo de saída de um novo objeto
- Você não pode alterar a data de início programada de um objeto (em vez disso, ative o pipeline com uma data e uma hora específicas)

Editar um pipeline usando a AWS CLI

Você pode editar um pipeline usando as ferramentas de linha de comando.

Primeiro, faça download de uma cópia da definição do pipeline atual usando o comando [get-pipeline-definition](#). Fazendo isso, você pode ter a certeza de que está modificando a definição do pipeline mais recente. O exemplo a seguir usa a definição do pipeline para a saída padrão (stdout).

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE
```

Salve a definição do pipeline em um arquivo e a edite conforme necessário. Atualize a definição do pipeline usando o comando [put-pipeline-definition](#) a seguir. O exemplo a seguir faz upload do arquivo de definição de pipeline atualizado.

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --  
pipeline-definition file://MyEmrPipelineDefinition.json
```

Você pode recuperar novamente a definição do pipeline usando o comando [get-pipeline-definition](#) para garantir que a atualização tenha sido bem-sucedida. Para ativar o pipeline, use o seguinte comando [activate-pipeline](#):

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Se preferir, você poderá ativar o pipeline em uma data e uma hora específicas usando a opção `--start-timestamp` da seguinte forma:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --start-  
timestamp YYYY-MM-DDTHH:MM:SSZ
```

Para reexecutar um ou mais componentes do pipeline, use o comando [set-status](#).

Clonar o pipeline

Clonar faz uma cópia de um pipeline e permite especificar um nome para o novo pipeline. Você pode clonar um pipeline que esteja em qualquer estado, mesmo se tiver erros; no entanto, o novo pipeline permanecerá no estado PENDING até você ativá-lo manualmente. Para o novo pipeline, a operação de clonagem usa a versão mais recente da definição do pipeline original, e não a versão ativa. Na

operação de clonagem, a programação completa do pipeline original não é copiada para o novo pipeline, e sim somente a configuração do período.

Para clonar um pipeline usando a CLI de AWS:

1. Crie um novo pipeline com um novo nome e ID exclusivo. Observe o ID do pipeline retornado.
2. Use a CLI de `get-pipeline-definition` para obter a definição do pipeline existente a ser clonado e para gravá-la em um arquivo temporário. Observe o caminho absoluto do arquivo.
3. Use a CLI de `put-pipeline-definition` para copiar a definição do pipeline a partir do pipeline existente em um novo pipeline.
4. Use a CLI de `get-pipeline-definition` para obter a definição do novo pipeline e verificar a definição do pipeline.

```
# Create Pipeline (returns <new-pipeline-id>)
aws datapipeline create-pipeline --name my-cloned-pipeline --unique-id my-cloned-pipeline --region ap-northeast-1

#Get pipeline definition of existing pipeline
aws datapipeline get-pipeline-definition --pipeline-id <existing-pipeline-id> --region ap-northeast-1 > existing_pipeline_definition.json

# Put pipeline definition to new pipeline
aws datapipeline put-pipeline-definition --pipeline-id <new-pipeline-id> --region ap-northeast-1 --pipeline-definition file://<absolute_path_to_existing_pipeline_definition.json>

# get pipeline definition of new pipeline
aws datapipeline get-pipeline-definition --pipeline-id <new-pipeline-id> --region ap-northeast-1
```

Marcar o pipeline

Tags são pares de chave/valor que diferenciam maiúsculas de minúsculas e consistem em uma chave e um valor opcional, ambos definidos pelo usuário. Você pode aplicar até 10 tags a cada pipeline. As chaves de tag devem ser exclusivas para cada pipeline. Se você adicionar uma tag a uma chave que já esteja associada ao pipeline, isso atualizará o valor dessa tag.

Aplicar uma tag a um pipeline também propaga as tags para os recursos subjacentes (por exemplo, os clusters do Amazon EMR e as instâncias do Amazon EC2). No entanto, ele não aplica essas tags

a recursos em um estado FINISHED ou em um estado encerrado. Se necessário, você pode usar a CLI para aplicar tags a esses recursos.

Quando tiver terminado uma tag, você poderá removê-la do pipeline.

Para marcar o pipeline usando a CLI da AWS

Para adicionar tags a um novo pipeline, adicione a opção `--tags` ao comando [create-pipeline](#). Por exemplo, a opção a seguir cria um pipeline com duas tags, uma tag `environment` com um valor `production` e uma tag `owner` com um valor `sales`.

```
--tags key=environment,value=production key=owner,value=sales
```

Para adicionar tags a um pipeline existente, use o comando [add-tags](#) da seguinte maneira:

```
aws datapipeline add-tags --pipeline-id df-00627471S0VYZEXAMPLE --tags  
key=environment,value=production key=owner,value=sales
```

Para remover tags de um pipeline existente, use o comando [remove-tags](#) da seguinte maneira:

```
aws datapipeline remove-tags --pipeline-id df-00627471S0VYZEXAMPLE --tag-keys  
environment owner
```

Desativar o pipeline

Desativar um pipeline em execução pausa a execução do pipeline. Para retomar a execução do pipeline, você pode ativar o pipeline. Isso permite fazer alterações. Por exemplo, se estiver gravando dados em um banco de dados programado para passar por manutenção, você poderá desativar o pipeline, aguardar a conclusão da manutenção e ativar o pipeline.

Ao desativar um pipeline, você pode especificar o que acontece com atividades em execução. Por padrão, essas atividades são canceladas imediatamente. Como alternativa, você pode fazer o AWS Data Pipeline aguardar até as atividades serem concluídas antes de desativar o pipeline.

Ao ativar um pipeline desativado, você pode especificar quando ele é retomado. Usando a AWS CLI ou a API, o pipeline retoma a partir da execução concluída mais recentemente por padrão, ou você pode especificar a data e a hora para retomar o pipeline.

Desativar o pipeline usando a AWS CLI

Use o seguinte comando [deactivate-pipeline](#) para desativar um pipeline:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Para desativar o pipeline somente depois que todas as atividades em execução forem concluídas, adicione a opção `--no-cancel-active` da seguinte maneira:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --no-cancel-active
```

Quando estiver pronto, você poderá retomar a execução do pipeline de onde ela parou usando o comando [activate-pipeline](#):

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Para iniciar o pipeline em uma data e uma hora específicas, adicione a opção `--start-timestamp` da seguinte maneira:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --start-timestamp YYYY-MM-DDTHH:MM:SSZ
```

Excluir o pipeline

Quando não precisar mais de um pipeline, como um pipeline criado durante o teste de aplicativo, você deverá excluí-lo para removê-lo do uso ativo. Excluir um pipeline o coloca em um estado de exclusão. Quando o pipeline estiver no estado excluído, a definição do pipeline e o histórico de execuções serão eliminados. Por isso, você não pode mais realizar operações no pipeline, inclusive descrevê-lo.

Important

Você não poderá restaurar um pipeline depois de excluí-lo. Dessa forma, certifique-se de que você não precise do pipeline no futuro antes de excluí-lo.

Para excluir um pipeline usando a AWS CLI

Para excluir um pipeline, use o comando [delete-pipeline](#). O comando a seguir exclui o pipeline especificado.

```
aws datapipeline delete-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Preparar dados e tabelas com atividades de pipeline

O AWS Data Pipeline pode preparar dados de entrada e saída nos pipelines para facilitar o uso de determinadas atividades, como `ShellCommandActivity` e `HiveActivity`.

A preparação de dados permite a você copiar os dados do nó de dados de entrada para o recurso que executa a atividade e, de maneira semelhante, do recurso para o nó de dados de saída.

Os dados preparados no recurso Amazon EMR ou Amazon EC2 são disponibilizados usando variáveis especiais nos comandos shell da atividade ou nos scripts do Hive.

A preparação da tabela é semelhante à preparação dos dados, exceto pelos dados preparados assumirem a forma de tabelas de banco de dados, mais especificamente.

O AWS Data Pipeline oferece suporte aos seguintes cenários de preparação:

- Preparação de dados com `ShellCommandActivity`
- Preparação da tabela com Hive e nós de dados compatíveis com preparação
- Preparação da tabela com Hive e nós de dados incompatíveis com preparação

Note

A preparação funciona somente quando o campo `stage` é definido como `true` em uma atividade, como `ShellCommandActivity`. Para obter mais informações, consulte [ShellCommandActivity](#).

Além disso, os nós de dados e as atividades podem estar relacionados de quatro maneiras:

Preparar dados localmente em um recurso

Os dados de entrada são copiados automaticamente para o sistema de arquivos local. Os dados de saída são copiados automaticamente do sistema de arquivos local para o nó de dados de

saída. Por exemplo, quando você configura entradas e saídas `ShellCommandActivity` com `staging = true`, os dados de entrada são disponibilizados como `INPUTx_STAGING_DIR` e os dados de saída são disponibilizados como `OUTPUTx_STAGING_DIR`, em que `x` é o número de entrada ou saída.

Preparar definições de entrada e saída para uma atividade

O formato de dados de entrada (nomes de coluna e de tabela) é copiado automaticamente para o recurso da atividade. Por exemplo, quando você configura `HiveActivity` com `staging = true`. O formato de dados especificado na entrada `S3DataNode` é usado para preparar a definição da tabela Hive.

Preparação não ativada

Os objetos de entrada e saída e os campos estão disponíveis para a atividade, mas os dados não. Por exemplo, `EmrActivity` por padrão, ou quando você configura outras atividades com `staging = false`. Nessa configuração, os campos de dados estão disponíveis para a atividade fazer uma referência a eles usando a sintaxe da expressão do AWS Data Pipeline, e isso ocorre somente quando a dependência é atendida. Isso funciona somente como verificação de dependência. O código na atividade é responsável por copiar os dados da entrada para o recurso que executa a atividade.

Relação de dependência entre objetos

Existe uma relação de dependência entre dois objetos, o que resulta em uma situação semelhante a quando a preparação não está ativada. Isso faz um nó de dados ou uma atividade funcionar como uma pré-condição para a execução de outra atividade.

Preparação de dados com `ShellCommandActivity`

Considere um cenário que use um `ShellCommandActivity` com objetos `S3DataNode` como entrada e saída de dados. O AWS Data Pipeline prepara automaticamente os nós de dados para torná-los acessíveis ao comando shell como se fossem pastas de arquivos locais usando as variáveis de ambiente `${INPUT1_STAGING_DIR}` e `${OUTPUT1_STAGING_DIR}`, conforme mostrado no exemplo a seguir. A parte numérica das variáveis chamadas `INPUT1_STAGING_DIR` e `OUTPUT1_STAGING_DIR` é incrementada dependendo do número de nós de dados e das referências de atividade.

Note

Esse cenário funcionará somente conforme descrito se as entradas e as saídas de dados forem objetos S3DataNode. Além disso, a preparação de dados de saída é permitida somente quando `directoryPath` é definido no objeto S3DataNode de saída.

```
{
  "id": "AggregateFiles",
  "type": "ShellCommandActivity",
  "stage": "true",
  "command": "cat ${INPUT1_STAGING_DIR}/part* > ${OUTPUT1_STAGING_DIR}/aggregated.csv",
  "input": {
    "ref": "MyInputData"
  },
  "output": {
    "ref": "MyOutputData"
  }
},
{
  "id": "MyInputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "filePath": "s3://my_bucket/source/#{format(@scheduledStartTime, 'YYYY-MM-dd_HH:mm:ss')}/items"
},
{
  "id": "MyOutputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://my_bucket/destination/#{format(@scheduledStartTime, 'YYYY-MM-dd_HH:mm:ss')}"
},
...
```

Preparação da tabela com Hive e nós de dados compatíveis com preparação

Considere um cenário que use um `HiveActivity` com objetos `S3DataNode` como entrada e saída de dados. O AWS Data Pipeline prepara automaticamente os nós de dados para torná-los acessíveis ao script do Hive como se fossem tabelas do Hive usando as variáveis de ambiente `${input1}` e `${output1}`, conforme mostrado no exemplo a seguir para `HiveActivity`. A parte numérica das variáveis chamadas `input` e `output` é incrementada dependendo do número de nós de dados e das referências de atividade.

Note

Esse cenário funcionará somente conforme descrito se as entradas e as saídas de dados forem objetos `S3DataNode` ou `MySQLDataNode`. A preparação de tabelas não é compatível com `DynamoDBDataNode`.

```
{
  "id": "MyHiveActivity",
  "type": "HiveActivity",
  "schedule": {
    "ref": "MySchedule"
  },
},
"runsOn": {
  "ref": "MyEmrResource"
},
"input": {
  "ref": "MyInputData"
},
"output": {
  "ref": "MyOutputData"
},
"hiveScript": "INSERT OVERWRITE TABLE ${output1} select * from ${input1};"
},
{
  "id": "MyInputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
},
```

```
"directoryPath": "s3://test-hive/input"
}
},
{
  "id": "MyOutputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://test-hive/output"
}
},
...
```

Preparação da tabela com Hive e nós de dados incompatíveis com preparação

Considere um cenário que use um `HiveActivity` com `DynamoDBDataNode` como entrada de dados e um objeto `S3DataNode` como a saída. Nenhuma preparação de dados está disponível para `DynamoDBDataNode`. Por isso, você deve primeiro criar manualmente a tabela dentro do script do Hive usando o nome da variável `#{input.tableName}` para referenciar a tabela do DynamoDB. Uma nomenclatura semelhante se aplicará se a tabela do DynamoDB for a saída, exceto se você usar a variável `#{output.tableName}`. A preparação está disponível para o objeto `S3DataNode` de saída neste exemplo. Por isso, você pode se referir ao nó de dados de saída como `#{output1}`.

Note

Neste exemplo, a variável do nome da tabela tem o prefixo de caractere `#` (hash), pois o AWS Data Pipeline usa expressões para acessar o `tableName` ou `directoryPath`. Para obter mais informações sobre como a avaliação da expressão funciona no AWS Data Pipeline, consulte [Avaliação de expressões](#).

```
{
  "id": "MyHiveActivity",
  "type": "HiveActivity",
  "schedule": {
    "ref": "MySchedule"
  },
}
```

```
"runsOn": {
  "ref": "MyEmrResource"
},
"input": {
  "ref": "MyDynamoData"
},
"output": {
  "ref": "MyS3Data"
},
"hiveScript": "-- Map DynamoDB Table
SET dynamodb.endpoint=dynamodb.us-east-1.amazonaws.com;
SET dynamodb.throughput.read.percent = 0.5;
CREATE EXTERNAL TABLE dynamodb_table (item map<string,string>)
STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler'
TBLPROPERTIES ("dynamodb.table.name" = "#{input.tableName}");
INSERT OVERWRITE TABLE ${output1} SELECT * FROM dynamodb_table;"
},
{
  "id": "MyDynamoData",
  "type": "DynamoDBDataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "tableName": "MyDDBTable"
},
{
  "id": "MyS3Data",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://test-hive/output"
}
},
...
```

Usar um pipeline com recursos em várias regiões

Por padrão, os recursos `Ec2Resource` e `EmrCluster` são executados na mesma região do AWS Data Pipeline. No entanto, o AWS Data Pipeline oferece suporte à possibilidade de orquestrar fluxos de dados em várias regiões, como executar recursos em uma região que consolida dados de entrada de outra região. Ao permitir que os recursos sejam executados uma região específica,

você também tem a flexibilidade de colocar os recursos com conjuntos de dados dependentes e maximizar o desempenho reduzindo a latência e evitando cobranças de transferência de dados entre regiões. Você pode configurar recursos para serem executados em uma região diferente do AWS Data Pipeline usando o campo `region` em `Ec2Resource` e `EmrCluster`.

O arquivo JSON do pipeline de exemplo a seguir mostra como executar um recurso `EmrCluster` na região Europa (Irlanda), presumindo que haja uma grande quantidade de dados para o cluster trabalhar na mesma região. Neste exemplo, a única diferença em relação a um pipeline típico é que o `EmrCluster` tem um valor de campo `region` definido como `eu-west-1`.

```
{
  "objects": [
    {
      "id": "Hourly",
      "type": "Schedule",
      "startDateTime": "2014-11-19T07:48:00",
      "endDateTime": "2014-11-21T07:48:00",
      "period": "1 hours"
    },
    {
      "id": "MyCluster",
      "type": "EmrCluster",
      "masterInstanceType": "m3.medium",
      "region": "eu-west-1",
      "schedule": {
        "ref": "Hourly"
      }
    },
    {
      "id": "MyEmrActivity",
      "type": "EmrActivity",
      "schedule": {
        "ref": "Hourly"
      },
      "runsOn": {
        "ref": "MyCluster"
      },
      "step": "/home/hadoop/contrib/streaming/hadoop-streaming.jar, -input, s3n://elasticmapreduce/samples/wordcount/input, -output, s3://eu-west-1-bucket/wordcount/output/#{@scheduledStartTime}, -mapper, s3n://elasticmapreduce/samples/wordcount/wordSplitter.py, -reducer, aggregate"
    }
  ]
}
```

```
}
```

A tabela a seguir lista as regiões que você pode escolher e os códigos de região associados a serem usados no campo `region`.

Note

A lista a seguir inclui regiões em que o AWS Data Pipeline pode orquestrar fluxos de trabalho e executar os recursos do Amazon EMR ou Amazon EC2. O Pode ser que essas regiões não ofereçam suporte ao AWS Data Pipeline. Para obter informações sobre as regiões com suporte para o AWS Data Pipeline, consulte [Regiões e endpoints da AWS](#).

| Nome da região | Código da região |
|----------------------------------|------------------|
| Leste dos EUA (N. da Virgínia) | us-east-1 |
| Leste dos EUA (Ohio) | us-east-2 |
| Oeste dos EUA (N. da Califórnia) | us-west-1 |
| Oeste dos EUA (Oregon) | us-west-2 |
| Canadá (Central) | ca-central-1 |
| Europa (Irlanda) | eu-west-1 |
| Europa (Londres) | eu-west-2 |
| Europa (Frankfurt) | eu-central-1 |
| Ásia-Pacífico (Singapura) | ap-southeast-1 |
| Ásia-Pacífico (Sydney) | ap-southeast-2 |
| Ásia-Pacífico (Mumbai) | ap-south-1 |
| Ásia-Pacífico (Tóquio) | ap-northeast-1 |
| Ásia-Pacífico (Seul) | ap-northeast-2 |

| Nome da região | Código da região |
|----------------------------|------------------|
| América do Sul (São Paulo) | sa-east-1 |

Falhas e novas execuções em cascata

O AWS Data Pipeline permite configurar a maneira como objetos de pipeline se comportam quando há uma falha na dependência ou ela é cancelada por um usuário. Você pode verificar se as falhas chegam em cascata até outros objetos de pipeline (clientes) para evitar uma espera indefinida. Todas as atividades, nós de dados e precondições têm um campo chamado `failureAndRerunMode` com um valor padrão `none`. Para habilitar falhas em cascata, defina o campo `failureAndRerunMode` como `cascade`.

Quando esse campo está habilitado, haverá falhas em cascata se um objeto de pipeline for bloqueado no estado `WAITING_ON_DEPENDENCIES` e eventuais dependências tiverem falhado sem comando pendente. Durante uma falha em cascata, ocorrem os seguintes eventos:

- Quando um objeto falha, os clientes são definidos como `CASCADE_FAILED`, e o objeto original e as precondições dos clientes são definidos como `CANCELED`.
- Todos os objetos que já estejam em `FINISHED`, `FAILED` ou `CANCELED` são ignorados.

A falha em cascata não funciona em dependências (upstream) de um objeto com falha, exceto em precondições associadas ao objeto de falha original. Os objetos de pipeline afetados por uma falha em cascata podem disparar eventuais novas tentativas ou pós-ações, como `onFail`.

Os efeitos detalhados de uma falha em cascata dependem do tipo de objeto.

Atividades

Uma atividade será alterada para `CASCADE_FAILED` se alguma das dependências falhar e, assim, disparar uma falha em cascata nos clientes da atividade. Se um recurso do qual a atividade depende falhar, a atividade será `CANCELED`, e todos os clientes serão alterados para `CASCADE_FAILED`.

Nós de dados e pré-condições

Se um nó de dados for configurado como a saída de uma atividade de falha, o nó de dados será alterado para o estado `CASCADE_FAILED`. A falha de um nó de dados é propagada para qualquer precondição associada, que muda para o estado `CANCELED`.

Recursos

Se os objetos dos quais um recurso dependa estiverem no estado FAILED e o recurso propriamente dito estiver no estado WAITING_ON_DEPENDENCIES, o recurso mudará para o estado FINISHED.

Reexecutar objetos de falha em cascata

Por padrão, reexecutar qualquer atividade ou nó de dados reexecuta somente o recurso associado. No entanto, definir o campo `failureAndRerunMode` como `cascade` em um objeto de pipeline permite um comando rerun em um objeto de destino a ser propagado para todos os clientes, sob as seguintes condições:

- Os clientes do objeto de destino estão no estado `CASCADE_FAILED`.
- As dependências do objeto de destino não têm comandos rerun pendentes.
- As dependências do objeto de destino não estão no estado `FAILED`, `CASCADE_FAILED` ou `CANCELED`.

Se você tentar reexecutar um objeto `CASCADE_FAILED` e qualquer uma das dependências for `FAILED`, `CASCADE_FAILED` ou `CANCELED`, a nova execução vai falhar e retornar o objeto ao estado `CASCADE_FAILED`. Para reexecutar o objeto de falha, você deve rastrear a falha até a cadeia de dependência para localizar a origem da falha e reexecutar esse objeto. Ao executar um comando rerun em um recurso, você também tenta reexecutar todos os objetos que dependam dele.

Falha em cascata e preenchimentos

Se você habilitar a falha em cascata e tiver um pipeline que cria várias alocações, erros de tempo de execução do pipeline podem fazer com que os recursos sejam criados e excluídos em rápida sucessão sem realizar trabalho útil. O AWS Data Pipeline tenta alertar você sobre essa situação com a seguinte mensagem de aviso exibida ao salvar um pipeline: *Pipeline_object_name* has 'failureAndRerunMode' field set to 'cascade' and you are about to create a backfill with scheduleStartTime *start_time*. This can result in rapid creation of pipeline objects in case of failures. Isso acontece porque a falha em cascata pode definir rapidamente atividades de downstream como `CASCADE_FAILED` e desligar clusters do EMR e recursos do EC2 que deixaram de ser necessários. Recomendamos testar pipelines com intervalos de tempo curtos para limitar os efeitos dessa situação.

Sintaxe do arquivo de definição do pipeline

As instruções nesta seção são para trabalhar manualmente com os arquivos de definição de pipeline usando a interface de linha de comando (CLI) do AWS Data Pipeline. Trata-se de uma alternativa ao projeto de um pipeline de maneira interativa usando o console do AWS Data Pipeline.

Você pode criar manualmente arquivos de definição de pipeline usando qualquer editor de texto que ofereça suporte à gravação de arquivos usando o formato de arquivo UTF-8 e enviar os arquivos usando a interface de linha de comando do AWS Data Pipeline.

O AWS Data Pipeline também oferece suporte a uma grande variedade de expressões e funções complexas dentro de definições do pipeline. Para obter mais informações, consulte [Expressões e funções do pipeline](#).

Estrutura do arquivo

A primeira etapa na criação do pipeline é escrever objetos de definição do pipeline em um arquivo de definição do pipeline. O exemplo a seguir ilustra a estrutura geral de um arquivo de definição do pipeline. Esse arquivo define dois objetos, que são delimitados por '{' e '}' e separados por uma vírgula.

No exemplo a seguir, o primeiro objeto define dois pares de nome/valor, conhecidos como campos. O segundo objeto define três campos.

```
{
  "objects" : [
    {
      "name1" : "value1",
      "name2" : "value2"
    },
    {
      "name1" : "value3",
      "name3" : "value4",
      "name4" : "value5"
    }
  ]
}
```

Ao criar um arquivo de definição de pipeline, você deve selecionar os tipos de objetos de pipeline dos quais precisará, adicioná-los ao arquivo de definição de pipeline e incluir os campos apropriados. Para obter mais informações sobre objetos de pipeline, consulte [Referência de objeto de pipeline](#).

Por exemplo, você pode criar um objeto de definição de pipeline para um nó de dados de entrada e outro para o nó de dados de saída. Em seguida, crie outro objeto de definição de pipeline para uma atividade, como processar os dados de entrada usando o Amazon EMR.

Campos de pipeline

Depois que souber quais tipos de objeto incluir no arquivo de definição de pipeline, você adicionará campos à definição de cada objeto de pipeline. Os nomes de campo estão entre aspas e são separados por valores de campo por um espaço, uma vírgula e um espaço, conforme mostrado no exemplo a seguir.

```
"name" : "value"
```

O valor do campo pode ser uma string de texto, uma referência a outro objeto, uma chamada à função, uma expressão ou uma lista ordenada de qualquer um dos tipos anteriores. Para obter mais informações sobre os tipos de dados que podem ser usados em valores de campo, consulte [Tipos de dados simples](#). Para obter mais informações sobre funções que você pode usar para avaliar valores de campo, consulte [Avaliação de expressões](#).

Os campos são limitados a 2048 caracteres. Os objetos podem ter 20 KB, o que significa que você não pode adicionar muitos campos grandes a um objeto.

Cada objeto de pipeline deve conter os seguintes campos: `id` e `type`, conforme mostrado no exemplo a seguir. Outros campos também podem ser necessários com base no tipo de objeto. Selecione um valor para `id` que seja significativo para você e exclusivo dentro da definição de pipeline. O valor de `type` especifica o tipo do objeto. Especifique um dos tipos de objeto de definição de pipeline compatíveis, listados no tópico [Referência de objeto de pipeline](#).

```
{
  "id": "MyCopyToS3",
  "type": "CopyActivity"
}
```

Para obter mais informações sobre os campos obrigatórios e opcionais de cada objeto, consulte a documentação do objeto.

Para incluir campos de um objeto em outro objeto, use o campo `parent` com uma referência ao objeto. Por exemplo, o objeto "B" inclui os campos, "B1" e "B2", mais os campos de objeto "A", "A1" e "A2".

```
{
  "id" : "A",
  "A1" : "value",
  "A2" : "value"
},
{
  "id" : "B",
  "parent" : {"ref" : "A"},
  "B1" : "value",
  "B2" : "value"
}
```

Você pode definir campos comuns em um objeto com o ID "padrão". Esses campos são incluídos automaticamente em todos os objetos no arquivo de definição de pipeline que não definam explicitamente o campo `parent` para referenciar um objeto diferente.

```
{
  "id" : "Default",
  "onFail" : {"ref" : "FailureNotification"},
  "maximumRetries" : "3",
  "workerGroup" : "myWorkerGroup"
}
```

Campos definidos pelo usuário

Você pode criar campos personalizados ou definidos pelo usuário nos componentes de pipeline e consultá-los com expressões. O exemplo a seguir mostra um campo personalizado chamado `myCustomField` e `my_customFieldReference` adicionado a um objeto `S3DataNode`:

```
{
  "id": "S3DataInput",
  "type": "S3DataNode",
  "schedule": {"ref": "TheSchedule"},
  "filePath": "s3://bucket_name",
  "myCustomField": "This is a custom value in a custom field.",
  "my_customFieldReference": {"ref": "AnotherPipelineComponent"}
},
```

Um campo definido pelo usuário deve ter um nome prefixado com a palavra "my" em todas as letras minúsculas, seguido de uma letra maiúscula ou sublinhado. Além disso, um campo definido pelo

usuário pode ser um valor de string, como o exemplo `myCustomField` anterior, ou uma referência a outro componente de pipeline, como o exemplo `my_customFieldReference` anterior.

Note

Em campos definidos pelo usuário, o AWS Data Pipeline verifica somente se há referências válidas a outros componentes, e não a qualquer valor de string de campo personalizado adicionado por você.

Trabalhar com a API

Note

Se você não estiver escrevendo programas que interagem com o AWS Data Pipeline, não precisará instalar nenhum SDK da AWS. Você pode criar e executar pipelines usando o console ou a interface da linha de comando. Para obter mais informações, consulte [Configurando para AWS Data Pipeline](#).

A maneira mais fácil de escrever aplicativos que interagem com o AWS Data Pipeline ou de implementar um Task Runner personalizado é usar um dos SDKs da AWS. Os SDKs da AWS fornecem funcionalidades que simplificam a chamada das APIs de serviço web a partir do seu ambiente de programação preferido. Para obter mais informações, consulte [Instalar o SDK da AWS](#).

Instalar o SDK da AWS

Os SDKs da AWS fornecem funções que encapsulam a API e cuidam de muitos dos detalhes de conexão, como o cálculo de assinaturas, o tratamento de novas tentativas de solicitação e o tratamento de erros. Os SDKs também contêm código de exemplo, tutoriais e outros recursos para ajudá-lo a começar a escrever aplicativos que fazem chamadas à AWS. Uma chamada para uma função wrapper em um SDK pode simplificar muito o processo de criação de um aplicativo da AWS. Para obter mais informações sobre como fazer download e usar os SDKs da AWS, acesse [Código de exemplo e bibliotecas](#).

O suporte do AWS Data Pipeline está disponível em SDKs das seguintes plataformas:

- [AWS SDK para Java](#)

- [AWS SDK para Node.js](#)
- [AWS SDK para PHP](#)
- [AWS SDK para Python \(Boto\)](#)
- [AWS SDK para Ruby](#)
- [AWS SDK para .NET](#)

Fazer uma solicitação HTTP para o AWS Data Pipeline

Para obter uma descrição completa dos objetos programáticos no AWS Data Pipeline, consulte a [Referência de API do AWS Data Pipeline](#).

Se você não usa nenhum dos SDKs da AWS, pode executar as operações do AWS Data Pipeline por meio de HTTP usando o método de solicitação POST. O método POST exige a especificação da operação no cabeçalho da solicitação e o fornecimento de dados para operação no formato JSON no corpo da solicitação.

Conteúdo de cabeçalho HTTP

O AWS Data Pipeline exige as informações a seguir no cabeçalho de uma solicitação HTTP:

- `host` O endpoint do AWS Data Pipeline.

Para obter informações sobre endpoints, consulte [Regiões e endpoints](#).

- `x-amz-date` Você precisa fornecer o time stamp no cabeçalho `Date` em HTTP ou no cabeçalho `x-amz-date` da AWS. (Algumas bibliotecas de cliente HTTP não permitem a definição do cabeçalho `Date`). Quando existe um cabeçalho `x-amz-date`, o sistema ignora qualquer cabeçalho `Date` durante a autenticação de uma solicitação.

A data precisa ser especificada em um destes três formatos, conforme especificado em HTTP/1.1 RFC:

- Domingo, 06-Nov-1994 08:49:37 GMT (RFC 822, atualizada pela RFC 1123)
- Domingo, 06-Nov-94 08:49:37 GMT (RFC 850, substituído por RFC 1036)
- Dom Nov 6 08:49:37 1994 (formato ANSI C `asctime()`)
- `Authorization` O conjunto de parâmetros de autorização que a AWS usa para garantir a validade e a autenticidade da solicitação. Para obter mais informações sobre como criar esse cabeçalho, acesse [Processo de assinatura do Signature versão 4](#).

- `x-amz-target` O serviço de destino da solicitação e a operação para os dados, no formato: `<<serviceName>>_<<API version>>.<<operationName>>`

Por exemplo, `DataPipeline_20121129.ActivatePipeline`

- `content-type` Especifica o JSON e a versão. Por exemplo, `Content-Type: application/x-amz-json-1.0`

Veja a seguir um exemplo de cabeçalho para uma solicitação HTTP para ativar um pipeline.

```
POST / HTTP/1.1
host: https://datapipeline.us-east-1.amazonaws.com
x-amz-date: Mon, 12 Nov 2012 17:49:52 GMT
x-amz-target: DataPipeline_20121129.ActivatePipeline
Authorization: AuthParams
Content-Type: application/x-amz-json-1.1
Content-Length: 39
Connection: Keep-Alive
```

Conteúdo do corpo HTTP

O corpo de uma solicitação HTTP apresenta os dados da operação especificada no cabeçalho da solicitação HTTP. Os dados devem estar formatados de acordo com o esquema de dados JSON para cada API do AWS Data Pipeline. O esquema de dados JSON do AWS Data Pipeline define os tipos de dados e parâmetros (como operadores de comparação e constantes de enumeração) disponíveis para cada operação.

Formatar o corpo de uma solicitação HTTP

Use o formato de dados JSON para transmitir valores e estrutura de dados, simultaneamente. Os elementos podem ser aninhados dentro de outros elementos usando a notação de colchetes. O exemplo a seguir mostra uma solicitação para colocação de uma definição de pipeline que consiste em três objetos e os seus slots correspondentes.

```
{
  "pipelineId": "df-00627471S0VYZEXAMPLE",
  "pipelineObjects":
  [
    {"id": "Default",
```

```
"name": "Default",
"slots":
[
  {"key": "workerGroup",
   "stringValue": "MyWorkerGroup"}
]
},
{"id": "Schedule",
 "name": "Schedule",
 "slots":
  [
    {"key": "startDateTime",
     "stringValue": "2012-09-25T17:00:00"},
    {"key": "type",
     "stringValue": "Schedule"},
    {"key": "period",
     "stringValue": "1 hour"},
    {"key": "endDateTime",
     "stringValue": "2012-09-25T18:00:00"}
  ]
},
{"id": "SayHello",
 "name": "SayHello",
 "slots":
  [
    {"key": "type",
     "stringValue": "ShellCommandActivity"},
    {"key": "command",
     "stringValue": "echo hello"},
    {"key": "parent",
     "refValue": "Default"},
    {"key": "schedule",
     "refValue": "Schedule"}
  ]
}
]
```

Lidar com resposta HTTP

A seguir são apresentados alguns cabeçalhos importantes na resposta HTTP e a explicação sobre como você deve lidar com eles em seu aplicativo:

- HTTP/1.1 – Esse cabeçalho é acompanhado de um código de status. O valor de código 200 indica uma operação bem-sucedida. Qualquer outro valor indica um erro.
- x-amzn-RequestId – Esse cabeçalho contém um ID de solicitação que você pode usar se precisar solucionar problemas de uma solicitação com o AWS Data Pipeline. Um exemplo de ID de solicitação: K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG.
- x-amz-crc32 – O AWS Data Pipeline calcula uma soma de verificação CRC32 da carga útil HTTP e retorna essa soma de verificação no cabeçalho x-amz-crc32. Recomendamos que você calcule sua própria soma de verificação CRC32 no lado do cliente e a compare com a do cabeçalho x-amz-crc32. Se as somas de verificação não se corresponderem, é possível que os dados tenham sido corrompidos em trânsito. Se isso acontecer, tente enviar sua solicitação novamente.

Os usuários do SDK da AWS não precisam realizar essa verificação manualmente, pois os SDKs calculam a soma de verificação de cada resposta do Amazon DynamoDB e realizam automaticamente novas tentativas se for detectada falta de correspondência.

Exemplo de solicitação e resposta JSON AWS Data Pipeline

Os exemplos a seguir mostram uma solicitação para criar um novo pipeline. Em seguida, a resposta do AWS Data Pipeline é exibida, incluindo o identificador do pipeline recém-criado.

Solicitação HTTP POST

```
POST / HTTP/1.1
host: https://datapipeline.us-east-1.amazonaws.com
x-amz-date: Mon, 12 Nov 2012 17:49:52 GMT
x-amz-target: DataPipeline_20121129.CreatePipeline
Authorization: AuthParams
Content-Type: application/x-amz-json-1.1
Content-Length: 50
Connection: Keep-Alive

{"name": "MyPipeline",
 "uniqueId": "12345ABCDEF"}
```


Resposta do AWS Data Pipeline

```
HTTP/1.1 200
x-amzn-RequestId: b16911ce-0774-11e2-af6f-6bc7a6be60d9
x-amz-crc32: 2215946753
Content-Type: application/x-amz-json-1.0
Content-Length: 2
Date: Mon, 16 Jan 2012 17:50:53 GMT

{"pipelineId": "df-00627471S0VYZEXAMPLE"}
```

Segurança em AWS Data Pipeline

A segurança para com a nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de datacenters e arquiteturas de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem:** a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao AWS Data Pipeline, consulte [Serviços da AWS no escopo por programa de conformidade](#).
- **Segurança na nuvem:** sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AWS Data Pipeline. Os tópicos a seguir mostram como configurar o AWS Data Pipeline para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros serviços da AWS que ajudam a monitorar e proteger os recursos do AWS Data Pipeline.

Tópicos

- [Proteção de dados no AWS Data Pipeline](#)
- [Identity and Access Management para o AWS Data Pipeline](#)
- [Registro em log e monitoramento no AWS Data Pipeline](#)
- [Resposta a incidentes no AWS Data Pipeline](#)
- [Validação de Compatibilidade para AWS Data Pipeline](#)
- [Resiliência no AWS Data Pipeline](#)
- [Segurança da infraestrutura no AWS Data Pipeline](#)
- [Análise de vulnerabilidade e configuração no AWS Data Pipeline](#)

Proteção de dados no AWS Data Pipeline

O [modelo de responsabilidade compartilhada](#) da AWS se aplica à proteção de dados no AWS Data Pipeline. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos serviços da AWS que você usa. Para ter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#). Para ter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and GDPR](#) no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da Conta da AWS e configure as contas de usuário individuais com o AWS IAM Identity Center ou o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Recomendamos TLS 1.2 ou posterior.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão dos serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).
- O AWS Data Pipeline suporta IMDSv2 para recursos do Amazon EMR e do Amazon EC2. Para usar o IMDSv2 com o Amazon EMR, use as versões 5.23.1, 5.27.1 ou 5.32 ou posterior ou a versão 6.2 ou posterior. Para obter mais informações, consulte [Configurar solicitações de serviço de metadados para instâncias do Amazon EC2](#) e [Usar o IMDSv2](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso inclui trabalhar com a AWS Data Pipeline ou outros serviços da AWS

usando o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Identity and Access Management para o AWS Data Pipeline

Suas credenciais de segurança identificam você para os serviços na AWS e concedem permissões para usar recursos da AWS, como os pipelines. Você pode usar recursos do AWS Data Pipeline e do AWS Identity and Access Management (IAM) para permitir que o AWS Data Pipeline e outros usuários acessem seus recursos do AWS Data Pipeline sem compartilhar suas credenciais de segurança.

As organizações podem compartilhar o acesso aos pipelines para que os indivíduos dessa organização possam desenvolvê-los e mantê-los de maneira colaborativa. No entanto, por exemplo, pode ser necessário fazer o seguinte:

- Controlar quais usuários podem acessar pipelines específicos
- Proteger um pipeline de produção contra edições não intencionais
- Permitir que um auditor tenha acesso somente leitura aos pipelines, mas evitar que eles façam alterações

O AWS Data Pipeline é integrado ao AWS Identity and Access Management (IAM), que oferece uma ampla variedade de recursos:

- Criar usuários e grupos na sua Conta da AWS.
- Compartilhar facilmente seus AWS recursos entre os usuários do seu Conta da AWS.
- Atribuir credenciais de segurança exclusivas a cada usuário.
- Controlar o acesso do usuário a serviços e recursos.
- Obter uma única fatura para todos os usuários do seu Conta da AWS.

Ao usar o IAM com o AWS Data Pipeline, você pode controlar se os usuários de sua organização podem executar uma tarefa usando ações específicas da API do e se podem usar recursos específicos da AWS. Você pode usar as políticas do IAM com base nas tags de pipeline e em grupos de operadores para compartilhar seus pipelines com outros usuários e controlar o nível de acesso desses usuários.

Índice

- [Políticas do IAM para o AWS Data Pipeline](#)
- [Exemplo de políticas para o AWS Data Pipeline](#)
- [Funções do IAM para o AWS Data Pipeline](#)

Políticas do IAM para o AWS Data Pipeline

Por padrão, entidades do IAM não têm permissão para criar ou modificar recursos da AWS. Para permitir que entidades do IAM criem ou modifiquem recursos e realizem tarefas, crie políticas do IAM que concedam às entidades do IAM permissão para usar os recursos específicos e as ações de API de que precisam e, então, anexar essas políticas às entidades do IAM que exijam essas permissões.

Quando você anexa uma política a um usuário ou grupo de usuários, isso concede ou nega aos usuários permissão para realizar as tarefas especificadas nos recursos especificados. Para obter mais informações gerais sobre as políticas do IAM, consulte [Permissões e políticas](#) no Guia do usuário do IAM. Para obter mais informações sobre como gerenciar e criar políticas personalizadas do IAM, consulte [Gerenciamento de políticas do IAM](#).

Índice

- [Sintaxe da política](#)
- [Controlar acesso aos pipelines usando tags](#)
- [Controlar acesso aos pipelines usando grupos de operadores](#)

Sintaxe da política

A política do IAM é um documento JSON que consiste em uma ou mais declarações. Cada instrução é estruturada da seguinte maneira:

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "*",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  ]
}
```

```
    }  
  }  
}  
]  
}
```

Uma instrução de política inclui os seguintes elementos:

- **Effect:** o efeito pode ser `Allow` ou `Deny`. Por padrão, as entidades do IAM não têm permissão para usar recursos e ações da API. Por isso, todas as solicitações são negadas. Uma permissão explícita substitui o padrão. Uma negação explícita substitui todas as permissões.
- **Action:** a ação é a ação de API específica para a qual você está concedendo ou negando permissão. Para ver a lista de ações do AWS Data Pipeline, consulte [Ações](#) na Referência da API do AWS Data Pipeline.
- **Resource:** o recurso afetado pela ação. O único valor válido aqui é `"*"`.
- **Condition:** condições são opcionais. Elas podem ser usadas para controlar quando as políticas entrarão em vigor.

O AWS Data Pipeline implementa as chaves de contexto de toda AWS (consulte [Chaves disponíveis para condições](#)), mais as chaves específicas do serviço a seguir.

- `datapipeline:PipelineCreator` — Para conceder acesso ao usuário que criou o pipeline. Para obter um exemplo, consulte [Conceder acesso total ao proprietário do pipeline](#).
- `datapipeline:Tag` — Para conceder acesso com base na marcação de pipeline. Para obter mais informações, consulte [Controlar acesso aos pipelines usando tags](#).
- `datapipeline:workerGroup` — Para conceder acesso de acordo com o nome do grupo de operadores. Para obter mais informações, consulte [Controlar acesso aos pipelines usando grupos de operadores](#).

Controlar acesso aos pipelines usando tags

Você pode criar políticas do IAM que fazem referência às tags para o pipeline. Isso permite que você use a marcação de pipeline para fazer o seguinte:

- Conceder acesso somente leitura ao pipeline
- Conceder acesso de leitura/gravação ao pipeline
- Bloquear o acesso ao pipeline

Por exemplo, imagine que um gerente tenha dois ambientes de pipeline, produção e desenvolvimento, e um grupo do IAM para cada ambiente. Para os pipelines no ambiente de produção, o gerente concede acesso de leitura/gravação aos usuários no grupo do IAM de produção, mas concede acesso somente leitura aos usuários no grupo do IAM de desenvolvedor. Para os pipelines no ambiente de desenvolvimento, o gerente concede acesso de leitura/gravação aos grupos do IAM de produção e desenvolvedor.

Para alcançar esse cenário, o gerente marca os pipelines de produção com a tag "environment=production" e anexa a política a seguir para o grupo do IAM de desenvolvedor. A primeira instrução concede acesso somente leitura a todos os pipelines. A segunda instrução concede acesso de leitura/gravação aos pipelines que não têm uma tag "environment=production".

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:Describe*",
        "datapipeline:ListPipelines",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:QueryObjects"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "datapipeline:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {"datapipeline:Tag/environment": "production"}
      }
    }
  ]
}
```

Além disso, o gerente anexa a política a seguir ao grupo do IAM de produção. Esta instrução concede acesso total a todos os pipelines.

```
{
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": "datapipeline:*",  
    "Resource": "*"  
  }  
]  
}
```

Para obter mais exemplos, consulte [Conceder acesso somente leitura aos usuários com base em uma tag](#) e [Conceder acesso total aos usuários com base em uma tag](#).

Controlar acesso aos pipelines usando grupos de operadores

Você pode criar políticas do IAM que fazem referência a nomes de grupos de operadores.

Por exemplo, imagine que um gerente tenha dois ambientes de pipeline, produção e desenvolvimento, e um grupo do IAM para cada ambiente. O gerente tem três servidores de banco de dados com executores de tarefas configurados para ambientes de produção, pré-produção e desenvolvimento, respectivamente. O gerente deseja garantir que os usuários no grupo do IAM de produção possam criar pipelines que enviam tarefas para recursos de produção, e que os usuários no grupo do IAM de desenvolvimento possam criar pipelines que enviam tarefas para recursos de pré-produção e desenvolvimento.

Para alcançar esse cenário, o gerente instala um executor de tarefas nos recursos de produção com credenciais de produção e define `workerGroup` como "prodresource". Além disso, o gerente instala um executor de tarefas nos recursos de desenvolvimento com credenciais de desenvolvimento e define `workerGroup` como "pre-production" e "development". O gerente anexa a política a seguir ao grupo do IAM de desenvolvedor para bloquear o acesso aos recursos "prodresource". A primeira instrução concede acesso somente leitura a todos os pipelines. A segunda instrução concede acesso de leitura/gravação a pipelines quando o nome do grupo de operadores tem um prefixo "dev" ou "pre-prod".

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "datapipeline:Describe*",  
        "datapipeline:ListPipelines",  
      ]  
    }  
  ]  
}
```



```

    "datapipeline:GetPipelineDefinition",
    "datapipeline:QueryObjects"
  ],
  "Resource": "*"
},
{
  "Action": "datapipeline:*",
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "datapipeline:workerGroup": ["dev*", "pre-prod*"]
    }
  }
}
]
}

```

Além disso, o gerente anexa a política a seguir ao grupo do IAM de produção para conceder acesso aos recursos "prodresource". A primeira instrução concede acesso somente leitura a todos os pipelines. A segunda instrução concede acesso de leitura/gravação quando o nome do grupo de operadores tem um prefixo "prod".

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:Describe*",
        "datapipeline:ListPipelines",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:QueryObjects"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "datapipeline:*",
      "Resource": "*",
      "Condition": {
        "StringLike": {"datapipeline:workerGroup": "prodresource*"}
      }
    }
  ]
}

```

```
}  
]  
}
```

Exemplo de políticas para o AWS Data Pipeline

Os exemplos a seguir demonstram como conceder aos usuários acesso total ou restrito a pipelines.

Índice

- [Exemplo 1: Conceder aos usuários acesso somente leitura baseado em uma tag](#)
- [Exemplo 2: Conceder aos usuários acesso total baseado em uma tag](#)
- [Exemplo 3: Conceder acesso total ao proprietário do pipeline](#)
- [Exemplo 4: Conceder aos usuários acesso ao console do AWS Data Pipeline](#)

Exemplo 1: Conceder aos usuários acesso somente leitura baseado em uma tag

A política a seguir permite que os usuários usem as ações somente leitura da API do AWS Data Pipeline, mas apenas com pipelines que têm a tag “environment=production”.

A ação de API ListPipelines não oferece suporte à autorização com base na tag.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "datapipeline:Describe*",  
        "datapipeline:GetPipelineDefinition",  
        "datapipeline:ValidatePipelineDefinition",  
        "datapipeline:QueryObjects"  
      ],  
      "Resource": [  
        "*"   
      ],  
      "Condition": {  
        "StringEquals": {  
          "datapipeline:Tag/environment": "production"  
        }  
      }  
    }  
  ]  
}
```

```
]
}
```

Exemplo 2: Conceder aos usuários acesso total baseado em uma tag

A política a seguir permite que os usuários usem todas as ações de API do AWS Data Pipeline exceto ListPipelines, mas apenas com pipelines com a tag “environment=test”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "datapipeline:Tag/environment": "test"
        }
      }
    }
  ]
}
```

Exemplo 3: Conceder acesso total ao proprietário do pipeline

A política a seguir permite que os usuários usem todas as ações de API do AWS Data Pipeline, mas apenas com seus próprios pipelines.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:*"
      ],
      "Resource": [
```

```
        "*"
    ],
    "Condition": {
        "StringEquals": {
            "datapipeline:PipelineCreator": "${aws:userid}"
        }
    }
}
]
```

Exemplo 4: Conceder aos usuários acesso ao console do AWS Data Pipeline

A política a seguir permite que os usuários criem e gerenciem um pipeline com o console do AWS Data Pipeline.

Esta política inclui a ação de permissões do PassRole para recursos específicos vinculados ao roleARN de que o AWS Data Pipeline precisa. Para obter mais informações sobre a permissão PassRole baseada em identidade (IAM), consulte a publicação do blog [Conceder permissão para executar instâncias do EC2 com funções do IAM \(Permissão PassRole\)](#).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudwatch:*",
      "datapipeline:*",
      "dynamodb:DescribeTable",
      "elasticmapreduce:AddJobFlowSteps",
      "elasticmapreduce:ListInstance*",
      "iam:AddRoleToInstanceProfile",
      "iam:CreateInstanceProfile",
      "iam:GetInstanceProfile",
      "iam:GetRole",
      "iam:GetRolePolicy",
      "iam:ListInstanceProfiles",
      "iam:ListInstanceProfilesForRole",
      "iam:ListRoles",
      "rds:DescribeDBInstances",
      "rds:DescribeDBSecurityGroups",
      "redshift:DescribeClusters",
      "redshift:DescribeClusterSecurityGroups",
      "s3:List*",

```

```
    "sns:ListTopics"
  ],
  "Effect": "Allow",
  "Resource": [
    "*"
  ]
},
{
  "Action": "iam:PassRole",
  "Effect": "Allow",
  "Resource": [
    "arn:aws:iam::*:role/DataPipelineDefaultResourceRole",
    "arn:aws:iam::*:role/DataPipelineDefaultRole"
  ]
}
]
```

Funções do IAM para o AWS Data Pipeline

AWS Data Pipeline usa funções de AWS Identity and Access Management. As políticas de permissões associadas ao perfil do IAM determinam quais ações AWS Data Pipeline e seus aplicativos podem realizar e quais recursos de AWS eles podem acessar. Para obter mais informações, consulte [Funções do IAM](#) no Guia do usuário do IAM.

AWS Data Pipeline requer dois perfis do IAM:

- O perfil do pipeline controla o AWS Data Pipeline acesso aos recursos da AWS. Nas definições de objetos de pipeline, o campo `role` especifica essa função.
- O perfil de instância do EC2 controla o acesso que os aplicativos executados nas instâncias do EC2, incluindo as instâncias do EC2 nos clusters do Amazon EMR, têm aos recursos de AWS. Nas definições de objetos de pipeline, o campo `resourceRole` especifica essa função.

Important

Se você criou um pipeline antes de 3 de outubro de 2022 usando o console do AWS Data Pipeline com funções padrão, AWS Data Pipeline criou o `DataPipelineDefaultRole` para você e anexou a política gerenciada pelo `AWSDataPipelineRole` à função. A partir de 3 de outubro de 2022, a política gerenciada pelo `AWSDataPipelineRole` foi descontinuada e a função do pipeline deve ser especificada para um pipeline ao usar o console.

Recomendamos que você analise os pipelines existentes e determine se `DataPipelineDefaultRole` está associado ao pipeline e se `AWSDataPipelineRole` está associado a essa função. Nesse caso, revise o acesso que essa política permite para garantir que ela seja adequada aos seus requisitos de segurança. Adicione, atualize ou substitua as políticas e declarações de política anexadas a essa função, conforme necessário. Como alternativa, você pode atualizar um pipeline para usar uma função criada com diferentes políticas de permissões.

Exemplo de políticas de permissões para funções de AWS Data Pipeline

Cada função tem uma ou mais políticas de permissões anexadas que determinam os recursos de AWS que a função pode acessar e as ações que a função pode realizar. Este tópico fornece um exemplo de política de permissões para a função de pipeline. Ele também fornece o conteúdo da `AmazonEC2RoleforDataPipelineRole`, que é a política gerenciada para a função de instância padrão do EC2, `DataPipelineDefaultResourceRole`.

Exemplo de política de permissões da função de pipeline

O exemplo de política a seguir tem como escopo permitir funções essenciais que AWS Data Pipeline exigem para executar um pipeline com recursos do Amazon EC2 e do Amazon EMR. Ele também fornece permissões para acessar outros recursos de AWS, como o Amazon Simple Storage Service e o Amazon Simple Notification Service, que muitos pipelines exigem. Se os objetos definidos em um pipeline não exigirem os recursos de um serviço de AWS, recomendamos que você remova as permissões para acessar esse serviço. Por exemplo, se seu pipeline não definir um [y`ynamoDBDataNodo D`](#) ou usar a ação [SnsAlarm](#), recomendamos que você remova as instruções de permissão para essas ações.

- Substitua `111122223333` pelo ID de sua conta da AWS.
- Substitua `NameOfDataPipelineRole` pelo nome da função do pipeline (a função à qual essa política está anexada).
- Substitua `NameOfDataPipelineResourceRole` pelo nome da função da instância EC2.
- Substitua `us-west-1` pela região apropriada para sua aplicação.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  

```

```
{
  "Effect": "Allow",
  "Action": [
    "iam:GetInstanceProfile",
    "iam:GetRole",
    "iam:GetRolePolicy",
    "iam:ListAttachedRolePolicies",
    "iam:ListRolePolicies",
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam::111122223333:role/NameOfDataPipelineRole",
    "arn:aws:iam::111122223333 :role/NameOfDataPipelineResourceRole"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:AuthorizeSecurityGroupEgress",
    "ec2:AuthorizeSecurityGroupIngress",
    "ec2:CancelSpotInstanceRequests",
    "ec2:CreateNetworkInterface",
    "ec2:CreateSecurityGroup",
    "ec2:CreateTags",
    "ec2>DeleteNetworkInterface",
    "ec2>DeleteSecurityGroup",
    "ec2>DeleteTags",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeImages",
    "ec2:DescribeInstanceStatus",
    "ec2:DescribeInstances",
    "ec2:DescribeKeyPairs",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeNetworkAcls",
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribePrefixLists",
    "ec2:DescribeRouteTables",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSpotInstanceRequests",
    "ec2:DescribeSpotPriceHistory",
    "ec2:DescribeSubnets",
    "ec2:DescribeTags",
```

```
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcs",
        "ec2:DetachNetworkInterface",
        "ec2:ModifyImageAttribute",
        "ec2:ModifyInstanceAttribute",
        "ec2:RequestSpotInstances",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVolumes",
        "elasticmapreduce:TerminateJobFlows",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:ListClusters",
        "elasticmapreduce:RunJobFlow",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:AddTags",
        "elasticmapreduce:RemoveTags",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ModifyInstanceGroups",
        "elasticmapreduce:GetCluster",
        "elasticmapreduce:DescribeStep",
        "elasticmapreduce:AddJobFlowSteps",
        "elasticmapreduce:ListInstances",
        "iam:ListInstanceProfiles",
        "redshift:DescribeClusters"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "sns:GetTopicAttributes",
        "sns:Publish"
    ],
    "Resource": [
        "arn:aws:sns:us-west-1:111122223333:MyFirstSNSTopic",
        "arn:aws:sns:us-west-1:111122223333:MySecondSNSTopic",
        "arn:aws:sns:us-west-1:111122223333:AnotherSNSTopic"
    ]
}
```



```
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:ListMultipartUploads"
      ],
      "Resource": [
        "arn:aws:s3:::MyStagingS3Bucket",
        "arn:aws:s3:::MyLogsS3Bucket",
        "arn:aws:s3:::MyInputS3Bucket",
        "arn:aws:s3:::MyOutputS3Bucket",
        "arn:aws:s3:::AnotherRequiredS3Buckets"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectMetadata",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::MyStagingS3Bucket/*",
        "arn:aws:s3:::MyLogsS3Bucket/*",
        "arn:aws:s3:::MyInputS3Bucket/*",
        "arn:aws:s3:::MyOutputS3Bucket/*",
        "arn:aws:s3:::AnotherRequiredS3Buckets/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:Scan",
        "dynamodb:DescribeTable"
      ],
      "Resource": [
        "arn:aws:dynamodb:us-west-1:111122223333:table/MyFirstDynamoDBTable",
        "arn:aws:dynamodb:us-west-1:111122223333:table/MySecondDynamoDBTable",
        "arn:aws:dynamodb:us-west-1:111122223333:table/AnotherDynamoDBTable"
      ]
    },
    {
      "Effect": "Allow",
```

```

    "Action": [
      "rds:DescribeDBInstances"
    ],
    "Resource": [
      "arn:aws:rds:us-west-1:111122223333:db:MyFirstRdsDb",
      "arn:aws:rds:us-west-1:111122223333:db:MySecondRdsDb",
      "arn:aws:rds:us-west-1:111122223333:db:AnotherRdsDb"
    ]
  }
]
}

```

Política gerenciada padrão para a função de instância do EC2

O conteúdo de `AmazonEC2RoleforDataPipelineRole` é mostrado abaixo. Essa é a política gerenciada anexada à função de recurso padrão para AWS Data Pipeline e `DataPipelineDefaultResourceRole`. Ao definir uma função de recurso para seu pipeline, recomendamos que você comece com essa política de permissões e, em seguida, remova as permissões para ações de serviço de AWS que não são necessárias.

É exibida a versão 3 da política, que é a versão mais recente no momento da preparação deste artigo. Veja a versão mais recente da política usando o console do IAM.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudwatch:*",
      "datapipeline:*",
      "dynamodb:*",
      "ec2:Describe*",
      "elasticmapreduce:AddJobFlowSteps",
      "elasticmapreduce:Describe*",
      "elasticmapreduce:ListInstance*",
      "elasticmapreduce:ModifyInstanceGroups",
      "rds:Describe*",
      "redshift:DescribeClusters",
      "redshift:DescribeClusterSecurityGroups",
      "s3:*",
      "sdb:*",
      "sns:*",
      "sqs:*"
    ]
  }]
}

```

```
    ],  
    "Resource": ["*"]  
  }  
}
```

Criar perfis de IAM para AWS Data Pipeline e editar as permissões de função

Use o procedimento a seguir para criar uma função para AWS Data Pipeline usando o console do IAM. O processo consiste em duas etapas. Primeiro, você deve criar uma política de permissões para anexar à função. Em seguida, crie uma função e anexe a política a ela. Depois de criar uma função, você pode alterar as permissões da função ao anexar e desanexar políticas de permissões.

Note

Quando você cria funções para AWS Data Pipeline usar o console conforme descrito abaixo, o IAM cria e anexa as políticas de confiança adequadas que a função exige.

Para criar uma política de permissões para usar com uma função para AWS Data Pipeline

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Políticas e, em seguida, Criar política.
3. Escolha a guia JSON.
4. Se você estiver criando uma função de pipeline, copie e cole o conteúdo do exemplo de política em [Exemplo de política de permissões da função de pipeline](#), editando-o conforme apropriado para seus requisitos de segurança. Como alternativa, se você estiver criando uma função de instância de EC2 personalizada, faça o mesmo com o exemplo em [Política gerenciada padrão para a função de instância do EC2](#).
5. Escolha Review policy (Revisar política).
6. Insira um nome e uma descrição opcional, como, por exemplo, MyDataPipelineRolePolicy, e uma Descrição opcional e, então, selecione Criar política.
7. Anote o nome da política. Você precisa dele ao criar a função.

Para criar uma função do IAM para o AWS Data Pipeline

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.

2. No painel de navegação, selecione Funções e, então, selecione Criar função.
3. Em Selecione um caso de uso, selecione Pipeline de dados.
4. Em Selecione seu caso de uso, siga um destes procedimentos:
 - Selecione Data Pipeline para criar uma função de pipeline.
 - Selecione EC2 Role for Data Pipeline para criar uma função de recurso.
5. Escolha Next: Permissions (Próximo: permissões).
6. Se a política padrão para AWS Data Pipeline estiver listada, siga as etapas a seguir para criar a função e editá-la de acordo com as instruções do próximo procedimento. Caso contrário, insira o nome da política que você criou no procedimento acima e selecione-o na lista.
7. Selecione Próximo: Tags, insira as tags a serem adicionadas à função e selecione Próximo: Revisão.
8. Insira um nome para a função, como, por exemplo, MyDataPipelineRole, e uma Descrição opcional e, em seguida, selecione Criar função.

Para anexar ou desanexar uma política de permissões para um perfil do IAM para AWS Data Pipeline

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Funções
3. Na caixa de pesquisa, digite o nome da função que você deseja editar, como, por exemplo, DataPipelineDefaultRole ou MyDataPipelineRole e, então, selecione o Nome da função na lista.
4. Na guia Permissões, faça o seguinte:
 - Para desanexar uma política de permissões, em Políticas de permissões, selecione o botão remover na extremidade direita da entrada da política. Selecione Desanexar quando solicitado para confirmar.
 - Para anexar uma política que você criou anteriormente, selecione Anexar políticas. Na caixa de pesquisa, digite o nome da política que você deseja editar, selecione-o na lista e selecione Anexar política.

Alterar as funções de um pipeline existente

Se você quiser atribuir uma função de pipeline ou função de recurso diferente a um pipeline, você pode usar o editor de arquitetura no console do AWS Data Pipeline.

Para editar as funções atribuídas a um pipeline usando o console

1. Abra o console do AWS Data Pipeline em <https://console.aws.amazon.com/datapipeline/>.
2. Selecione o pipeline na lista e selecione Ações, Editar.
3. No painel direito do editor de arquitetura, selecione Outros.
4. Nas listas Função do Recurso e Função, selecione as funções para AWS Data Pipeline que você deseja atribuir e selecione Salvar.

Registro em log e monitoramento no AWS Data Pipeline

O AWS Data Pipeline é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um serviço da AWS no AWS Data Pipeline.

O CloudTrail captura as chamadas de API do AWS Data Pipeline como eventos. As chamadas capturadas incluem as chamadas do console do AWS Data Pipeline e as chamadas de código para as operações da API do AWS Data Pipeline. 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 Data Pipeline. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o AWS Data Pipeline, 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 Data Pipeline no CloudTrail

O CloudTrail é habilitado em sua conta da AWS quando ela é criada. Quando ocorre uma atividade no AWS Data Pipeline, essa atividade é registrada em um evento do CloudTrail com outros eventos de produtos da AWS em Event history (Histórico de eventos). Você pode visualizar, pesquisar e baixar os eventos recentes em sua conta da AWS. Para obter mais informações, consulte [Como visualizar eventos com o histórico de eventos do CloudTrail](#).

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos do AWS Data Pipeline, 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 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](#) e [receber arquivos de log do CloudTrail de várias contas](#)

Todas as ações do AWS Data Pipeline estão registradas em log pelo CloudTrail e documentadas no capítulo [Ações de referência da API do AWS Data Pipeline](#). Por exemplo, as chamadas para a ação `CreatePipeline` 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:

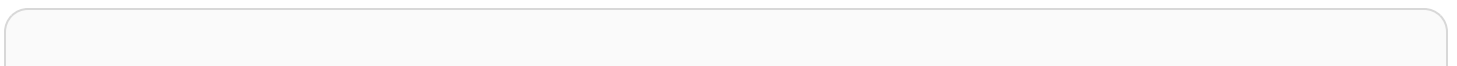
- Se a solicitação foi feita com credenciais de raiz ou de perfil do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte o [Elemento `userIdentity` do CloudTrail](#).

Noções básicas das entradas dos arquivos de log do AWS Data Pipeline

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 fonte 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 de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O seguinte exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a operação `CreatePipeline`:



```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::aws-account-id:role/role-name",
        "accountId": "role-account-id",
        "accessKeyId": "role-access-key"
      },
      "eventTime": "2014-11-13T19:15:15Z",
      "eventSource": "datapipeline.amazonaws.com",
      "eventName": "CreatePipeline",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "72.21.196.64",
      "userAgent": "aws-cli/1.5.2 Python/2.7.5 Darwin/13.4.0",
      "requestParameters": {
        "name": "testpipeline",
        "uniqueId": "sounique"
      },
      "responseElements": {
        "pipelineId": "df-06372391ZG65EXAMPLE"
      },
      "requestID": "65cbf1e8-6b69-11e4-8816-cfcbadd04c45",
      "eventID": "9f99dce0-0864-49a0-bffa-f72287197758",
      "eventType": "AwsApiCall",
      "recipientAccountId": "role-account-id"
    },
    ...additional entries
  ]
}
```

Resposta a incidentes no AWS Data Pipeline

A resposta a incidentes do AWS Data Pipeline é uma responsabilidade da AWS. A AWS tem uma política formal e documentada e um programa que rege a resposta a incidentes.

Problemas operacionais da AWS com impacto amplo são publicados no AWS Service Health Dashboard. Problemas operacionais também são publicados em contas individuais por meio do Personal Health Dashboard.

Validação de Compatibilidade para AWS Data Pipeline

O AWS Data Pipeline não está no escopo de nenhum programa de conformidade da AWS. Para obter uma lista dos serviços da AWS no escopo de programas de conformidade específicos, consulte [Serviços da AWS no escopo por programa de conformidade](#). Para obter informações gerais, consulte [Programas de conformidade da AWS](#).

Resiliência no AWS Data Pipeline

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, que são conectadas com baixa latência, throughput elevada e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas 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](#).

Segurança da infraestrutura no AWS Data Pipeline

Como um serviço gerenciado, o AWS Data Pipeline é protegido pelos procedimentos de segurança de rede global da AWS que estão descritos no whitepaper [Amazon Web Services: Overview of Security Processes](#).

Você usa chamadas de API publicadas pela AWS para acessar o AWS Data Pipeline por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos TLS 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Análise de vulnerabilidade e configuração no AWS Data Pipeline

A configuração e os controles de TI são uma responsabilidade compartilhada entre a AWS e você, nosso cliente. Para obter mais informações, consulte o [modelo de responsabilidade compartilhada da AWS](#).

Tutoriais

Os tutoriais a seguir conduzem você passo a passo pelo processo de criação e uso de pipelines com o AWS Data Pipeline.

Tutoriais

- [Processe dados usando a Amazon EMR com o Hadoop Streaming](#)
- [Copiar dados CSV entre buckets do Amazon S3 usando o AWS Data Pipeline](#)
- [Exportar dados do MySQL para o Amazon S3 usando a AWS Data Pipeline](#)
- [Copiar dados para o Amazon Redshift usando AWS Data Pipeline](#)

Processe dados usando a Amazon EMR com o Hadoop Streaming

Você pode usar AWS Data Pipeline para gerenciar seus EMR clusters da Amazon. Com isso, AWS Data Pipeline você pode especificar condições prévias que devem ser atendidas antes do lançamento do cluster (por exemplo, garantir que os dados atuais sejam enviados para o Amazon S3), um cronograma para executar repetidamente o cluster e a configuração do cluster a ser usada. O tutorial a seguir fornece o passo a passo para que você inicie um cluster simples.

Neste tutorial, você cria um pipeline para que um EMR cluster simples da Amazon execute uma tarefa preexistente do Hadoop Streaming fornecida pela Amazon EMR e envie uma SNS notificação à Amazon após a conclusão bem-sucedida da tarefa. Você usa o recurso de EMR cluster da Amazon fornecido por AWS Data Pipeline para essa tarefa. O aplicativo de amostra é chamado WordCount e também pode ser executado manualmente no EMR console da Amazon. Observe que os clusters gerados AWS Data Pipeline em seu nome são exibidos no EMR console da Amazon e são cobrados em sua AWS conta.

Objetos de pipeline

O pipeline usa os seguintes objetos:

[EmrActivity](#)

Define o trabalho a ser executado no pipeline (execute uma tarefa preexistente do Hadoop Streaming fornecida pela Amazon). EMR

[EmrCluster](#)

Recursos AWS Data Pipeline usados para realizar essa atividade.

Um cluster é um conjunto de EC2 instâncias da Amazon. AWS Data Pipeline inicia o cluster e, em seguida, o encerra após a conclusão da tarefa.

[Schedule](#)

Data e hora de início, e a duração dessa atividade. Se preferir, você pode especificar a data e a hora de término.

[SnsAlarm](#)

Envia uma SNS notificação da Amazon para o tópico que você especificar depois que a tarefa for concluída com sucesso.

Conteúdo

- [Antes de começar](#)
- [Iniciar um cluster usando a linha de comando](#)

Antes de começar

Certifique-se de que você concluiu as etapas a seguir.

- Conclua as tarefas em [Configurando para AWS Data Pipeline](#).
- (Opcional) Configure um VPC para o cluster e um grupo de segurança para VPC o.
- Crie um tópico para enviar notificações por e-mail e anote o tópico Amazon Resource Name (ARN). Para obter mais informações, consulte [Criar um tópico](#) no Guia de conceitos básicos do Amazon Simple Notification Service.

Iniciar um cluster usando a linha de comando

Se você executa regularmente um EMR cluster da Amazon para analisar registros da web ou realizar análises de dados científicos, você pode usá-lo AWS Data Pipeline para gerenciar seus EMR clusters da Amazon. Com AWS Data Pipeline, você pode especificar condições prévias que devem ser atendidas antes do lançamento do cluster (por exemplo, garantir que os dados de hoje sejam enviados para o Amazon S3). Este tutorial orienta você no lançamento de um cluster que pode ser

um modelo para um simples pipeline EMR baseado na Amazon ou como parte de um pipeline mais complexo.

Pré-requisitos

Antes de usar o CLI, você deve concluir as seguintes etapas:

1. Instale e configure uma interface de linha de comando (CLI). Para obter mais informações, consulte [Acessando AWS Data Pipeline](#).
2. Certifique-se de que as IAM funções tenham sido nomeadas `DataPipelineDefaultRole` e `DataPipelineDefaultResourceRole` existam. O AWS Data Pipeline console cria essas funções para você automaticamente. Se você não usou o AWS Data Pipeline console pelo menos uma vez, deverá criar essas funções manualmente. Para obter mais informações, consulte [Funções do IAM para o AWS Data Pipeline](#).

Tarefas

- [Criar o arquivo de definição de pipeline](#)
- [Fazer upload e ativar a definição do pipeline](#)
- [Monitorar as execuções do pipeline](#)

Criar o arquivo de definição de pipeline

O código a seguir é o arquivo de definição de pipeline para um EMR cluster simples da Amazon que executa uma tarefa de streaming existente do Hadoop fornecida pela Amazon. Esse aplicativo de amostra é chamado `WordCount` e você também pode executá-lo usando o EMR console da Amazon.

Copie este código em um arquivo de texto e salve-o como `MyEmrPipelineDefinition.json`. Você deve substituir o local do bucket do Amazon S3 pelo nome de um bucket do Amazon S3 que você possui. Você também deve substituir as datas de início e término. Para iniciar clusters imediatamente, `startDateTime` defina uma data para um dia no passado e `endDateTime` para um dia no futuro. AWS Data Pipeline em seguida, começa a lançar os clusters “vencidos” imediatamente, na tentativa de resolver o que considera um acúmulo de trabalho. Esse preenchimento significa que você não precisa esperar uma hora para AWS Data Pipeline iniciar seu primeiro cluster.

```
{
```

```
"objects": [  
  {  
    "id": "Hourly",  
    "type": "Schedule",  
    "startDateTime": "2012-11-19T07:48:00",  
    "endDateTime": "2012-11-21T07:48:00",  
    "period": "1 hours"  
  },  
  {  
    "id": "MyCluster",  
    "type": "EmrCluster",  
    "masterInstanceType": "m1.small",  
    "schedule": {  
      "ref": "Hourly"  
    }  
  },  
  {  
    "id": "MyEmrActivity",  
    "type": "EmrActivity",  
    "schedule": {  
      "ref": "Hourly"  
    },  
    "runsOn": {  
      "ref": "MyCluster"  
    },  
    "step": "/home/hadoop/contrib/streaming/hadoop-streaming.jar, -input, s3n://  
elasticmapreduce/samples/wordcount/input, -output, s3://myawsbucket/wordcount/  
output/#{@scheduledStartTime}, -mapper, s3n://elasticmapreduce/samples/wordcount/  
wordSplitter.py, -reducer, aggregate"  
  }  
]
```

Este pipeline tem três objetos:

- **Hourly**, que representa o agendamento do trabalho. Você pode definir uma programação como um dos campos em uma atividade. Quando você fizer isso, a atividade será executada de acordo com a programação, ou neste caso, de hora em hora.
- **MyCluster**, que representa o conjunto de EC2 instâncias da Amazon usadas para executar o cluster. Você pode especificar o tamanho e o número de EC2 instâncias a serem executadas como cluster. Se você não especificar o número de instâncias, o cluster será iniciado com duas, um nó principal e um nó de tarefa. Você pode especificar uma sub-rede para executar o cluster.

Você pode adicionar configurações adicionais ao cluster, como ações de bootstrap para carregar software adicional no fornecido pela AmazonEMR. AMI

- `MyEmrActivity`, que representa o cálculo para processar com o cluster. A Amazon EMR oferece suporte a vários tipos de clusters, incluindo streaming, cascading e Scripted Hive. O `runsOn` campo se refere novamente a `MyCluster`, usando isso como especificação para os fundamentos do cluster.

Fazer upload e ativar a definição do pipeline

Você deve fazer o upload da definição do pipeline e ativá-lo. Nos comandos de exemplo a seguir, substitua `pipeline_name` com uma etiqueta para seu funil e `pipeline_file` com o caminho totalmente qualificado para o arquivo de definição `.json` do pipeline.

AWS CLI

Para criar sua definição de pipeline e ativar seu pipeline, use o seguinte comando: [create-pipeline](#). Anote o ID do seu pipeline, pois você usará esse valor com a maioria dos CLI comandos.

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
  "pipelineId": "df-00627471S0VYZEXAMPLE"
}
```

Para fazer o upload da definição do pipeline, use o [put-pipeline-definition](#) comando a seguir.

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

Se o pipeline for validado com êxito, o campo `validationErrors` estará vazio. Você deve revisar todos os avisos.

Para ativar o pipeline, use o seguinte comando: [activate-pipeline](#).

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Você pode verificar se seu pipeline aparece na lista de pipeline usando o seguinte comando: [list-pipelines](#).

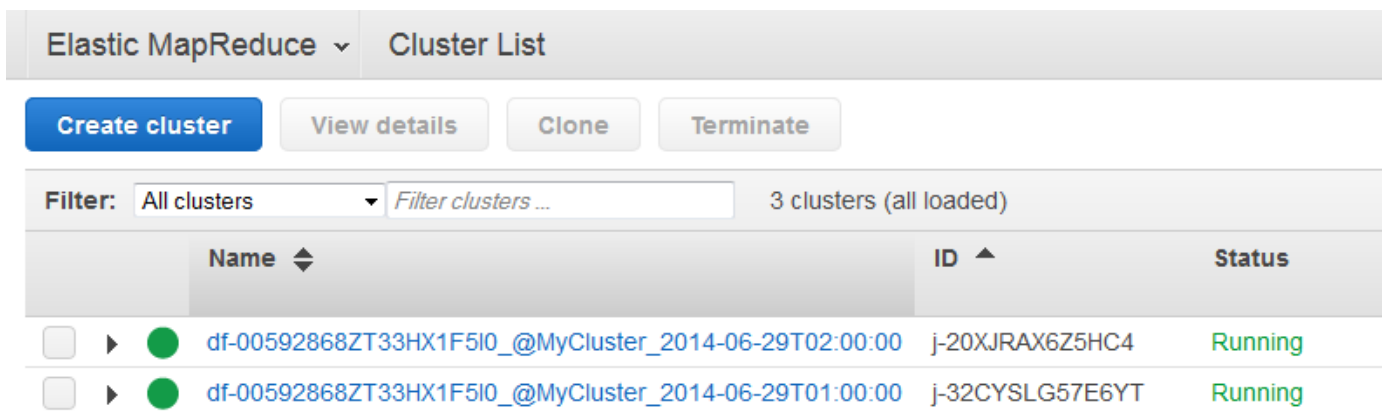
```
aws datapipeline list-pipelines
```

Monitorar as execuções do pipeline

Você pode visualizar os clusters lançados AWS Data Pipeline usando o EMR console da Amazon e você pode visualizar a pasta de saída usando o console do Amazon S3.

Para verificar o progresso dos clusters lançados pelo AWS Data Pipeline

1. Abra o EMR console da Amazon.
2. Os clusters que foram gerados por AWS Data Pipeline têm um nome formatado da seguinte forma: `<pipeline-identifier>_@<emr-cluster-name>_<launch-time>`.



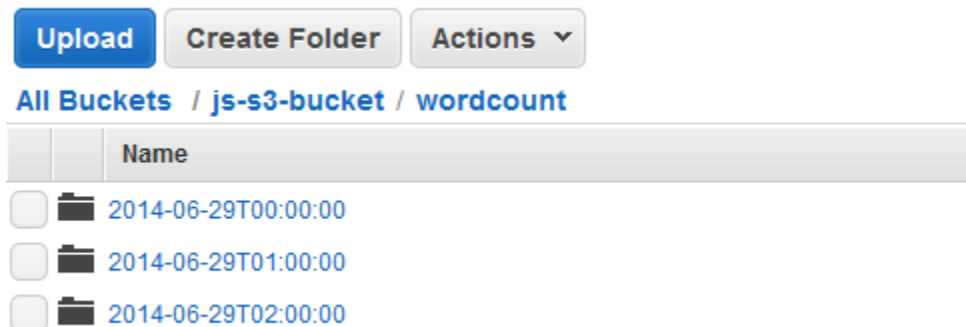
Elastic MapReduce ▾ Cluster List

Create cluster View details Clone Terminate

Filter: All clusters ▾ Filter clusters ... 3 clusters (all loaded)

| Name | ID | Status |
|---|-----------------|---------|
| df-00592868ZT33HX1F5I0_@MyCluster_2014-06-29T02:00:00 | j-20XJRAX6Z5HC4 | Running |
| df-00592868ZT33HX1F5I0_@MyCluster_2014-06-29T01:00:00 | j-32CYSLG57E6YT | Running |

3. Depois que uma das execuções for concluída, abra o console do Amazon S3 e verifique se a data e hora da pasta de saída existe e contém os resultados esperados do cluster.



Upload Create Folder Actions ▾

All Buckets / js-s3-bucket / wordcount

| Name |
|---------------------|
| 2014-06-29T00:00:00 |
| 2014-06-29T01:00:00 |
| 2014-06-29T02:00:00 |

Copiar dados CSV entre buckets do Amazon S3 usando o AWS Data Pipeline

Depois de ler [O que é AWS Data Pipeline?](#) e decidir que você deseja usar o AWS Data Pipeline para automatizar o movimento e a transformação dos seus dados, comece a criar os pipelines de dados.

Para ajudar você a entender como o AWS Data Pipeline funciona, mostraremos o passo a passo de uma tarefa simples.

Este tutorial orienta você no processo de criação de um pipeline de dados para copiar dados de um bucket do Amazon S3 para outro e, em seguida, enviar uma notificação do Amazon SNS após a conclusão com êxito da atividade de cópia. Para esta atividade de cópia, use uma instância do EC2 gerenciada pelo AWS Data Pipeline.

Objetos de pipeline

O pipeline usa os seguintes objetos:

[CopyActivity](#)

A atividade que o AWS Data Pipeline executa neste pipeline (cópia de dados CSV de um bucket do Amazon S3 para outro).

Important

Há limitações ao usar o formato de arquivo CSV com CopyActivity e S3DataNode. Para obter mais informações, consulte [CopyActivity](#).

[Schedule](#)

A data de início, hora e recorrência dessa atividade. Se preferir, você pode especificar a data e a hora de término.

[Ec2Resource](#)

O recurso (uma instância do EC2) que o AWS Data Pipeline utiliza para executar esta atividade.

[S3 DataNode](#)

Os nós de entrada e saída (buckets do Amazon S3) deste pipeline.

[SnsAlarm](#)

Ação que o AWS Data Pipeline precisa tomar quando as condições especificadas são atendidas (envio de notificações do Amazon SNS para um tópico após a conclusão bem-sucedida da tarefa).

Índice

- [Antes de começar](#)
- [Copiar dados CSV usando a linha de comando](#)

Antes de começar

Certifique-se de que você concluiu as etapas a seguir.

- Conclua as tarefas em [Configurando para AWS Data Pipeline](#).
- (Opcional) Configure uma VPC para a instância e um security group para a VPC.
- Crie um bucket do Amazon S3 como uma fonte de dados.

Para obter mais informações, consulte [Criar um bucket](#) no Guia do usuário do Amazon Simple Storage Service.

- Carregar seus dados para seu bucket do Amazon S3.

Para obter mais informações, consulte [Adicionar um objeto a um bucket](#) no Guia do Amazon Simple Storage Service.

- Crie outro bucket do Amazon S3 como um destino de dados
- Crie um tópico para envio de notificação por e-mail e anote o nome de recurso da Amazon (ARN) do tópico. Para obter mais informações, consulte [Criar um tópico](#) no Guia de conceitos básicos do Amazon Simple Notification Service.
- (Opcional) Este tutorial usa as políticas de função do IAM padrão criadas pelo AWS Data Pipeline. Se você preferir criar e configurar sua própria política de perfil do IAM e as relações de confiança, siga as instruções descritas em [Funções do IAM para o AWS Data Pipeline](#).

Copiar dados CSV usando a linha de comando

Você pode criar e usar pipelines para copiar dados de um bucket do Amazon S3 para outro.

Pré-requisitos

Antes de começar, é necessário concluir as seguintes etapas:

1. Instale e configure a Interface da linha de comando (CLI). Para obter mais informações, consulte [Acessando AWS Data Pipeline](#).
2. Certifique-se de que os perfis do IAM chamados DataPipelineDefaultRole e DataPipelineDefaultResourceRole existem. O console do AWS Data Pipeline cria essas funções

para você automaticamente. Se você não usou o console do AWS Data Pipeline pelo menos uma vez, deverá criar essas funções manualmente. Para obter mais informações, consulte [Funções do IAM para o AWS Data Pipeline](#).

Tarefas

- [Definir um pipeline no formato JSON](#)
- [Fazer upload e ativar a definição do pipeline](#)

Definir um pipeline no formato JSON

Este cenário de exemplo mostra como usar as definições do pipeline JSON e a CLI do AWS Data Pipeline para programar a cópia de dados entre dois buckets do Amazon S3 em um intervalo de tempo específico. Este é o arquivo JSON de definição de pipeline completo, seguido de uma explicação para cada uma das seções.

Note

Recomendamos que você use um editor de texto que possa ajudá-lo a verificar a sintaxe dos arquivos formatados com JSON e nomeie o arquivo usando a extensão de arquivo .json.

Para ficar mais claro, neste exemplo ignoraremos os campos opcionais e mostramos apenas os campos obrigatórios. O arquivo JSON de pipeline completo para este exemplo é:

```
{
  "objects": [
    {
      "id": "MySchedule",
      "type": "Schedule",
      "startDateTime": "2013-08-18T00:00:00",
      "endDateTime": "2013-08-19T00:00:00",
      "period": "1 day"
    },
    {
      "id": "S3Input",
      "type": "S3DataNode",
      "schedule": {
        "ref": "MySchedule"
      }
    }
  ]
}
```

```
    "filePath": "s3://example-bucket/source/inputfile.csv"
  },
  {
    "id": "S3Output",
    "type": "S3DataNode",
    "schedule": {
      "ref": "MySchedule"
    },
    "filePath": "s3://example-bucket/destination/outputfile.csv"
  },
  {
    "id": "MyEC2Resource",
    "type": "Ec2Resource",
    "schedule": {
      "ref": "MySchedule"
    },
    "instanceType": "m1.medium",
    "role": "DataPipelineDefaultRole",
    "resourceRole": "DataPipelineDefaultResourceRole"
  },
  {
    "id": "MyCopyActivity",
    "type": "CopyActivity",
    "runsOn": {
      "ref": "MyEC2Resource"
    },
    "input": {
      "ref": "S3Input"
    },
    "output": {
      "ref": "S3Output"
    },
    "schedule": {
      "ref": "MySchedule"
    }
  }
]
}
```

Schedule

O pipeline define uma programação com uma data de início e fim, além de um período para determinar com que frequência a atividade neste pipeline é executada.

```
{
  "id": "MySchedule",
  "type": "Schedule",
  "startDateTime": "2013-08-18T00:00:00",
  "endDateTime": "2013-08-19T00:00:00",
  "period": "1 day"
},
```

Nós de dados do Amazon S3

Em seguida, o componente de pipeline S3DataNode de entrada definirá um local para os arquivos de entrada. Nesse caso, o local de um bucket do Amazon S3. O componente S3DataNode de entrada é definido pelos seguintes campos:

```
{
  "id": "S3Input",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "filePath": "s3://example-bucket/source/inputfile.csv"
},
```

Id

O nome definido pelo usuário para o local de entrada (somente um rótulo para sua referência).

Tipo

O tipo de componente do pipeline, que é "S3DataNode" para corresponder com o local em que os dados residem, em um bucket do Amazon S3.

Schedule

Uma referência para o componente de programação que criamos nas linhas anteriores do arquivo JSON "MySchedule".

Path

O caminho para os dados associados ao nó de dados. A sintaxe de um nó de dados é determinada pelo seu tipo. Por exemplo, a sintaxe para um caminho do Amazon S3 segue uma sintaxe diferente que é apropriada para uma tabela de banco de dados.

Em seguida, o componente S3DataNode de saída definirá o local de destino para saída dos dados. Ele segue o mesmo formato que o componente S3DataNode de entrada, exceto pelo nome do componente e um caminho diferente para indicar o arquivo de destino.

```
{
  "id": "S3Output",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "filePath": "s3://example-bucket/destination/outputfile.csv"
},
```

Recurso

Esta é uma definição do recurso computacional que executa a operação de cópia. Neste exemplo, o AWS Data Pipeline deve criar automaticamente uma instância do EC2 para executar a tarefa de cópia e encerrar o recurso após a conclusão da tarefa. Os campos definidos aqui controlam a criação e a função da instância do EC2 que faz o trabalho. O EC2Resource é definido pelos seguintes campos:

```
{
  "id": "MyEC2Resource",
  "type": "Ec2Resource",
  "schedule": {
    "ref": "MySchedule"
  },
  "instanceType": "m1.medium",
  "role": "DataPipelineDefaultRole",
  "resourceRole": "DataPipelineDefaultResourceRole"
},
```

Id

O nome definido pelo usuário para a programação do pipeline, que é apenas um rótulo para sua referência.

Tipo

O tipo de recurso computacional para executar o trabalho. Nesse caso, uma instância do EC2. Existem outros tipos de recursos disponíveis, por exemplo, o tipo EmrCluster.

Schedule

A programação para criar este recurso computacional.

instanceType

O tamanho da instância do EC2 a ser criada. Certifique-se de configurar o tamanho da instância do EC2 que melhor corresponda à carga de trabalho que você deseja executar com o AWS Data Pipeline. Nesse caso, configuramos uma instância do EC2 m1.medium. Para obter mais informações sobre os diferentes tipos de instância e quando usar cada uma delas, consulte o tópico [Tipos de instância do Amazon EC2](http://aws.amazon.com/ec2/instance-types/) no site <http://aws.amazon.com/ec2/instance-types/>.

Função

O perfil do IAM da conta que acessa os recursos, como acesso ao bucket do Amazon S3 para recuperação de dados.

resourceRole

A função do IAM da conta que cria recursos, como criação e configuração de uma instância do EC2 em seu nome. Role e ResourceRole podem ser a mesma função, mas fornecem separadamente uma granularidade maior na sua configuração de segurança.

Atividade

A última seção no arquivo JSON é a definição da atividade que representa o trabalho a ser executado. Este exemplo usa CopyActivity para copiar dados de um arquivo CSV em um bucket <http://aws.amazon.com/ec2/instance-types/> para outro. O componente CopyActivity é definido pelos seguintes campos:

```
{
  "id": "MyCopyActivity",
  "type": "CopyActivity",
  "runsOn": {
    "ref": "MyEC2Resource"
  },
  "input": {
    "ref": "S3Input"
  },
  "output": {
    "ref": "S3Output"
  },
}
```

```
"schedule": {  
  "ref": "MySchedule"  
}  
}
```

Id

O nome definido pelo usuário para a atividade, que é apenas um rótulo para sua referência.

Tipo

O tipo de atividade a ser realizada, como `MyCopyActivity`.

runsOn

O recurso computacional que realiza o trabalho definido por essa atividade. Neste exemplo, fornecemos uma referência à instância do EC2 anteriormente definida. Usar o campo `runsOn` faz com que o AWS Data Pipeline crie a instância do EC2 para você. O campo `runsOn` indica que o recurso existe na infraestrutura da AWS, enquanto o valor `workerGroup` indica que você deseja usar seus próprios recursos locais para executar o trabalho.

Entrada

O local dos dados a serem copiados.

Resultado

Os dados do local de destino.

Schedule

A programação na qual esta atividade será executada.

Fazer upload e ativar a definição do pipeline

Você deve fazer o upload da definição do pipeline e ativá-lo. Nos comandos de exemplo a seguir, substitua *pipeline_name* por um rótulo para seu pipeline e *pipeline_file* pelo caminho totalmente qualificado para o arquivo `.json` de definição de pipeline.

AWS CLI

Para criar sua definição de pipeline e ativar seu pipeline, use o seguinte comando: [create-pipeline](#). Observe a ID do seu pipeline, pois você usará esse valor com a maioria dos comandos da CLI.

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
  "pipelineId": "df-00627471S0VYZEXAMPLE"
}
```

Atualize a definição do pipeline usando o seguinte comando: [put-pipeline-definition](#).

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

Se o pipeline for validado com êxito, o campo `validationErrors` estará vazio. Você deve revisar todos os avisos.

Para ativar o pipeline, use o seguinte comando: [activate-pipeline](#).

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Você pode verificar se seu pipeline aparece na lista de pipeline usando o seguinte comando: [list-pipelines](#).

```
aws datapipeline list-pipelines
```

Exportar dados do MySQL para o Amazon S3 usando a AWS Data Pipeline

Este tutorial orienta você no processo de criação de um pipeline de dados para copiar dados (linhas) de uma tabela no banco de dados MySQL para um arquivo CSV (valores separados por vírgulas) em um bucket do Amazon S3 e, em seguida, enviar uma notificação do Amazon SNS após a conclusão bem-sucedida da atividade de cópia. Para esta atividade de cópia, use uma instância do EC2 fornecida pelo AWS Data Pipeline.

Objetos de pipeline

O pipeline usa os seguintes objetos:

- [CopyActivity](#)
- [Ec2Resource](#)

- [MySQLDataNode](#)
- [S3 DataNode](#)
- [SnsAlarm](#)

Índice

- [Antes de começar](#)
- [Copiar dados do MySQL usando a linha de comando](#)

Antes de começar

Certifique-se de que você concluiu as etapas a seguir.

- Conclua as tarefas em [Configurando para AWS Data Pipeline](#).
- (Opcional) Configure uma VPC para a instância e um security group para a VPC.
- Crie um bucket do Amazon S3 como uma saída de dados.

Para obter mais informações, consulte [Criar um bucket](#) no Guia do usuário do Amazon Simple Storage Service.

- Crie e inicie uma instância de banco de dados MySQL como fonte de dados.

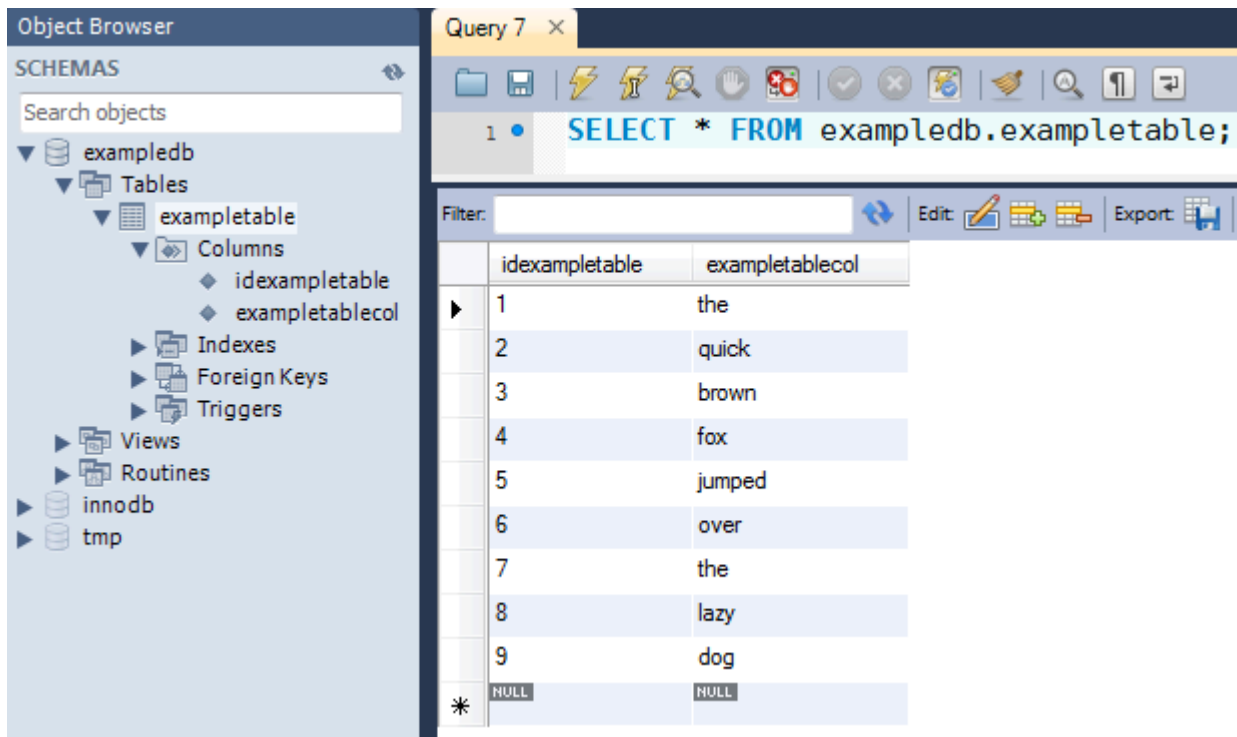
Para obter mais informações, consulte [Executar uma instância de banco de dados](#) no Guia de conceitos básicos do Amazon RDS. Depois que você criar uma instância do Amazon RDS, consulte [Criar uma tabela](#) na documentação do MySQL.

Note

Anote o nome do usuário e a senha que você usou para criar a instância do MySQL. Depois de iniciar sua instância de banco de dados MySQL, anote o endpoint da instância. Você precisará dessas informações posteriormente.

- Conecte-se à sua instância de banco de dados MySQL, crie uma tabela e adicione valores de dados de teste à tabela recém-criada.

Para fins de ilustração, criamos este tutorial usando uma tabela do MySQL com a configuração e os dados de amostra a seguir. A captura de tela a seguir é do MySQL Workbench 5.2 CE:



Para obter mais informações, consulte [Criar uma tabela](#) na documentação do MySQL e a [página do produto MySQL Workbench](#).

- Crie um tópico para envio de notificação por e-mail e anote o nome de recurso da Amazon (ARN) do tópico. Para obter mais informações, consulte [Criar um tópico](#) no Guia de conceitos básicos do Amazon Simple Notification Service.
- (Opcional) Este tutorial usa as políticas de função do IAM padrão criadas pelo AWS Data Pipeline. Se você preferir criar e configurar uma política de perfil do IAM e as relações de confiança, siga as instruções descritas em [Funções do IAM para o AWS Data Pipeline](#).

Copiar dados do MySQL usando a linha de comando

Você pode criar um pipeline para copiar dados de uma tabela do MySQL para um arquivo em um bucket do Amazon S3.

Pré-requisitos

Antes de começar, é necessário concluir as seguintes etapas:

1. Instale e configure a Interface da linha de comando (CLI). Para obter mais informações, consulte [Acessando AWS Data Pipeline](#).

2. Certifique-se de que os perfis do IAM chamados `DataPipelineDefaultRole` e `DataPipelineDefaultResourceRole` existem. O console do AWS Data Pipeline cria essas funções para você automaticamente. Se você não usou o console do AWS Data Pipeline pelo menos uma vez, deverá criar essas funções manualmente. Para obter mais informações, consulte [Funções do IAM para o AWS Data Pipeline](#).
3. Configure um bucket do Amazon S3 e uma instância do Amazon RDS. Para obter mais informações, consulte [Antes de começar](#).

Tarefas

- [Definir um pipeline no formato JSON](#)
- [Fazer upload e ativar a definição do pipeline](#)

Definir um pipeline no formato JSON

Este cenário de exemplo mostra como usar as definições de pipeline JSON e a CLI do AWS Data Pipeline para copiar dados (linhas) de uma tabela em um banco de dados MySQL para um arquivo CSV (valores separados por vírgulas) de um bucket do Amazon S3 em um intervalo especificado.

Este é o arquivo JSON de definição de pipeline completo, seguido de uma explicação para cada uma das seções.

Note

Recomendamos que você use um editor de texto que possa ajudá-lo a verificar a sintaxe dos arquivos formatados com JSON e nomeie o arquivo usando a extensão de arquivo `.json`.

```
{
  "objects": [
    {
      "id": "ScheduleId113",
      "startDateTime": "2013-08-26T00:00:00",
      "name": "My Copy Schedule",
      "type": "Schedule",
      "period": "1 Days"
    },
    {
      "id": "CopyActivityId112",
```

```

    "input": {
      "ref": "MySQLDataNodeId115"
    },
    "schedule": {
      "ref": "ScheduleId113"
    },
    "name": "My Copy",
    "runsOn": {
      "ref": "Ec2ResourceId116"
    },
    "onSuccess": {
      "ref": "ActionId1"
    },
    "onFail": {
      "ref": "SnsAlarmId117"
    },
    "output": {
      "ref": "S3DataNodeId114"
    },
    "type": "CopyActivity"
  },
  {
    "id": "S3DataNodeId114",
    "schedule": {
      "ref": "ScheduleId113"
    },
    "filePath": "s3://example-bucket/rds-output/output.csv",
    "name": "My S3 Data",
    "type": "S3DataNode"
  },
  {
    "id": "MySQLDataNodeId115",
    "username": "my-username",
    "schedule": {
      "ref": "ScheduleId113"
    },
    "name": "My RDS Data",
    "*password": "my-password",
    "table": "table-name",
    "connectionString": "jdbc:mysql://your-sql-instance-name.id.region-name.rds.amazonaws.com:3306/database-name",
    "selectQuery": "select * from #{table}",
    "type": "SqlDataNode"
  },

```

```
{
  "id": "Ec2ResourceId116",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "name": "My EC2 Resource",
  "role": "DataPipelineDefaultRole",
  "type": "Ec2Resource",
  "resourceRole": "DataPipelineDefaultResourceRole"
},
{
  "message": "This is a success message.",
  "id": "ActionId1",
  "subject": "RDS to S3 copy succeeded!",
  "name": "My Success Alarm",
  "role": "DataPipelineDefaultRole",
  "topicArn": "arn:aws:sns:us-east-1:123456789012:example-topic",
  "type": "SnsAlarm"
},
{
  "id": "Default",
  "scheduleType": "timeseries",
  "failureAndRerunMode": "CASCADE",
  "name": "Default",
  "role": "DataPipelineDefaultRole",
  "resourceRole": "DataPipelineDefaultResourceRole"
},
{
  "message": "There was a problem executing #{node.name} at for period
#{node.@scheduledStartTime} to #{node.@scheduledEndTime}",
  "id": "SnsAlarmId117",
  "subject": "RDS to S3 copy failed",
  "name": "My Failure Alarm",
  "role": "DataPipelineDefaultRole",
  "topicArn": "arn:aws:sns:us-east-1:123456789012:example-topic",
  "type": "SnsAlarm"
}
]
```

Nó de dados do MySQL

O componente de pipeline `MySQLDataNode` de entrada define um local para os dados de entrada. Nesse caso, uma instância do Amazon RDS. O componente `MySQLDataNode` de entrada é definido pelos seguintes campos:

```
{
  "id": "MySQLDataNodeId115",
  "username": "my-username",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "name": "My RDS Data",
  "*password": "my-password",
  "table": "table-name",
  "connectionString": "jdbc:mysql://your-sql-instance-name.id.region-
name.rds.amazonaws.com:3306/database-name",
  "selectQuery": "select * from #{table}",
  "type": "SqlDataNode"
},
```

Id

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

Nome de usuário

O nome de usuário da conta do banco de dados que tem permissão suficiente para recuperar dados da tabela do banco de dados. Substitua `my-username` pelo nome do seu usuário.

Schedule

Uma referência para o componente de programação que criamos nas linhas anteriores do arquivo JSON.

Nome

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

*Password

A senha da conta do banco de dados com o prefixo de asterisco para indicar que o AWS Data Pipeline precisa criptografar o valor da senha. Substitua `my-password` pela senha correta do seu usuário. O campo de senha é precedido pelo caractere especial asterisco. Para obter mais informações, consulte [Caracteres especiais](#).

Tabela

O nome da tabela do banco de dados que contém os dados a serem copiados. Substitua *table-name* pelo nome da tabela do seu banco de dados.

connectionString

A string de conexão JDBC para o objeto CopyActivity que se conectará ao banco de dados.

selectQuery

Uma consulta SQL SELECT válida que especifica quais dados da tabela do banco de dados serão copiados. `#{table}` é uma expressão que reutiliza o nome da tabela fornecido pela variável "table" nas linhas que precedem o arquivo JSON.

Tipo

O tipo `SqlDataNode`, que é uma instância do Amazon RDS usando o MySQL neste exemplo.

Note

O tipo `MySqlDataNode` está obsoleto. Você ainda pode usar `MySqlDataNode`. No entanto, recomendamos o uso de `SqlDataNode`.

Nó de dados do Amazon S3

Em seguida, o componente de pipeline `S3Output` definirá um local para o arquivo de saída. Nesse caso, um arquivo CSV no local de um bucket do Amazon S3. O componente `S3DataNode` de saída é definido pelos seguintes campos:

```
{
  "id": "S3DataNodeId114",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "filePath": "s3://example-bucket/rds-output/output.csv",
  "name": "My S3 Data",
  "type": "S3DataNode"
},
```

Id

O ID definido pelo usuário, que é apenas um rótulo para sua referência.

Schedule

Uma referência para o componente de programação que criamos nas linhas anteriores do arquivo JSON.

filePath

O caminho para os dados associados ao nó de dados, que é um arquivo de saída CSV neste exemplo.

Nome

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

Tipo

O tipo de objeto do pipeline, que é `S3DataNode`, para corresponder com o local em que os dados residem, em um bucket do Amazon S3.

Recurso

Esta é uma definição do recurso computacional que executa a operação de cópia. Neste exemplo, o AWS Data Pipeline deve criar automaticamente uma instância do EC2 para executar a tarefa de cópia e encerrar o recurso após a conclusão da tarefa. Os campos definidos aqui controlam a criação e a função da instância do EC2 que faz o trabalho. O `EC2Resource` é definido pelos seguintes campos:

```
{
  "id": "Ec2ResourceId116",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "name": "My EC2 Resource",
  "role": "DataPipelineDefaultRole",
  "type": "Ec2Resource",
  "resourceRole": "DataPipelineDefaultResourceRole"
},
```

Id

O ID definido pelo usuário, que é apenas um rótulo para sua referência.

Schedule

A programação para criar este recurso computacional.

Nome

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

Função

O perfil do IAM da conta que acessa os recursos, como acesso ao bucket do Amazon S3 para recuperação de dados.

Tipo

O tipo de recurso computacional para executar o trabalho. Nesse caso, uma instância do EC2. Existem outros tipos de recursos disponíveis, por exemplo, o tipo `EmrCluster`.

resourceRole

A função do IAM da conta que cria recursos, como criação e configuração de uma instância do EC2 em seu nome. `Role` e `ResourceRole` podem ser a mesma função, mas fornecem separadamente uma granularidade maior na sua configuração de segurança.

Atividade

A última seção no arquivo JSON é a definição da atividade que representa o trabalho a ser executado. Neste caso, usamos um componente `CopyActivity` para copiar dados de um arquivo em um bucket do Amazon S3 para outro arquivo. O componente `CopyActivity` é definido pelos seguintes campos:

```
{
  "id": "CopyActivityId112",
  "input": {
    "ref": "MySqlDataNodeId115"
  },
  "schedule": {
    "ref": "ScheduleId113"
  },
  "name": "My Copy",
  "runsOn": {
    "ref": "Ec2ResourceId116"
  },
  "onSuccess": {
```

```
    "ref": "ActionId1"
  },
  "onFail": {
    "ref": "SnsAlarmId117"
  },
  "output": {
    "ref": "S3DataNodeId114"
  },
  "type": "CopyActivity"
},
```

Id

O ID definido pelo usuário, que é apenas um rótulo para sua referência

Entrada

O local dos dados do MySQL a serem copiados

Schedule

A programação na qual esta atividade será executada

Nome

O nome definido pelo usuário, que é apenas um rótulo para sua referência

runsOn

O recurso computacional que realiza o trabalho definido por essa atividade. Neste exemplo, fornecemos uma referência à instância do EC2 anteriormente definida. Usar o campo `runsOn` faz com que o AWS Data Pipeline crie a instância do EC2 para você. O campo `runsOn` indica que o recurso existe na infraestrutura da AWS, enquanto o valor `workerGroup` indica que você deseja usar seus próprios recursos locais para executar o trabalho.

onSuccess

[SnsAlarm](#) a ser enviado se a atividade for concluída com sucesso

onFail

[SnsAlarm](#) a ser enviado se a atividade falhar

Resultado

Local do arquivo CSV de saída no Amazon S3

Tipo

O tipo da atividade a ser executada.

Fazer upload e ativar a definição do pipeline

Você deve fazer o upload da definição do pipeline e ativá-lo. Nos comandos de exemplo a seguir, substitua *pipeline_name* por um rótulo para seu pipeline e *pipeline_file* pelo caminho totalmente qualificado para o arquivo `.json` de definição de pipeline.

AWS CLI

Para criar sua definição de pipeline e ativar seu pipeline, use o seguinte comando: [create-pipeline](#). Observe a ID do seu pipeline, pois você usará esse valor com a maioria dos comandos da CLI.

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
  "pipelineId": "df-00627471S0VYZEXAMPLE"
}
```

Atualize a definição do pipeline usando o seguinte comando: [put-pipeline-definition](#).

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

Se o pipeline for validado com êxito, o campo `validationErrors` estará vazio. Você deve revisar todos os avisos.

Para ativar o pipeline, use o seguinte comando: [activate-pipeline](#).

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Você pode verificar se seu pipeline aparece na lista de pipeline usando o seguinte comando: [list-pipelines](#).

```
aws datapipeline list-pipelines
```

Copiar dados para o Amazon Redshift usando AWS Data Pipeline

Este tutorial orientará você no processo de criação de um pipeline que move periodicamente dados do Amazon S3 para o Amazon Redshift usando o modelo Copiar para o Redshift no console do AWS Data Pipeline ou um arquivo de definição de pipeline com a CLI do AWS Data Pipeline.

O Amazon S3 é um web service que permite o armazenamento de dados na nuvem. Para obter mais detalhes, consulte o [Manual do usuário do Amazon Simple Storage Service](#).

O Amazon Redshift é um serviço de data warehouse na nuvem. Para obter mais informações, consulte o [Guia de gerenciamento do Amazon Redshift](#).

Este tutorial tem vários pré-requisitos. Depois de concluir as etapas a seguir, você poderá continuar o tutorial usando o console ou a CLI.

Índice

- [Antes de começar: configurar as opções COPY e carregar dados](#)
- [Configurar pipeline, criar um grupo de segurança e criar um cluster do Amazon Redshift](#)
- [Copiar dados para o Amazon Redshift usando a linha de comando](#)

Antes de começar: configurar as opções COPY e carregar dados

Antes de copiar dados para o Amazon Redshift no AWS Data Pipeline, verifique se você pode:

- Carregar dados do Amazon S3.
- Configure a atividade COPY no Amazon Redshift.

Assim que você tiver essas opções funcionando e concluir com êxito um carregamento de dados, transfira essas opções para o AWS Data Pipeline, para fazer a cópia dentro dele.

Para opções COPY, consulte [COPY](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Para obter informações sobre as etapas para carregar dados do Amazon S3, consulte [Carregamento de dados do Amazon S3](#) no Guia do desenvolvedor do banco de dados do Amazon Redshift.

Por exemplo, o seguinte comando SQL no Amazon Redshift cria uma nova tabela chamada LISTING e copia dados de exemplo de um bucket disponível publicamente no Amazon S3.

Substitua o <iam-role-arn> e a região pelos seus próprios.

Para obter mais detalhes sobre este exemplo, consulte [Carregamento de dados de exemplo do Amazon S3](#) no Guia de conceitos básicos do Amazon Redshift.

```
create table listing(  
  listid integer not null distkey,  
  sellerid integer not null,  
  eventid integer not null,  
  dateid smallint not null sortkey,  
  numtickets smallint not null,  
  priceperticket decimal(8,2),  
  totalprice decimal(8,2),  
  listtime timestamp);  
  
copy listing from 's3://awssampleduswest2/ticket/listings_pipe.txt'  
credentials 'aws_iam_role=<iam-role-arn>'  
delimiter '|' region 'us-west-2';
```

Configurar pipeline, criar um grupo de segurança e criar um cluster do Amazon Redshift

Para se preparar para o tutorial

1. Conclua as tarefas em [Configurando para AWS Data Pipeline](#).
2. Crie um grupo de segurança.
 - a. Abra o console do Amazon EC2.
 - b. No painel de navegação, clique em Security Groups.
 - c. Clique em Create Security Group.
 - d. Especifique um nome e uma descrição para o grupo de segurança.
 - e. [EC2-Classic] Selecione No VPC para VPC.
 - f. [EC2-VPC] Selecione o ID da sua VPC para VPC.
 - g. Clique em Criar.
3. [EC2-Classic] Crie um grupo de segurança de cluster do Amazon Redshift e especifique o grupo de segurança do Amazon EC2.
 - a. Abra o console do Amazon Redshift.

- b. No painel de navegação, clique em Security Groups.
 - c. Clique em Create Cluster Security Group.
 - d. Na caixa de diálogo Create Cluster Security Group, especifique um nome e forneça uma descrição para o security group do cluster.
 - e. Clique no nome do novo security group do cluster.
 - f. Clique em Add Connection Type.
 - g. Na caixa de diálogo Add Connection Type, selecione EC2 Security Group em Connection Type, selecione o security group que criou em EC2 Security Group Name e, em seguida, clique em Authorize.
4. [EC2-VPC] Crie um grupo de segurança de cluster do Amazon Redshift e especifique o grupo de segurança da VPC.
- a. Abra o console do Amazon EC2.
 - b. No painel de navegação, clique em Security Groups.
 - c. Clique em Create Security Group.
 - d. Na caixa de diálogo Create Security Group, especifique um nome e forneça uma descrição para o security group e, em seguida, selecione o ID da sua VPC em VPC.
 - e. Clique em Add Rule. Especifique tipo, protocolo e alcance de porta, e comece a digitar o ID do security group em Source. Selecione o security group que você criou na segunda etapa.
 - f. Clique em Criar.

5. A seguir, veja um resumo das etapas:

Se você tem um cluster existente no Amazon Redshift, anote a ID do cluster.

Para criar um cluster e carregar dados de exemplo, siga as etapas de [Conceitos básicos do Amazon Redshift](#). Para obter mais informações sobre a criação de clusters, consulte [Criar um cluster](#) no Guia de gerenciamento do Amazon Redshift.

- a. Abra o console do Amazon Redshift.
- b. Clique em Launch Cluster.
- c. Forneça os detalhes necessários para o seu cluster e clique em Continue.
- d. Informe a configuração do nó e clique em Continue.
- e. Na página de informações de configuração adicionais, selecione o security group do cluster **que você criou e clique em Continue**.

- f. Revise as especificações do seu cluster e clique em Launch Cluster.

Copiar dados para o Amazon Redshift usando a linha de comando

Este tutorial demonstra como copiar dados do Amazon S3 para o Amazon Redshift. Você pode criar uma nova tabela no Amazon Redshift e, em seguida, usar o AWS Data Pipeline para transferir dados para ela a partir de um bucket público do Amazon S3, que contenha exemplos de dados de entrada no formato CSV. Os logs são salvos em um bucket do Amazon S3 que você possui.

O Amazon S3 é um web service que permite o armazenamento de dados na nuvem. Para obter mais detalhes, consulte o [Manual do usuário do Amazon Simple Storage Service](#). O Amazon Redshift é um serviço de data warehouse na nuvem. Para obter mais informações, consulte o [Guia de gerenciamento do Amazon Redshift](#).

Pré-requisitos

Antes de começar, é necessário concluir as seguintes etapas:

1. Instale e configure a Interface da linha de comando (CLI). Para obter mais informações, consulte [Acessando AWS Data Pipeline](#).
2. Certifique-se de que os perfis do IAM chamados DataPipelineDefaultRole e DataPipelineDefaultResourceRole existem. O console do AWS Data Pipeline cria essas funções para você automaticamente. Se você não usou o console do AWS Data Pipeline pelo menos uma vez, deverá criar essas funções manualmente. Para obter mais informações, consulte [Funções do IAM para o AWS Data Pipeline](#).
3. Configure o comando COPY no Amazon Redshift, pois você precisará ter essas mesmas opções funcionando ao fazer a cópia no AWS Data Pipeline. Para obter mais informações, consulte [Antes de começar: configurar as opções COPY e carregar dados](#).
4. Configure um banco de dados do Amazon Redshift. Para obter mais informações, consulte [Configurar pipeline, criar um grupo de segurança e criar um cluster do Amazon Redshift](#).

Tarefas

- [Definir um pipeline no formato JSON](#)
- [Fazer upload e ativar a definição do pipeline](#)

Definir um pipeline no formato JSON

Este cenário de exemplo mostra como copiar dados de um bucket do Amazon S3 para o Amazon Redshift.

Este é o arquivo JSON de definição de pipeline completo, seguido de uma explicação para cada uma das seções. Recomendamos que você use um editor de texto que possa ajudá-lo a verificar a sintaxe dos arquivos formatados com JSON e nomeie o arquivo usando a extensão de arquivo `.json`.

```
{
  "objects": [
    {
      "id": "CSVId1",
      "name": "DefaultCSV1",
      "type": "CSV"
    },
    {
      "id": "RedshiftDatabaseId1",
      "databaseName": "dbname",
      "username": "user",
      "name": "DefaultRedshiftDatabase1",
      "*password": "password",
      "type": "RedshiftDatabase",
      "clusterId": "redshiftclusterId"
    },
    {
      "id": "Default",
      "scheduleType": "timeseries",
      "failureAndRerunMode": "CASCADE",
      "name": "Default",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "id": "RedshiftDataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "tableName": "orders",
      "name": "DefaultRedshiftDataNode1",
      "createTableSql": "create table StructuredLogs (requestBeginTime CHAR(30)
PRIMARY KEY DISTKEY SORTKEY, requestEndTime CHAR(30), hostname CHAR(100), requestDate
varchar(20));",

```



```

    "type": "RedshiftDataNode",
    "database": {
      "ref": "RedshiftDatabaseId1"
    }
  },
  {
    "id": "Ec2ResourceId1",
    "schedule": {
      "ref": "ScheduleId1"
    },
    "securityGroups": "MySecurityGroup",
    "name": "DefaultEc2Resource1",
    "role": "DataPipelineDefaultRole",
    "logUri": "s3://myLogs",
    "resourceRole": "DataPipelineDefaultResourceRole",
    "type": "Ec2Resource"
  },
  {
    "id": "ScheduleId1",
    "startDateTime": "yyyy-mm-ddT00:00:00",
    "name": "DefaultSchedule1",
    "type": "Schedule",
    "period": "period",
    "endDateTime": "yyyy-mm-ddT00:00:00"
  },
  {
    "id": "S3DataNodeId1",
    "schedule": {
      "ref": "ScheduleId1"
    },
    "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
    "name": "DefaultS3DataNode1",
    "dataFormat": {
      "ref": "CSVId1"
    },
    "type": "S3DataNode"
  },
  {
    "id": "RedshiftCopyActivityId1",
    "input": {
      "ref": "S3DataNodeId1"
    },
    "schedule": {
      "ref": "ScheduleId1"
    }
  }
}

```

```
    },
    "insertMode": "KEEP_EXISTING",
    "name": "DefaultRedshiftCopyActivity1",
    "runsOn": {
      "ref": "Ec2ResourceId1"
    },
    "type": "RedshiftCopyActivity",
    "output": {
      "ref": "RedshiftDataNodeId1"
    }
  }
]
}
```

Para obter mais informações sobre esses objetos, consulte a documentação a seguir.

Objetos

- [Nós de dados](#)
- [Recurso](#)
- [Atividade](#)

Nós de dados

Este exemplo usa um nó de dados de entrada, um nó de dados de saída e um banco de dados.

Nó de dados de entrada

O componente de pipeline `S3DataNode` de entrada define o local dos dados de entrada no Amazon S3 e o formato dos dados de entrada. Para obter mais informações, consulte [S3 DataNode](#).

Esse componente de entrada é definido pelos seguintes campos:

```
{
  "id": "S3DataNodeId1",
  "schedule": {
    "ref": "ScheduleId1"
  },
  "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
  "name": "DefaultS3DataNode1",
  "dataFormat": {
```

```
    "ref": "CSVId1"
  },
  "type": "S3DataNode"
},
```

id

O ID definido pelo usuário, que é apenas um rótulo para sua referência.

schedule

Uma referência para o componente de programação.

filePath

O caminho para os dados associados ao nó de dados, que é um arquivo de entrada CSV neste exemplo.

name

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

dataFormat

Uma referência para o formato de dados da atividade a ser processada.

Nó de dados de saída

O componente do pipeline `RedshiftDataNode` de saída define um local para os dados de saída. Neste caso, uma tabela em um banco de dados do Amazon Redshift. Para obter mais informações, consulte [RedshiftDataNode](#). Esse componente de saída é definido pelos seguintes campos:

```
{
  "id": "RedshiftDataNodeId1",
  "schedule": {
    "ref": "ScheduleId1"
  },
  "tableName": "orders",
  "name": "DefaultRedshiftDataNode1",
  "createTableSql": "create table StructuredLogs (requestBeginTime CHAR(30) PRIMARY KEY DISTKEY SORTKEY, requestEndTime CHAR(30), hostname CHAR(100), requestDate varchar(20));",
  "type": "RedshiftDataNode",
```

```
"database": {  
  "ref": "RedshiftDatabaseId1"  
}  
},
```

id

O ID definido pelo usuário, que é apenas um rótulo para sua referência.

schedule

Uma referência para o componente de programação.

tableName

O nome da tabela do Amazon Redshift.

name

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

createTableSql

Uma expressão SQL para criar a tabela no banco de dados.

database

Uma referência ao banco de dados do Amazon Redshift.

Banco de dados

O componente `RedshiftDatabase` é definido pelos seguintes campos. Para obter mais informações, consulte [RedshiftDatabase](#).

```
{  
  "id": "RedshiftDatabaseId1",  
  "databaseName": "dbname",  
  "username": "user",  
  "name": "DefaultRedshiftDatabase1",  
  "*password": "password",  
  "type": "RedshiftDatabase",  
  "clusterId": "redshiftclusterId"  
},
```

id

O ID definido pelo usuário, que é apenas um rótulo para sua referência.

databaseName

O nome do banco de dados lógico.

username

O nome de usuário para se conectar ao banco de dados.

name

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

password

A senha para se conectar ao banco de dados.

clusterId

O ID do cluster do Redshift.

Recurso

Esta é uma definição do recurso computacional que executa a operação de cópia. Neste exemplo, o AWS Data Pipeline deve criar automaticamente uma instância do EC2 para executar a tarefa de cópia e encerrar a instância após a conclusão da tarefa. Os campos definidos aqui controlam a criação e a função da instância que faz o trabalho. Para obter mais informações, consulte [Ec2Resource](#).

O Ec2Resource é definido pelos seguintes campos:

```
{
  "id": "Ec2ResourceId1",
  "schedule": {
    "ref": "ScheduleId1"
  },
  "securityGroups": "MySecurityGroup",
  "name": "DefaultEc2Resource1",
  "role": "DataPipelineDefaultRole",
  "logUri": "s3://myLogs",
```

```
"resourceRole": "DataPipelineDefaultResourceRole",  
"type": "Ec2Resource"  
},
```

id

O ID definido pelo usuário, que é apenas um rótulo para sua referência.

schedule

A programação para criar este recurso computacional.

securityGroups

O security group a ser usado nas instâncias do grupo de recursos.

name

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

role

O perfil do IAM da conta que acessa os recursos, como acesso ao bucket do Amazon S3 para recuperação de dados.

logUri

O caminho de destino do Amazon S3 para fazer backup dos logs do Task Runner a partir do Ec2Resource.

resourceRole

A função do IAM da conta que cria recursos, como criação e configuração de uma instância do EC2 em seu nome. Role e ResourceRole podem ser a mesma função, mas fornecem separadamente uma granularidade maior na sua configuração de segurança.

Atividade

A última seção no arquivo JSON é a definição da atividade que representa o trabalho a ser executado. Neste caso, usamos um componente RedshiftCopyActivity para copiar dados do Amazon S3 para o Amazon Redshift. Para obter mais informações, consulte [RedshiftCopyActivity](#).

O componente RedshiftCopyActivity é definido pelos seguintes campos:

```
{
  "id": "RedshiftCopyActivityId1",
  "input": {
    "ref": "S3DataNodeId1"
  },
  "schedule": {
    "ref": "ScheduleId1"
  },
  "insertMode": "KEEP_EXISTING",
  "name": "DefaultRedshiftCopyActivity1",
  "runsOn": {
    "ref": "Ec2ResourceId1"
  },
  "type": "RedshiftCopyActivity",
  "output": {
    "ref": "RedshiftDataNodeId1"
  }
},
```

id

O ID definido pelo usuário, que é apenas um rótulo para sua referência.

input

Uma referência para o arquivo de origem do Amazon S3.

schedule

A programação na qual esta atividade será executada.

insertMode

O tipo de inserção (KEEP_EXISTING, OVERWRITE_EXISTING ou TRUNCATE).

name

O nome definido pelo usuário, que é apenas um rótulo para sua referência.

runsOn

O recurso computacional que realiza o trabalho definido por essa atividade.

output

Uma referência à tabela de destino do Amazon Redshift.

Fazer upload e ativar a definição do pipeline

Você deve fazer o upload da definição do pipeline e ativá-lo. Nos comandos de exemplo a seguir, substitua *pipeline_name* por um rótulo para seu pipeline e *pipeline_file* pelo caminho totalmente qualificado para o arquivo `.json` de definição de pipeline.

AWS CLI

Para criar sua definição de pipeline e ativar seu pipeline, use o seguinte comando: [create-pipeline](#). Observe a ID do seu pipeline, pois você usará esse valor com a maioria dos comandos da CLI.

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
  "pipelineId": "df-00627471S0VYZEXAMPLE"
}
```

Atualize a definição do pipeline usando o seguinte comando: [put-pipeline-definition](#).

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

Se o pipeline for validado com êxito, o campo `validationErrors` estará vazio. Você deve revisar todos os avisos.

Para ativar o pipeline, use o seguinte comando: [activate-pipeline](#).

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Você pode verificar se seu pipeline aparece na lista de pipeline usando o seguinte comando: [list-pipelines](#).

```
aws datapipeline list-pipelines
```


Expressões e funções do pipeline

Esta seção explica a sintaxe para o uso de expressões e funções nos pipelines, incluindo os tipos de dados associados.

Tipos de dados simples

Os tipos de dados a seguir podem ser definidos como valores de campo.

Tipos

- [DateTime](#)
- [Numérico](#)
- [Referências de objeto](#)
- [Período](#)
- [Segmento](#)

DateTime

O AWS Data Pipeline oferece suporte somente à data e à hora expressas no formato “YYYY-MM-DDTHH:MM:SS” em UTC/GMT. O exemplo a seguir define o campo `startDateTime` de um objeto `Schedule` como `1/15/2012, 11:59 p.m.`, no fuso horário UTC/GMT.

```
"startDateTime" : "2012-01-15T23:59:00"
```

Numérico

O AWS Data Pipeline oferece suporte a números inteiros e valores de ponto flutuante.

Referências de objeto

Um objeto na definição do pipeline. Ele pode ser o objeto atual, o nome de um objeto definido em outro lugar no pipeline ou um objeto que lista o objeto atual em um campo, referenciado pela palavra-chave `node`. Para ter mais informações sobre o `node`, consulte [Referenciar campos e objetos](#). Para obter mais informações sobre os tipos de objetos de pipeline, consulte [Referência de objeto de pipeline](#).

Período

Indica a frequência com que um evento programado deve ser executado. Expresso no formato "N [years|months|weeks|days|hours|minutes]", em que N é um valor inteiro positivo.

O período mínimo é de 15 minutos, e o máximo é de 3 anos.

O exemplo a seguir define o campo `period` do objeto `Schedule` como "3 hours". Isso cria uma programação que é executada a cada três horas.

```
"period" : "3 hours"
```

Segmento

Valores de string padrão. As strings precisam estar entre aspas duplas ("). Você pode usar a barra invertida (\) nos caracteres de escape em uma string. Não há suporte para strings de várias linhas.

Veja a seguir exemplos de valores de string válidos para o campo `id`.

```
"id" : "My Data Object"
```

```
"id" : "My \"Data\" Object"
```

As strings também podem conter expressões avaliadas como valores de string. Elas são inseridas na string e são delimitadas com "#{}" e "}". O exemplo a seguir usa uma expressão para inserir o nome do objeto atual em um caminho.

```
"filePath" : "s3://myBucket/#{name}.csv"
```

Para obter mais informações sobre como usar expressões, consulte [Referenciar campos e objetos](#) e [Avaliação de expressões](#).

Expressões

Com as expressões, é possível compartilhar um valor nos objetos relacionados. As expressões são processadas pelo serviço web do AWS Data Pipeline no tempo de execução, o que garante que todas elas sejam substituídas pelo valor da expressão.

As expressões são delimitadas por "{" e "}". Você pode usar uma expressão em qualquer objeto de definição de pipeline em que uma string é válida. Se um slot for uma referência ou destes tipos: ID, NAME, TYPE ou SPHERE, o valor dele não será avaliado nem usado textualmente.

A expressão a seguir chama uma das funções do AWS Data Pipeline. Para obter mais informações, consulte [Avaliação de expressões](#).

```
#{format(myDateTime, 'YYYY-MM-dd hh:mm:ss')}
```

Referenciar campos e objetos

As expressões podem usar campos do objeto atual em que a expressão existe ou campos de outro objeto vinculado por uma referência.

Um slot consiste em uma data de criação seguida pelo horário de criação do objeto, como @S3BackupLocation_2018-01-31T11:05:33.

Você também pode fazer referência ao ID do slot exato especificado na definição do pipeline, como o ID do slot do local de backup do Amazon S3. Para fazer referência ao ID do slot, use #{parent.@id}.

No exemplo a seguir, o campo filePath faz referência ao campo id no mesmo objeto para formar um nome de arquivo. O valor de filePath é avaliado para "s3://mybucket/ExampleDataNode.csv".

```
{
  "id" : "ExampleDataNode",
  "type" : "S3DataNode",
  "schedule" : {"ref" : "ExampleSchedule"},
  "filePath" : "s3://mybucket/#{parent.@id}.csv",
  "precondition" : {"ref" : "ExampleCondition"},
  "onFail" : {"ref" : "FailureNotify"}
}
```

Para usar um campo que existe em outro objeto vinculado por uma referência, use a palavra-chave node. Essa palavra-chave só está disponível com objetos de alarme e pré-condição.

Continuando com o exemplo anterior, uma expressão em SnsAlarm pode fazer referência ao intervalo de data e de hora em Schedule, pois S3DataNode faz referência a ambas.

Especificamente, o campo `message` de `FailureNotify` pode usar os campos de tempo de execução `@scheduledStartTime` e `@scheduledEndTime` de `ExampleSchedule`, pois o campo `onFail` do `ExampleDataNode` faz referência a `FailureNotify` e seu respectivo campo `schedule` faz referência a `ExampleSchedule`.

```
{
  "id" : "FailureNotify",
  "type" : "SnsAlarm",
  "subject" : "Failed to run pipeline component",
  "message": "Error for interval
#{node.@scheduledStartTime}..#{node.@scheduledEndTime}.",
  "topicArn":"arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic"
},
```

Note

Você pode criar pipelines com dependências, por exemplo, tarefas no seu pipeline que dependem do trabalho de outros sistemas ou de outras tarefas. Se o pipeline exigir determinados recursos, adicione essas dependências a ele usando precondições associadas a nós de dados e a tarefas. Isso faz com que os pipelines sejam depurados com mais facilidade e sejam mais resilientes. Além disso, mantenha suas dependências em um único pipeline sempre que possível, pois é difícil solucionar problemas em entre vários pipelines.

Expressões aninhadas

O AWS Data Pipeline permite o uso de valores aninhados para criar expressões mais complexas. Por exemplo, para executar um cálculo de tempo (subtrair 30 minutos de `scheduledStartTime`) e formatar o resultado para usar em uma definição de pipeline, você pode usar a seguinte expressão em uma atividade:

```
#{format(minusMinutes(@scheduledStartTime,30),'YYYY-MM-dd hh:mm:ss')}
```

e usar o prefixo `node` se a expressão for parte de um `SnsAlarm` ou de uma precondição:

```
#{format(minusMinutes(node.@scheduledStartTime,30),'YYYY-MM-dd hh:mm:ss')}
```

Listas

As expressões podem ser avaliadas em listas e em funções nas listas. Por exemplo, suponha que uma lista seja definida da seguinte maneira: `"myList":["one", "two"]`. Se essa lista for usada na expressão `#{'this is ' + myList}`, ela será avaliada como `["this is one", "this is two"]`. Se você tiver duas listas, o Data Pipeline as nivelará na avaliação. Por exemplo, se `myList1` for definida como `[1,2]` e `myList2` como `[3,4]`, a expressão `[#{myList1}, #{myList2}]` será avaliada como `[1,2,3,4]`.

Expressão de nó

O AWS Data Pipeline usa a expressão `#{node.*}` em `SnsAlarm` ou `PreCondition` para referência inversa ao objeto principal de um componente do pipeline. Como `SnsAlarm` e `PreCondition` são referenciados a partir de uma atividade ou um recurso sem referência inversa, `node` fornece uma forma consultar o indicador. Por exemplo, a definição do pipeline a seguir demonstra como uma notificação de falha pode usar o `node` para fazer referência ao nó principal, neste caso `ShellCommandActivity`, e incluir as horas de início e término programadas desse nó principal na mensagem do `SnsAlarm`. A referência `scheduledStartTime` em `ShellCommandActivity` não requer o prefixo `node`, pois `scheduledStartTime` faz referência própria.

Note

O sinal @ (arroba) que precede os campos indica que eles são campos de tempo de execução.

```
{
  "id" : "ShellOut",
  "type" : "ShellCommandActivity",
  "input" : {"ref" : "HourlyData"},
  "command" : "/home/userName/xxx.sh #{@scheduledStartTime} #{@scheduledEndTime}",
  "schedule" : {"ref" : "HourlyPeriod"},
  "stderr" : "/tmp/stderr:#{@scheduledStartTime}",
  "stdout" : "/tmp/stdout:#{@scheduledStartTime}",
  "onFail" : {"ref" : "FailureNotify"},
},
{
  "id" : "FailureNotify",
  "type" : "SnsAlarm",
```

```

"subject" : "Failed to run pipeline component",
"message": "Error for interval
#{node.@scheduledStartTime}..#{node.@scheduledEndTime}.",
"topicArn": "arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic"
},

```

O AWS Data Pipeline oferece suporte a referências transitivas para campos definidos pelo usuário, mas não para campos de tempo de execução. Uma referência transitiva é uma referência entre dois componentes de pipeline que dependem de outro componente de pipeline como intermediário. O exemplo a seguir mostra uma referência a um campo transitivo definido por usuário e uma referência a um campo não transitivo de tempo de execução, ambos válidos. Para obter mais informações, consulte [Campos definidos pelo usuário](#).

```

{
  "name": "DefaultActivity1",
  "type": "CopyActivity",
  "schedule": {"ref": "Once"},
  "input": {"ref": "s3nodeOne"},
  "onSuccess": {"ref": "action"},
  "workerGroup": "test",
  "output": {"ref": "s3nodeTwo"}
},
{
  "name": "action",
  "type": "SnsAlarm",
  "message": "S3 bucket '#{node.output.directoryPath}' succeeded at
#{node.@actualEndTime}.",
  "subject": "Testing",
  "topicArn": "arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic",
  "role": "DataPipelineDefaultRole"
}

```

Avaliação de expressões

O AWS Data Pipeline fornece um conjunto de funções que você pode usar para calcular o valor de um campo. O exemplo a seguir usa a função `makeDate` para definir o campo `startDateTime` de um objeto `Schedule` como `"2011-05-24T0:00:00" GMT/UTC`.

```

"startDateTime" : "makeDate(2011,5,24)"

```

Funções matemáticas

As funções a seguir estão disponíveis para uso com valores numéricos.

| Função | Descrição |
|--------|---|
| + | Adição. Exemplo: $\#{1 + 2}$ Result: 3 |
| - | Subtração. Exemplo: $\#{1 - 2}$ Result: -1 |
| * | Multiplicação. Exemplo: $\#{1 * 2}$ Result: 2 |
| / | Divisão. Se você dividir dois números inteiros, o resultado será truncado. Exemplo: $\#{1 / 2}$, resultado: 0 Exemplo: $\#{1.0 / 2}$, resultado: .5 |
| ^ | Expoente. Exemplo: $\#{2 ^ 2}$ Result: 4.0 |

Funções de string

As funções a seguir estão disponíveis para uso com valores de string.

| Função | Descrição |
|--------|---|
| + | <p>Concatenação. Os valores que não são de string são convertidos primeiro em valores de strings.</p> <p>Exemplo: <code>#{ "he1" + "1o" }</code></p> <p>Result: "hello"</p> |

Funções de data e hora

As funções a seguir estão disponíveis para uso com valores `DateTime`. Nos exemplos, o valor de `myDateTime` é `May 24, 2011 @ 5:10 pm GMT`.

Note

O formato de data/hora para o AWS Data Pipeline é Joda Time, que é um substituto para as classes de data e hora de Java. Para mais informações, consulte [Joda Time - Classe DateTimeFormat](#).

| Função | Descrição |
|---|---|
| <code>int day(DateTime myDateTime)</code> | <p>Obtém o dia do valor <code>DateTime</code> como um número inteiro.</p> <p>Exemplo: <code>#{ day(myDateTime) }</code></p> <p>Result: 24</p> |
| <code>int dayOfYear(DateTime myDateTime)</code> | <p>Obtém o dia do ano de <code>DateTime</code> como um número inteiro.</p> |

| Função | Descrição |
|--|--|
| | <p>Exemplo: <code>#{dayOfYear(myDateTime)}</code></p> <p>Result: 144</p> |
| <pre>DateTime firstOfMonth(DateTime myDateTime)</pre> | <p>Cria um objeto DateTime para o início do mês no DateTime especificado.</p> <p>Exemplo: <code>#{firstOfMonth(myDateTime)}</code></p> <p>Result: "2011-05-01T17:10:00z"</p> |
| <pre>String format(DateTime myDateTime, String format)</pre> | <p>Cria um objeto String que é o resultado da conversão do DateTime especificado usando a string de formato especificada.</p> <p>Exemplo: <code>#{format(myDateTime, 'YYYY-MM-dd HH:mm:ss z')}</code></p> <p>Result: "2011-05-24T17:10:00 UTC"</p> |
| <pre>int hour(DateTime myDateTime)</pre> | <p>Obtém a hora do valor DateTime como um número inteiro.</p> <p>Exemplo: <code>#{hour(myDateTime)}</code></p> <p>Result: 17</p> |

| Função | Descrição |
|--|---|
| <pre>DateTime makeDate(int year,int month,int day)</pre> | <p>Cria um objeto DateTime, em UTC, com ano, mês e dia especificados, à meia-noite.</p> <p>Exemplo: <code>#{makeDate(2011,5,24)}</code></p> <p>Result: "2011-05-24T0:00:00z"</p> |
| <pre>DateTime makeDateTime(int year,int month,int day,int hour,int minute)</pre> | <p>Cria um objeto DateTime, em UTC, com ano, mês, dia, hora e minuto especificados.</p> <p>Exemplo: <code>#{makeDateTime(2011,5,24,14,21)}</code></p> <p>Result: "2011-05-24T14:21:00z"</p> |
| <pre>DateTime midnight(DateTime myDateTime)</pre> | <p>Cria um objeto DateTime para a meia-noite atual, em relação ao DateTime especificado. Por exemplo, onde MyDateTime for 2011-05-25T17:10:00z , o resultado será o seguinte:</p> <p>Exemplo: <code>#{midnight(myDateTime)}</code></p> <p>Result: "2011-05-25T0:00:00z"</p> |

| Função | Descrição |
|--|---|
| <code>DateTime minusDays(DateTime myDateTime,int daysToSub)</code> | <p>Cria um objeto DateTime que é o resultado da subtração do número de dias especificado a partir do DateTime especificado.</p> <p>Exemplo: <code>#{minusDays(myDateTime,1)}</code></p> <p>Result: "2011-05-23T17:10:00z"</p> |
| <code>DateTime minusHours(DateTime myDateTime,int hoursToSub)</code> | <p>Cria um objeto DateTime que é o resultado da subtração do número de horas especificado a partir do DateTime especificado.</p> <p>Exemplo: <code>#{minusHours(myDateTime,1)}</code></p> <p>Result: "2011-05-24T16:10:00z"</p> |
| <code>DateTime minusMinutes(DateTime myDateTime,int minutesToSub)</code> | <p>Cria um objeto DateTime que é o resultado da subtração do número de minutos especificado a partir do DateTime especificado.</p> <p>Exemplo: <code>#{minusMinutes(myDateTime,1)}</code></p> <p>Result: "2011-05-24T17:09:00z"</p> |

| Função | Descrição |
|--|---|
| <code>DateTime minusMonths(DateTime myDateTime,int monthsToSub)</code> | <p>Cria um objeto DateTime que é o resultado da subtração do número de meses especificado a partir do DateTime especificado.</p> <p>Exemplo: <code>#{minusMonths(myDateTime,1)}</code></p> <p>Result: "2011-04-24T17:10:00z"</p> |
| <code>DateTime minusWeeks(DateTime myDateTime,int weeksToSub)</code> | <p>Cria um objeto DateTime que é o resultado da subtração do número de semanas especificado a partir do DateTime especificado.</p> <p>Exemplo: <code>#{minusWeeks(myDateTime,1)}</code></p> <p>Result: "2011-05-17T17:10:00z"</p> |
| <code>DateTime minusYears(DateTime myDateTime,int yearsToSub)</code> | <p>Cria um objeto DateTime que é o resultado da subtração do número de anos especificado a partir do DateTime especificado.</p> <p>Exemplo: <code>#{minusYears(myDateTime,1)}</code></p> <p>Result: "2010-05-24T17:10:00z"</p> |

| Função | Descrição |
|--|---|
| <code>int minute(DateTime myDateTime)</code> | <p>Obtém o minuto do valor <code>DateTime</code> como um número inteiro.</p> <p>Exemplo: <code>#{minute(myDateTime)}</code></p> <p>Result: 10</p> |
| <code>int month(DateTime myDateTime)</code> | <p>Obtém o mês do valor <code>DateTime</code> como um número inteiro.</p> <p>Exemplo: <code>#{month(myDateTime)}</code></p> <p>Result: 5</p> |
| <code>DateTime plusDays(DateTime myDateTime, int daysToAdd)</code> | <p>Cria um objeto <code>DateTime</code> que é o resultado da adição do número de dias especificado ao <code>DateTime</code> especificado.</p> <p>Exemplo: <code>#{plusDays(myDateTime, 1)}</code></p> <p>Result: "2011-05-25T17:10:00z"</p> |

| Função | Descrição |
|---|--|
| <code>DateTime plusHours(DateTime myDateTime,int hoursToAdd)</code> | <p>Cria um objeto DateTime que é o resultado da adição do número de horas especificado ao DateTime especificado.</p> <p>Exemplo: <code>#{plusHours(myDateTime,1)}</code></p> <p>Result: "2011-05-24T18:10:00z"</p> |
| <code>DateTime plusMinutes(DateTime myDateTime,int minutesToAdd)</code> | <p>Cria um objeto DateTime que é o resultado da adição do número de minutos especificado ao DateTime especificado.</p> <p>Exemplo: <code>#{plusMinutes(myDateTime,1)}</code></p> <p>Result: "2011-05-24 17:11:00z"</p> |
| <code>DateTime plusMonths(DateTime myDateTime,int monthsToAdd)</code> | <p>Cria um objeto DateTime que é o resultado da adição do número de meses especificado ao DateTime especificado.</p> <p>Exemplo: <code>#{plusMonths(myDateTime,1)}</code></p> <p>Result: "2011-06-24T17:10:00z"</p> |

| Função | Descrição |
|---|---|
| <code>DateTime plusWeeks(DateTime myDateTime,int weeksToAdd)</code> | <p>Cria um objeto <code>DateTime</code> que é o resultado da adição do número de semanas especificado ao <code>DateTime</code> especificado.</p> <p>Exemplo: <code>#{plusWeeks(myDateTime,1)}</code></p> <p>Result: "2011-05-31T17:10:00z"</p> |
| <code>DateTime plusYears(DateTime myDateTime,int yearsToAdd)</code> | <p>Cria um objeto <code>DateTime</code> que é o resultado da adição do número de anos especificado ao <code>DateTime</code> especificado.</p> <p>Exemplo: <code>#{plusYears(myDateTime,1)}</code></p> <p>Result: "2012-05-24T17:10:00z"</p> |
| <code>DateTime sunday(DateTime myDateTime)</code> | <p>Cria um objeto <code>DateTime</code> para o domingo anterior, em relação ao <code>DateTime</code> especificado. Se o <code>DateTime</code> especificado for um domingo, o resultado será o <code>DateTime</code> especificado.</p> <p>Exemplo: <code>#{sunday(myDateTime)}</code></p> <p>Result: "2011-05-22 17:10:00 UTC"</p> |

| Função | Descrição |
|--|--|
| <code>int year(DateTime myDateTime)</code> | <p>Obtém o ano do valor <code>DateTime</code> como um número inteiro.</p> <p>Exemplo: <code>#{year(myDateTime)}</code></p> <p>Result: 2011</p> |
| <code>DateTime yesterday(DateTime myDateTime)</code> | <p>Cria um objeto <code>DateTime</code> para o dia anterior, em relação ao <code>DateTime</code> especificado. O resultado é o mesmo que <code>minusDays (1)</code>.</p> <p>Exemplo: <code>#{yesterday(myDateTime)}</code></p> <p>Result: "2011-05-23T17:10:00z"</p> |

Caracteres especiais

O AWS Data Pipeline usa certos caracteres que têm um significado especial nas definições de pipeline, conforme mostrado na tabela a seguir.

| Caractere especial | Descrição | Exemplos |
|--------------------|---|--|
| @ | Campo de tempo de execução. Este caractere é um prefixo de nome de campo para um campo que fica disponível apenas quando um pipeline é executado. | <p>@actualStartTime</p> <p>@failureReason</p> <p>@resourceStatus</p> |

| Caractere especial | Descrição | Exemplos |
|--------------------|---|--|
| # | Expressão. As expressões são delimitadas por: “#{“ e ”}”, e o conteúdo das chaves é avaliado pelo AWS Data Pipeline. Para obter mais informações, consulte Expressões . | <code>#{format(myDateTime,'YYYY-MM-dd hh:mm:ss')}</code> <code>s3://mybucket/#{id}.csv</code> |
| * | Campo criptografado. Este caractere é um prefixo de nome de campo para indicar que o AWS Data Pipeline deve criptografar o conteúdo do campo transferido entre o console ou a CLI e o serviço do AWS Data Pipeline. | <code>*password</code> |

Referência de objeto de pipeline

Você pode usar os objetos e componentes de pipeline a seguir na sua definição de pipeline.

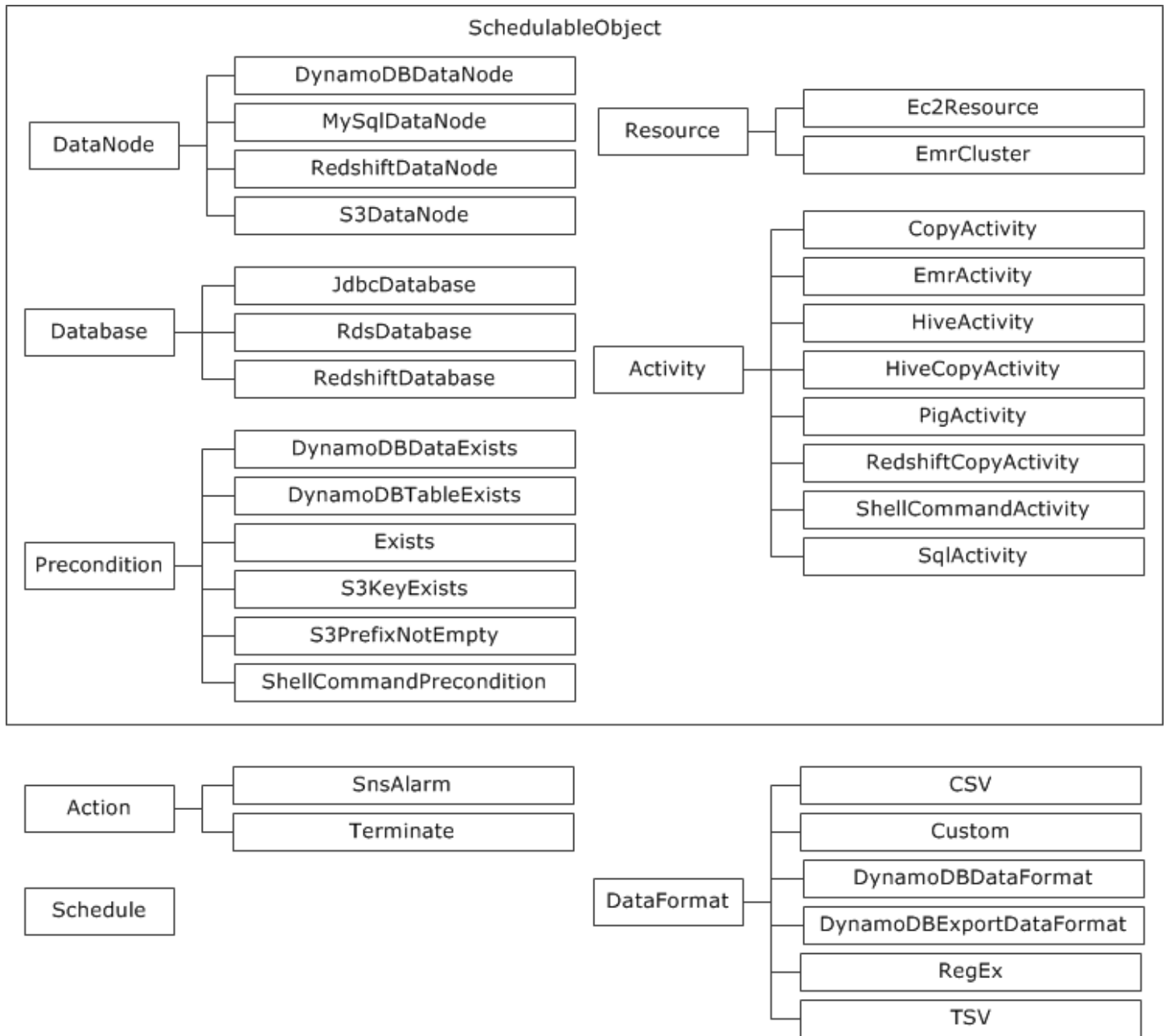
Conteúdo

- [Nós de dados](#)
- [Atividades](#)
- [Recursos](#)
- [Precondições](#)
- [Bancos de dados](#)
- [Formatos de dados](#)
- [Ações](#)
- [Schedule](#)
- [Utilitários](#)

Note

Para ver um exemplo de aplicativo que usa o AWS Data Pipeline JavaSDK, consulte [Data Pipeline DynamoDB Export Java Sample](#) on. GitHub

A seguir está a hierarquia de objetos para AWS Data Pipeline.



Nós de dados

A seguir estão os objetos do nó de AWS Data Pipeline dados:

Objetos

- [ynamoDBDataNodo D](#)
- [MySQLDataNode](#)
- [RedshiftDataNode](#)

- [S3 DataNode](#)
- [SqlDataNode](#)

ynamoDBDataNode D

Define um nó de dados usando o DynamoDB, que é especificado como uma entrada para um objeto HiveActivity ou EMRActivity.

Note

O objeto DynamoDBDataNode não oferece suporte à condição Exists.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Esse objeto faz referência a dois outros objetos definidos por você no mesmo arquivo de definição de pipeline. CopyPeriod é um objeto Schedule e Ready é um objeto de condição.

```
{
  "id" : "MyDynamoDBTable",
  "type" : "DynamoDBDataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "tableName" : "adEvents",
  "precondition" : { "ref" : "Ready" }
}
```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|-------------------------|--------------|
| tableName | Uma tabela do DynamoDB. | String |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|---|
| schedule | Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre configurações opcionais de programação de exemplo, consulte Programação . | Objeto de referência, por exemplo, "schedule": {"ref": "myScheduleId" } |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se esse campo estiver definido, uma nova atividade remota não concluída no tempo definido de início poderá ser repetida. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|----------------------------------|---|---|
| <code>dataFormat</code> | DataFormat para os dados descritos por esse nó de dados. Atualmente suportado por HiveActivity HiveCopyActivity e. | Objeto de referência, "dataFormat": {"ref": "myDynamoDBDataFormatId" } |
| <code>dependsOn</code> | Especifique a dependência em outro objeto executável | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId" } |
| <code>failureAndRerunModo</code> | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| <code>lateAfterTimeout</code> | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| <code>maxActiveInstances</code> | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| <code>maximumRetries</code> | Quantidade máxima de novas tentativas com falha. | Inteiro |
| <code>onFail</code> | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId" } |
| <code>onLateAction</code> | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId" } |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|--|---|
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId" } |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId" } |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId" } |
| readThroughputPercent | Define a taxa de operações de leitura para manter sua taxa de throughput provisionado do DynamoDB no intervalo alocado para sua tabela. O valor é um dobro entre 0,1 e 1, incluindo ambos. | Double |
| região | O código da região na qual a tabela do DynamoDB está. Por exemplo, us-east-1. Isso é usado HiveActivity quando ele executa a preparação de tabelas do DynamoDB no Hive. | Enumeração |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|---|
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |
| runsOn | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceId" } |
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|---|--------------|
| writeThroughputPercent | Define a taxa de operações de gravação para manter sua taxa de throughput provisionado do DynamoDB no intervalo alocado para sua tabela. O valor é um dobro entre 0,1 e 1,0, incluindo ambos. | Double |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstance | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

MySQLDataNode

Define um nó de dados usando MySQL.

Note

O tipo `MySQLDataNode` está obsoleto. Em vez disso, recomendamos o uso de [SqlDataNode](#).

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Esse objeto faz referência a dois outros objetos definidos por você no mesmo arquivo de definição de pipeline. `CopyPeriod` é um objeto `Schedule` e `Ready` é um objeto de condição.

```
{
  "id" : "Sql Table",
  "type" : "MySQLDataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "table" : "adEvents",
  "username": "user_name",
  "*password": "my_password",
  "connectionString": "jdbc:mysql://mysqlinstance-rds.example.us-
east-1.rds.amazonaws.com:3306/database_name",
  "selectQuery" : "select * from #{table} where eventTime >=
'#{@scheduledStartTime.format('YYYY-MM-dd HH:mm:ss')}' and eventTime <
'#{@scheduledEndTime.format('YYYY-MM-dd HH:mm:ss')}'",
  "precondition" : { "ref" : "Ready" }
}
```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| tabela | O nome da tabela no Meu SQL banco de dados. | String |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|---|
| schedule | <p>Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.</p> | Objeto de referência, por exemplo, "agenda": {"ref": "myScheduleId" } |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|--------------|
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|----------------------------------|---|---|
| <code>createTableSql</code> | SQLLE uma expressão de criação de tabela que cria a tabela. | String |
| <code>banco de dados</code> | O nome do banco de dados. | Objeto de referência, por exemplo, "banco de dados": {"ref": "myDatabaseId" } |
| <code>dependsOn</code> | Especifica uma dependência em outro objeto executável. | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId" } |
| <code>failureAndRerunModo</code> | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| <code>insertQuery</code> | Uma SQL declaração para inserir dados na tabela. | String |
| <code>lateAfterTimeout</code> | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como <code>ondemand</code> . | Período |
| <code>maxActiveInstances</code> | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| <code>maximumRetries</code> | Quantidade máxima de novas tentativas com falha. | Inteiro |
| <code>onFail</code> | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId" } |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|--|---|
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId" } |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId" } |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId" } |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId" } |
| reportProgressTime out | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |
| runsOn | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceId" } |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |
| schemaName | O nome do esquema que mantém a tabela | String |
| selectQuery | Uma SQL declaração para buscar dados da tabela. | String |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceid | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |
| Campos do sistema | Descrição | Tipo de slot |
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [S3 DataNode](#)

RedshiftDataNode

Define um nó de dados usando o Amazon Redshift. O `RedshiftDataNode` representa as propriedades dos dados em um banco de dados, como uma tabela de dados, usada pelo seu pipeline.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyRedshiftDataNode",
```

```

"type" : "RedshiftDataNode",
"database": { "ref": "MyRedshiftDatabase" },
"tableName": "adEvents",
"schedule": { "ref": "Hour" }
}

```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|--|--|
| banco de dados | O banco de dados em que a tabela reside. | Objeto de referência, por exemplo, "banco de dados": {"ref": "myRedshiftDatabaseId"} |
| tableName | O nome da tabela do Amazon Redshift. A tabela é criada se ainda não existir e você tiver fornecido createTableSql. | String |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|--|
| schedule | Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline | Objeto de referência, por exemplo, "agenda": {"ref": "myScheduleId"} |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|--------------|
| | tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html . | |
| Campos opcionais | Descrição | Tipo de slot |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| createTableSql | Uma SQL expressão para criar a tabela no banco de dados. Recomendamos que você especifique o esquema em que a tabela deve ser criada, por exemplo: CREATE TABLE mySchema.myTable (bestColumn varchar (25) chave primária (chave distinta, numberOfWins número inteiro). sortKey AWS Data Pipeline executará o script no createTableSql campo se a tabela especificada portableName, não existir no esquema especificado pelo schemaName e campo. Por exemplo, se você especificar schemaName como, mySchema mas não incluir mySchema no createTableSql campo, a tabela será criada no esquema errado (por | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--|
| | padrão, ela seria criada em PUBLIC). Isso ocorre porque o AWS Data Pipeline não analisa suas CREATE TABLE declarações. | |
| dependsOn | Especifique a dependência em outro objeto executável | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId"} |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | A quantidade máxima de novas tentativas após uma falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|--|---|
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId" } |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId" } |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId" } |
| primaryKeys | Se você não especificar uma tabela primaryKeys de destino em <code>RedShiftCopyActivity</code> , poderá especificar uma lista de colunas usando as primaryKeys quais funcionarão como mergeKey a. No entanto, se você tiver uma existente primaryKey definida em uma tabela do Amazon Redshift, essa configuração substituirá a chave existente. | String |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|---|
| runsOn | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceid" } |
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |
| schemaName | Este campo opcional especifica o nome do esquema para a tabela do Amazon Redshift. Se não for especificado, o nome do esquema seráPUBLIC, que é o esquema padrão no Amazon Redshift. Para obter mais informações, consulte o Guia do desenvolvedor do banco de dados do Amazon Redshift. | String |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|--|---|
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |
| Campos de tempo de execução | Descrição | Tipo de slot |
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceID | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

S3 DataNode

Define um nó de dados usando o Amazon S3. Por padrão, o S3 DataNode usa criptografia do lado do servidor. Se você quiser desabilitar isso, defina `s3 EncryptionType` como `NONE`.

Note

Quando você usa um `S3DataNode` como entrada para `CopyActivity`, somente os formatos de TSV dados CSV e são suportados.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Esse objeto faz referência a outro objeto definido por você no mesmo arquivo de definição de pipeline. `CopyPeriod` é um objeto `Schedule`.

```
{
  "id" : "OutputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://myBucket/#{@scheduledStartTime}.csv"
}
```

Sintaxe

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|---|
| schedule | Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando <code>"agenda": {"ref": "DefaultSchedule"}</code> . Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programaç | Objeto de referência, por exemplo, <code>"agenda": {"ref": "myScheduleId"}</code> |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|---|
| | <p>ão principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.</p> | |
| Campos opcionais | Descrição | Tipo de slot |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| compression | O tipo de compactação dos dados descritos pelo S3DataNode. "none" não é compressão e "gzip" é comprimido com o algoritmo gzip. Esse campo só é compatível para uso com o Amazon Redshift e quando você usa o DataNode S3 com. CopyActivity | Enumeração |
| dataFormat | DataFormat para os dados descritos por este S3DataNode. | Objeto de referência, por exemplo, "dataFormat": {"ref": "myDataFormat Id"} |
| dependsOn | Especifique a dependência em outro objeto executável | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId"} |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--------------|
| directoryPath | Caminho do diretório Amazon S3 comoURI: s3://my-bucket/. my-key-for-directory Você deve fornecer um directoryPath valor filePath ou. | String |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| filePath | O caminho para o objeto no Amazon S3 comoURI, por exemplo: s3://my-bucket/. my-key-for-file Você deve fornecer um directory Path valor filePath ou. Eles representam um nome de pasta e de arquivo. Use o directory Path valor para acomodar vários arquivos em um diretório. | String |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| manifestFilePath | O caminho do Amazon S3 para um arquivo de manifesto no formato suportado pelo Amazon Redshift. AWS Data Pipeline usa o arquivo manifesto para copiar os arquivos do Amazon S3 especificados na tabela. Esse campo é válido somente quando um RedShiftCopyActivity faz referência ao S3DataNode. | String |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|--|--|
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId"} |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|-------------------|--|--|
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |
| runsOn | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceId"} |
| s3 EncryptionType | Substitui o tipo de criptografia do Amazon S3. Os valores são SERVER _ SIDE _ ENCRYPTION ou NONE. A criptografia do lado do servidor é ativada por padrão. | Enumeração |
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceID | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [MySqlDataNode](#)

SqlDataNode

Define um nó de dados usando o SQL.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Esse objeto faz referência a dois outros objetos definidos por você no mesmo arquivo de definição de pipeline. CopyPeriod é um objeto Schedule e Ready é um objeto de condição.

```
{
  "id" : "Sql Table",
  "type" : "SqlDataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "table" : "adEvents",
  "database":"myDataBaseName",
  "selectQuery" : "select * from #{table} where eventTime >=
'#{@scheduledStartTime.format('YYYY-MM-dd HH:mm:ss')}' and eventTime <
'#{@scheduledEndTime.format('YYYY-MM-dd HH:mm:ss')}'",
  "precondition" : { "ref" : "Ready" }
}
```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| tabela | O nome da tabela no SQL banco de dados. | String |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|---|
| schedule | Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência | Objeto de referência, por exemplo, "agenda": {"ref": "myScheduleId" } |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|---|
| | a de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html . | |
| Campos opcionais | Descrição | Tipo de slot |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| createTableSql | SQL uma expressão de criação de tabela que cria a tabela. | String |
| banco de dados | O nome do banco de dados. | Objeto de referência, por exemplo, "banco de dados": {"ref": "myDatabaseId" } |
| dependsOn | Especifica a dependência em outro objeto executável. | Objeto de referência, por exemplo, |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|---|
| | | "dependsOn": {"ref": "myActivityId" } |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| insertQuery | Uma SQL declaração para inserir dados na tabela. | String |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId" } |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId" } |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId" } |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|---|--|
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "pré-condição": {"ref": " myPreconditionId "} |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress. Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |
| runsOn | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": " myResourceId "} |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |
| schemaName | O nome do esquema que mantém a tabela | String |
| selectQuery | Uma SQL declaração para buscar dados da tabela. | String |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceID | ID do último objeto da instância concluído. | String |
| @healthStatusUpdatedHour | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRunHour | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |
| Campos do sistema | Descrição | Tipo de slot |
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [S3 DataNode](#)

Atividades

A seguir estão os objetos da AWS Data Pipeline atividade:

Objetos

- [CopyActivity](#)
- [EmrActivity](#)
- [HadoopActivity](#)
- [HiveActivity](#)
- [HiveCopyActivity](#)

- [PigActivity](#)
- [RedshiftCopyActivity](#)
- [ShellCommandActivity](#)
- [SqlActivity](#)

CopyActivity

Copia dados de um local para outro. CopyActivity suporta [S3 DataNode](#) e [SqlDataNode](#) como entrada e saída e a operação de cópia é normalmente executada record-by-record. No entanto, CopyActivity fornece cópia do Amazon S3 para Amazon S3 de alto desempenho quando todas as condições a seguir são atendidas:

- A entrada e a saída são S3 DataNodes
- O campo dataFormat é igual para a entrada e a saída

Se você fornecer arquivos de dados compactados como entrada e não indicar isso usando o campo `compression` nos nós de dados do S3, CopyActivity poderá falhar. Nesse caso, CopyActivity não detecta corretamente o fim do caractere de gravação e ocorre falha na operação. Além disso, CopyActivity oferece suporte à cópia de um diretório para outro diretório e à cópia de um arquivo em um diretório, mas a record-by-record cópia ocorre ao copiar um diretório para um arquivo. Por fim, CopyActivity não oferece suporte à cópia de arquivos do Amazon S3 de várias partes.

CopyActivity tem limitações específicas em seu CSV suporte. Ao usar um S3 DataNode como entrada para CopyActivity, você só pode usar uma variante Unix/Linux do formato de arquivo de CSV dados para os campos de entrada e saída do Amazon S3. A variante Unix/Linux requer o seguinte:

- O separador precisa ser o caractere "," (vírgula).
- Os registros não ficam entre aspas.
- O caractere de escape padrão é ASCII o valor 92 (barra invertida).
- O identificador de fim de registro é ASCII o valor 10 (ou "\n").

Os sistemas baseados em Windows normalmente usam uma sequência de end-of-record caracteres diferente: um retorno de carro e alimentação de linha juntos (ASCII valor 13 e ASCII valor 10).

Você precisa acomodar essa diferença usando um mecanismo adicional, como um script de pré-cópia para modificação de dados de entrada, para garantir que CopyActivity possa detectar corretamente o final de um registro. Caso contrário, CopyActivity apresentará falhas repetidamente.

Ao usar CopyActivity para exportar de um SQL RDS objeto Postgre para um formato de TSV dados, o NULL caractere padrão é \n.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Esse objeto faz referência a três outros objetos definidos por você no mesmo arquivo de definição de pipeline. CopyPeriod é um objeto Schedule e InputData e OutputData são objetos de nó de dados.

```
{
  "id" : "S3ToS3Copy",
  "type" : "CopyActivity",
  "schedule" : { "ref" : "CopyPeriod" },
  "input" : { "ref" : "InputData" },
  "output" : { "ref" : "OutputData" },
  "runsOn" : { "ref" : "MyEc2Resource" }
}
```

Sintaxe

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|---|
| schedule | Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline | Objeto de referência, por exemplo, "agenda": {"ref": "myScheduleId" } |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--|--|---|
| | padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html . | |
| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
| runsOn | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceId" } |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |
| Campos opcionais | Descrição | Tipo de slot |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota | Período |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|---|
| | não concluída dentro do prazo definido poderá ser executada novamente. | |
| dependsOn | Especifique a dependência em outro objeto executável. | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId"} |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| input | A fonte de dados de entrada. | Objeto de referência, por exemplo, "input": {"ref": "myDataNodeId"} |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|---|--|
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| output | A fonte de dados de saída. | Objeto de referência, por exemplo, "output": {"ref": "myDataNodeId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId"} |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress. Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id"} |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--|
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id"} |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceid | ID do último objeto da instância concluído. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [ShellCommandActivity](#)
- [EmrActivity](#)
- [Exportar dados do MySQL para o Amazon S3 usando a AWS Data Pipeline](#)

EmrActivity

Executa um cluster do EMR.

AWS Data Pipeline usa um formato de etapas diferente do da AmazonEMR; por exemplo, AWS Data Pipeline usa argumentos separados por vírgula após o JAR nome no campo da `EmrActivity` etapa. O exemplo a seguir mostra uma etapa formatada para a AmazonEMR, seguida por sua AWS Data Pipeline equivalente:

```
s3://example-bucket/MyWork.jar arg1 arg2 arg3
```

```
"s3://example-bucket/MyWork.jar,arg1,arg2,arg3"
```

Exemplos

Veja a seguir um exemplo deste tipo de objeto. Este exemplo usa versões mais antigas da AmazonEMR. Verifique se este exemplo está correto com a versão do EMR cluster da Amazon que você está usando.

Esse objeto faz referência a três outros objetos definidos por você no mesmo arquivo de definição de pipeline. `MyEmrCluster` é um objeto `EmrCluster` e `MyS3Input` e `MyS3Output` são objetos `S3DataNode`.

Note

Neste exemplo, você pode substituir o `step` campo pela string de cluster desejada, que pode ser um script Pig, um cluster de streaming do Hadoop, seu próprio personalizado, JAR incluindo seus parâmetros, etc.

Hadoop 2.x (3.x) AMI

```
{
  "id" : "MyEmrActivity",
  "type" : "EmrActivity",
  "runsOn" : { "ref" : "MyEmrCluster" },
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : ["s3://mybucket/myPath/myStep.jar,firstArg,secondArg,-files,s3://mybucket/myPath/myFile.py,-input,s3://myinputbucket/path,-output,s3://myoutputbucket/path,-mapper,myFile.py,-reducer,reducerName","s3://mybucket/myPath/myotherStep.jar,..."],
  "postStepCommand" : "scp localFiles remoteFiles",
  "input" : { "ref" : "MyS3Input" },
  "output" : { "ref" : "MyS3Output" }
}
```

Note

Para transmitir argumentos para um aplicativo em uma etapa, é necessário especificar a Região no caminho do script, conforme mostrado no exemplo a seguir. Além disso, talvez seja necessário escapar os argumentos transmitidos. Por exemplo, se você usar `script-runner.jar` para executar um script de shell e quiser passar argumentos para o script, precisará escapar as vírgulas que os separam. O slot de etapa a seguir ilustra como fazer isso:

```
"step" : "s3://eu-west-1.elasticmapreduce/libs/script-runner/script-runner.jar,s3://datapipeline/echo.sh,a\\,b\\,c"
```

Esta etapa usa `script-runner.jar` para executar o script de shell `echo.sh` e passa a, b e c como um único argumento para o script. O primeiro caractere de escape é removido do argumento resultante. Por isso, talvez você precise realizar o escape novamente. Por exemplo, se você tivesse `File\.gz` como argumentoJSON, você poderia escapar usando `File\\\\.gz`. No entanto, como o primeiro escape é descartado, você precisa usar `File\\\\\\\\.gz`.

Sintaxe


| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|--|
| schedule | <p>Esse objeto é invocado durante a execução de um intervalo de programação. Especifique uma referência de programação para outro objeto para definir a ordem de execução de dependência desse objeto. É possível satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, ao especificar <code>"schedule": {"ref": "DefaultSchedule"}</code>. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), você poderá criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.</p> | <p>Objeto de referência, por exemplo, <code>"schedule": {"ref": "myScheduleId"}</code></p> |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|--|---|
| runsOn | O EMR cluster da Amazon no qual esse trabalho será executado. | Objeto de referência, por exemplo, "runsOn": {"ref": "myEmrCluster Id"} |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, será ignorado.workerGroup | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|--|---|
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se definida, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| dependsOn | Especifique a dependência em outro objeto executável. | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId"} |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| input | O local dos dados de entrada. | Objeto de referência, por exemplo, "input": |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------------------|---|--|
| | | <code>{"ref": "myDataNode Id"}</code> |
| <code>lateAfterTimeout</code> | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como <code>ondemand</code> . | Período |
| <code>maxActiveInstances</code> | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| <code>maximumRetries</code> | A quantidade máxima de novas tentativas após uma falha. | Inteiro |
| <code>onFail</code> | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, <code>"onFail": {"ref": "myActionId"}</code> |
| <code>onLateAction</code> | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, <code>"onLateAction": {"ref": "myActionId"}</code> |
| <code>onSuccess</code> | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, <code>"onSuccess": {"ref": "myActionId"}</code> |
| <code>output</code> | O local dos dados de saída. | Objeto de referência, por exemplo, <code>"output": {"ref": "myDataNode Id"}</code> |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|---|--|
| parent | O pai do objeto atual do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| pipelineLogUri | O Amazon S3URI, como 's3://BucketName/Prefix/' para carregar registros para o pipeline. | String |
| postStepCommand | Scripts de shell a serem executados depois que todas as etapas são concluídas. Para especificar vários scripts, até 255, adicione vários campos postStepCommand . | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId"} |
| preStepCommand | Scripts de shell a serem executados antes de qualquer etapa ser executada. Para especificar vários scripts, até 255, adicione vários campos preStepCommand . | String |
| reportProgressTimeout | O tempo limite para as chamadas sucessivas de trabalho remoto para reportProgress . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|---|--|--------------|
| <code>resizeClusterBeforeRunning</code> | <p>Redimensionar o cluster antes de executar esta atividade para acomodar tabelas do DynamoDB especificadas como entradas ou saídas.</p> <div data-bbox="472 445 1149 1003" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Se você <code>EmrActivity</code> usa a <code>DynamoDBDataNode</code> como nó de dados de entrada ou saída e define o <code>resizeClusterBeforeRunning</code> como <code>TRUE</code>, AWS Data Pipeline começa a usar tipos de <code>m3.xlarge</code> instância. Isso substitui suas escolhas de tipo de instância por <code>m3.xlarge</code>, o que pode aumentar seus custos mensais.</p></div> | Booleano |
| <code>resizeClusterMaxInstâncias</code> | Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento. | Inteiro |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------------|--|--------------|
| <code>scheduleType</code> | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou final do intervalo. Os valores são: <code>cron</code> , <code>ondemand</code> e <code>timeseries</code> . A programação <code>timeseries</code> significa que as instâncias são programadas no final de cada intervalo. A programação <code>cron</code> significa que as instâncias são programadas no início de cada intervalo. Uma programação <code>ondemand</code> permite que você execute um pipeline uma vez por ativação. Você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação <code>ondemand</code> , ela precisará ser especificada no objeto padrão, além de ser a única <code>scheduleType</code> especificada para objetos no pipeline. Para usar pipelines <code>ondemand</code> , chame a operação <code>ActivatePipeline</code> para cada execução subsequente. | Enumeração |
| <code>step (etapa)</code> | Uma ou mais etapas para que o cluster seja executado. Para especificar várias etapas, até 255, adicione vários campos de etapa. Use argumentos separados por vírgula após o JAR nome; por exemplo, <code>"s3://example-bucket/MyWork.jar, arg1, arg2, arg3"</code> . | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-------------------------------|--|---|
| <code>@activeInstances</code> | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, <code>"activeIn</code> |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--|
| | | stances": {" ref": "myRunnableObject Id"}" |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {" ref": "myRunnableObject Id "} |
| emrStepLog | Os registros de EMR etapas da Amazon estão disponíveis somente em tentativas de EMR atividade | String |
| errorId | O errorId se esse objeto apresentou falha. | String |
| errorMessage | O errorMessage se esse objeto apresentou falha. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceid | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término programado para o objeto. | DateTime |
| @scheduledStartTime | Horário de início programado para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id"}" |
| Campos do sistema | Descrição | Tipo de slot |
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [ShellCommandActivity](#)
- [CopyActivity](#)
- [EmrCluster](#)

HadoopActivity

Executa um MapReduce trabalho em um cluster. O cluster pode ser um EMR cluster gerenciado por AWS Data Pipeline ou outro recurso, se você usar TaskRunner. Use HadoopActivity quando quiser executar o trabalho em paralelo. Isso permite que você use os recursos de agendamento da YARN estrutura ou do negociador de MapReduce recursos no Hadoop 1. Se você quiser executar o trabalho sequencialmente usando a ação Amazon EMR Step, você ainda pode usar. [EmrActivity](#)

Exemplos

HadoopActivity usando um EMR cluster gerenciado pelo AWS Data Pipeline

O HadoopActivity objeto a seguir usa um EmrCluster recurso para executar um programa:

```
{
  "name": "MyHadoopActivity",
  "schedule": {"ref": "ResourcePeriod"},
  "runsOn": {"ref": "MyEmrCluster"},
  "type": "HadoopActivity",
  "preActivityTaskConfig": {"ref": "preTaskScriptConfig"},
  "jarUri": "/home/hadoop/contrib/streaming/hadoop-streaming.jar",
  "argument": [
    "-files",
    "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
    "-mapper",
    "wordSplitter.py",
    "-reducer",
    "aggregate",
    "-input",
    "s3://elasticmapreduce/samples/wordcount/input/",
    "-output",
    "s3://test-bucket/MyHadoopActivity/#{@pipelineId}/
    #{format(@scheduledStartTime, 'YYYY-MM-dd')}"
  ],
  "maximumRetries": "0",
  "postActivityTaskConfig": {"ref": "postTaskScriptConfig"},
  "hadoopQueue" : "high"
}
```

Aqui está o correspondente *MyEmrCluster*, que configura as filas FairScheduler e YARN para o Hadoop 2 baseado em: AMIs

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
  "hadoopSchedulerType" : "PARALLEL_FAIR_SCHEDULING",
  "amiVersion" : "3.7.0",
  "bootstrapAction" : ["s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop, -z, yarn.scheduler.capacity.root.queues=low
\,high\, default, -z, yarn.scheduler.capacity.root.high.capacity=50, -
```

```
z,yarn.scheduler.capacity.root.low.capacity=10,-
z,yarn.scheduler.capacity.root.default.capacity=30"]
}
```

Isso é o EmrCluster que você usa para configurar FairScheduler no Hadoop 1:

```
{
  "id": "MyEmrCluster",
  "type": "EmrCluster",
  "hadoopSchedulerType": "PARALLEL_FAIR_SCHEDULING",
  "amiVersion": "2.4.8",
  "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-m,mapred.queue.names=low\\\\\\\\,high\\\\\\\\,default,-
m,mapred.fairscheduler.poolnameproperty=mapred.job.queue.name"
}
```

As EmrCluster configurações a seguir são baseadas em CapacityScheduler Hadoop 2: AMIs

```
{
  "id": "MyEmrCluster",
  "type": "EmrCluster",
  "hadoopSchedulerType": "PARALLEL_CAPACITY_SCHEDULING",
  "amiVersion": "3.7.0",
  "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
\\\\\\\\,high,-z,yarn.scheduler.capacity.root.high.capacity=40,-
z,yarn.scheduler.capacity.root.low.capacity=60"
}
```

HadoopActivity usando um EMR cluster existente

Neste exemplo, você usa grupos de trabalho e a TaskRunner para executar um programa em um cluster existente. EMR A seguinte definição de pipeline é usada HadoopActivity para:

- Execute um MapReduce programa somente em *myWorkerGroup* recursos. Para obter mais informações sobre grupos de operadores, consulte [Executar trabalho em recursos existentes usando o Task Runner](#).
- Execute um preActivityTask Config e Config postActivityTask

```
{
```



```

"objects": [
  {
    "argument": [
      "-files",
      "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
      "-mapper",
      "wordSplitter.py",
      "-reducer",
      "aggregate",
      "-input",
      "s3://elasticmapreduce/samples/wordcount/input/",
      "-output",
      "s3://test-bucket/MyHadoopActivity/#{@pipelineId}/
#{format(@scheduledStartTime, 'YYYY-MM-dd')}"
    ],
    "id": "MyHadoopActivity",
    "jarUri": "/home/hadoop/contrib/streaming/hadoop-streaming.jar",
    "name": "MyHadoopActivity",
    "type": "HadoopActivity"
  },
  {
    "id": "SchedulePeriod",
    "startDateTime": "start_datetime",
    "name": "SchedulePeriod",
    "period": "1 day",
    "type": "Schedule",
    "endDateTime": "end_datetime"
  },
  {
    "id": "ShellScriptConfig",
    "scriptUri": "s3://test-bucket/scripts/preTaskScript.sh",
    "name": "preTaskScriptConfig",
    "scriptArgument": [
      "test",
      "argument"
    ],
    "type": "ShellScriptConfig"
  },
  {
    "id": "ShellScriptConfig",
    "scriptUri": "s3://test-bucket/scripts/postTaskScript.sh",
    "name": "postTaskScriptConfig",
    "scriptArgument": [
      "test",

```

```

    "argument"
  ],
  "type": "ShellScriptConfig"
},
{
  "id": "Default",
  "scheduleType": "cron",
  "schedule": {
    "ref": "SchedulePeriod"
  },
  "name": "Default",
  "pipelineLogUri": "s3://test-bucket/logs/2015-05-22T18:02:00.343Z642f3fe415",
  "maximumRetries": "0",
  "workerGroup": "myWorkerGroup",
  "preActivityTaskConfig": {
    "ref": "preTaskScriptConfig"
  },
  "postActivityTaskConfig": {
    "ref": "postTaskScriptConfig"
  }
}
]
}

```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| jarUri | Localização de um JAR no Amazon S3 ou no sistema de arquivos local do cluster com o qual executar. HadoopActivity | String |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|--|
| schedule | Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a | Objeto de referência, por exemplo, "agenda": {"ref": "myScheduleId"} |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--|---|---|
| | <p>definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.</p> | |
| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
| runsOn | EMRCluster no qual esse trabalho será executado. | Objeto de referência, por exemplo, "runsOn": {"ref": "myEmrCluster Id"} |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--|
| argument | Argumentos a serem passados para JAR o. | String |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| dependsOn | Especifique a dependência em outro objeto executável. | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId" } |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| hadoopQueue | O nome da fila do programador do Hadoop em que a atividade será enviada. | String |
| input | Local dos dados de entrada. | Objeto de referência, por exemplo, "input": {"ref": "myDataNodeId" } |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| mainClass | A classe principal da com HadoopActivity a JAR qual você está executando. | String |

| Campos opcionais | Descrição | Tipo de slot |
|--------------------|---|--|
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| output | Local dos dados de saída. | Objeto de referência, por exemplo, "output": {"ref": "myDataNodeId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|---|---|
| postActivityTaskConfig | Script de configuração pós-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, por exemplo, "postActivityTaskConfig": {"ref": "myShellScriptConfigId"} |
| preActivityTaskConfig | Script de configuração pré-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, por exemplo, "preActivityTaskConfig": {"ref": "myShellScriptConfigId"} |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId"} |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress. Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|--|---|
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |
| Campos de tempo de execução | Descrição | Tipo de slot |
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--|
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id"} |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstance | ID do último objeto da instância concluído. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [ShellCommandActivity](#)
- [CopyActivity](#)
- [EmrCluster](#)

HiveActivity

Executa uma consulta do Hive em um EMR cluster. HiveActivity facilita a configuração de uma EMR atividade na Amazon e cria automaticamente tabelas do Hive com base nos dados de entrada vindos do Amazon S3 ou da Amazon. RDS Tudo o que você precisa especificar é o HiveQL a ser executado nos dados de origem. AWS Data Pipeline cria automaticamente tabelas do Hive com `${input1},${input2}`, e assim por diante, com base nos campos de entrada no HiveActivity objeto.

Para as entradas do Amazon S3, o campo `dataFormat` é usado para criar os nomes das colunas do Hive.

Para as entradas My SQL (AmazonRDS), os nomes das colunas da SQL consulta são usados para criar os nomes das colunas do Hive.

Note

Esta atividade usa o Hive [CSVSerde](#).

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Esse objeto faz referência a três outros objetos definidos por você no mesmo arquivo de definição de pipeline. `MySchedule` é um objeto `Schedule` e `MyS3Input` e `MyS3Output` são objetos de nó de dados.

```
{
  "name" : "ProcessLogData",
  "id" : "MyHiveActivity",
  "type" : "HiveActivity",
  "schedule" : { "ref": "MySchedule" },
  "hiveScript" : "INSERT OVERWRITE TABLE ${output1} select
host,user,time,request,status,size from ${input1};",
  "input" : { "ref": "MyS3Input" },
  "output" : { "ref": "MyS3Output" },
  "runsOn" : { "ref": "MyEmrCluster" }
}
```

Sintaxe

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|---|
| <code>schedule</code> | Esse objeto é invocado durante a execução de um intervalo de programação. Especifique uma referência de programação para outro objeto para definir a ordem de execução de dependência desse objeto. Você pode satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando <code>"agenda": {"ref": "DefaultSchedule"}</code> . Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), você poderá criar um objeto principal que tenha uma referência | Objeto de referência, por exemplo, <code>"agenda": {"ref": "myScheduleId"}</code> |


| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|--------------|
| | de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html . | |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|---|--------------|
| hiveScript | O script Hive a ser executado. | String |
| scriptUri | A localização do script do Hive a ser executado (por exemplo, s3://scriptLocation). | String |

| Grupo obrigatório | Descrição | Tipo de slot |
|-------------------|--|---|
| runsOn | O EMR cluster no qual isso HiveActivity é executado. | Objeto de referência, por exemplo, "runsOn": {"ref": "myEmrCluster Id"} |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, será ignorado.workerGroup | String |
| input | A fonte de dados de entrada. | Objeto de referência, como "input": {"ref": "myDataNode Id"} |

| Grupo obrigatório | Descrição | Tipo de slot |
|---------------------|---|--|
| output | A fonte de dados de saída. | Objeto de referência, como "output": {"ref": "myDataNode Id "} |
| Campos opcionais | Descrição | Tipo de slot |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se definida, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| dependsOn | Especifique a dependência em outro objeto executável. | Objeto de referência, como "dependsOn": {"ref": "myActivityId "} |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| hadoopQueue | O nome da fila do programador do Hadoop em que o trabalho será enviado. | String |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|---|---|
| maximumRetries | A quantidade máxima de novas tentativas após uma falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, como "onFail": {"ref": "myActionId" } |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, como "onLateAction": {"ref": "myActionId" } |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, como "onSuccess": {"ref": "myActionId" } |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, como "parent": {"ref": "myBaseObject Id" } |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| postActivityTaskConfig | Script de configuração pós-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, como "postActivityTaskConfig": {"ref": "myShellScript ConfigId" } |
| preActivityTaskConfig | Script de configuração pré-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, como "preActivityTaskConfig": {"ref": "myShellScript ConfigId" } |

| Campos opcionais | Descrição | Tipo de slot |
|---|---|---|
| <code>precondition</code> | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, como "pré-condição": {"ref": "myPreconditionId"} |
| <code>reportProgressTimeout</code> | Tempo limite para as chamadas sucessivas de trabalho remoto para <code>reportProgress</code> . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| <code>resizeClusterBeforeRunning</code> | Redimensione o cluster antes de executar esta atividade para acomodar nós de dados do DynamoDB especificados como entradas ou saídas. <div data-bbox="472 989 1149 1549" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>Se sua atividade usa a DynamoDBD <code>ataNode</code> como um nó de dados de entrada ou saída, e se você definir <code>comoTRUE</code>, AWS Data Pipeline começa <code>resizeClusterBeforeRunning</code> a usar tipos de <code>m3.xlarge</code> instância. Isso substitui suas escolhas de tipo de instância por <code>m3.xlarge</code>, o que pode aumentar seus custos mensais.</p> </div> | Booleano |
| <code>resizeClusterMaxInstances</code> | Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento. | Inteiro |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|--------------|
| scheduleType | <p>O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries.</p> | Enumeração |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|--|--------------|
| <code>scriptVariable</code> | Especifica variáveis de script para EMR a Amazon passar para o Hive durante a execução de um script. Por exemplo, o exemplo a seguir de variáveis de script passaria uma DATE variável <code>SAMPLE</code> e <code>FILTER_</code> para o Hive: <code>SAMPLE=s3://elasticmapreduce/samples/hive-ads</code> e <code>FILTER_DATE=#{format(@scheduledStartTime, 'YYYY-MM-dd')}</code> . Este campo aceita vários valores e funciona com os campos <code>script</code> e <code>scriptUri</code> . Além disso, o <code>scriptVariable</code> funciona independentemente do estágio estar definido como <code>true</code> ou <code>false</code> . Este campo é especialmente útil para enviar valores dinâmicos para o Hive usando expressões e funções do AWS Data Pipeline. | String |
| <code>stage</code> | Determina se a migração de dados está habilitada antes ou depois de executar o script. Não é permitido com o Hive 11, então use uma EMR AMI versão 3.2.0 ou superior da Amazon. | Booleano |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-------------------------------|--|--|
| <code>@activeInstances</code> | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, como "activeInstances": {"ref": "myRunnableObject Id"} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, como "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | Os registros de EMR etapas da Amazon estão disponíveis somente em tentativas de EMR atividade. | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| @healthStatusFromInstanceid | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término programado para um objeto. | DateTime |
| @scheduledStartTime | Horário de início programado para um objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, como "waitingOn": {"ref": "myRunnableObject Id"} |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [ShellCommandActivity](#)
- [EmrActivity](#)

HiveCopyActivity

Executa uma consulta do Hive em um EMR cluster. `HiveCopyActivity` facilita a cópia de dados entre tabelas do DynamoDB. `HiveCopyActivity` aceita uma instrução HiveQL para filtrar dados de entrada do DynamoDB no nível da coluna e da linha.

Exemplo

O exemplo a seguir mostra como usar `HiveCopyActivity` e `DynamoDBExportDataFormat` para copiar dados de um `DynamoDBDataNode` para outro ao filtrar dados com base em um time stamp.

```
{
  "objects": [
    {
      "id" : "DataFormat.1",
      "name" : "DataFormat.1",
      "type" : "DynamoDBExportDataFormat",
      "column" : "timeStamp BIGINT"
    },
    {
      "id" : "DataFormat.2",
      "name" : "DataFormat.2",
      "type" : "DynamoDBExportDataFormat"
    }
  ]
}
```

```

    },
    {
      "id" : "DynamoDBDataNode.1",
      "name" : "DynamoDBDataNode.1",
      "type" : "DynamoDBDataNode",
      "tableName" : "item_mapped_table_restore_temp",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.1" }
    },
    {
      "id" : "DynamoDBDataNode.2",
      "name" : "DynamoDBDataNode.2",
      "type" : "DynamoDBDataNode",
      "tableName" : "restore_table",
      "region" : "us_west_1",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.2" }
    },
    {
      "id" : "EmrCluster.1",
      "name" : "EmrCluster.1",
      "type" : "EmrCluster",
      "schedule" : { "ref" : "ResourcePeriod" },
      "masterInstanceType" : "m1.xlarge",
      "coreInstanceCount" : "4"
    },
    {
      "id" : "HiveTransform.1",
      "name" : "Hive Copy Transform.1",
      "type" : "HiveCopyActivity",
      "input" : { "ref" : "DynamoDBDataNode.1" },
      "output" : { "ref" : "DynamoDBDataNode.2" },
      "schedule" : { "ref" : "ResourcePeriod" },
      "runsOn" : { "ref" : "EmrCluster.1" },
      "filterSql" : "`timeStamp` > unix_timestamp(\"#{@scheduledStartTime}\", \"yyyy-MM-dd'T'HH:mm:ss\")"
    },
    {
      "id" : "ResourcePeriod",
      "name" : "ResourcePeriod",
      "type" : "Schedule",
      "period" : "1 Hour",
      "startDateTime" : "2013-06-04T00:00:00",
      "endDateTime" : "2013-06-04T01:00:00"
    }
  ]
}

```

```

    }
  ]
}
```

Sintaxe


| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|--|
| schedule | <p>Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.</p> | <p>Objeto de referência, por exemplo, "agenda": {"ref": "myScheduleId" }</p> |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|--|---|
| runsOn | Especifique o cluster de execução. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceId" } |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, será ignorado.workerGroup | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|--|--|
| attemptStatus | O status mais recente da atividade remota. | String |
| attemptTimeout | O tempo limite para a conclusão do trabalho remoto. Se definida, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| dependsOn | Especifica a dependência em outro objeto executável. | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId" } |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| filterSql | Um fragmento de SQL instrução do Hive que filtra um subconjunto de dados do DynamoDB ou do Amazon S3 para cópia. O filtro deve | String |

| Campos opcionais | Descrição | Tipo de slot |
|--------------------|--|--|
| | conter apenas predicados e não começar com uma WHERE cláusula, pois a AWS Data Pipeline adiciona automaticamente. | |
| input | A fonte de dados de entrada. Deve ser <code>S3DataNode</code> ou <code>DynamoDBDataNode</code> . Se você usar <code>DynamoDBNode</code> , especifique um <code>DynamoDBExportDataFormat</code> . | Objeto de referência, por exemplo, "input": {"ref": "myDataNodeId"} |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como <code>ondemand</code> . | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | A quantidade máxima de novas tentativas após uma falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|---|---|
| output | A fonte de dados de saída. Se a entrada for <code>S3DataNode</code> , a saída precisará ser <code>DynamoDBDataNode</code> . Caso contrário, ela poderá ser <code>S3DataNode</code> ou <code>DynamoDBDataNode</code> . Se você usar <code>DynamoDBNode</code> , especifique um <code>DynamoDBExportDataFormat</code> . | Objeto de referência, por exemplo, "output": {"ref": "myDataNodeId"} |
| parent | O pai do objeto atual do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| pipelineLogUri | O Amazon S3URI, por exemplo 's3://BucketName/Key/' , para fazer upload de registros para o pipeline. | String |
| postActivityTaskConfig | O Script de configuração pós-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, por exemplo, "postActivityTaskConfig": {"ref": "myShellScriptConfigId"} |
| preActivityTaskConfig | O script de configuração pré-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, por exemplo, "preActivityTaskConfig": {"ref": "myShellScriptConfigId"} |
| precondition | Opcionalmente define uma pré-condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "pré-condição": {"ref": "myPreconditionId"} |

| Campos opcionais | Descrição | Tipo de slot |
|---|---|--------------|
| <code>reportProgressTimeout</code> | O tempo limite para as chamadas sucessivas de trabalho remoto para <code>reportProgress</code> . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| <code>resizeClusterBeforeRunning</code> | Redimensione o cluster antes de executar esta atividade para acomodar nós de dados do DynamoDB especificados como entradas ou saídas. <div data-bbox="472 768 1149 1325" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note</p> <p>Se sua atividade usa a DynamoDBD <code>ataNode</code> como um nó de dados de entrada ou saída, e se você definir o <code>comoTRUE</code>, AWS Data Pipeline comece <code>resizeClusterBeforeRunning</code> a usar tipos de <code>m3.xlarge</code> instância . Isso substitui suas escolhas de tipo de instância por <code>m3.xlarge</code> , o que pode aumentar seus custos mensais.</p> </div> | Booleano |
| <code>resizeClusterMaxInstâncias</code> | Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento | Inteiro |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|--|--|
| scheduleType | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a ActivatePipeline operação para cada execução subsequente. Os valores são: cron, ondemand e timeseries. | Enumeração |
| Campos de tempo de execução | Descrição | Tipo de slot |
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id"} |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--|
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id"} |
| emrStepLog | Os registros de EMR etapas da Amazon estão disponíveis somente em tentativas de EMR atividade. | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstance | ID do último objeto da instância concluído. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez em que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [ShellCommandActivity](#)
- [EmrActivity](#)

PigActivity

PigActivity fornece suporte nativo para scripts Pig AWS Data Pipeline sem a necessidade de usar ShellCommandActivity ou EmrActivity. Além disso, PigActivity oferece suporte ao armazenamento de dados. Quando o campo de estágio é definido como verdadeiro, o AWS Data Pipeline prepara os dados de entrada como um esquema em Pig sem um código adicional do usuário.

Exemplo

O exemplo de pipeline a seguir mostra como usar PigActivity. O exemplo de pipeline a seguir executa as seguintes etapas:

- MyPigActivity1 carrega dados do Amazon S3 e executa um script Pig que seleciona algumas colunas de dados e os carrega no Amazon S3.
- MyPigActivity2 carrega a primeira saída, seleciona algumas colunas e três linhas de dados e a carrega no Amazon S3 como uma segunda saída.
- MyPigActivity3 carrega os segundos dados de saída, insere duas linhas de dados e somente a coluna chamada “quinta” na AmazonRDS.

- MyPigActivity4 carrega RDS dados da Amazon, seleciona a primeira linha de dados e os carrega no Amazon S3.

```
{
  "objects": [
    {
      "id": "MyInputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "directoryPath": "s3://example-bucket/pigTestInput",
      "name": "MyInputData1",
      "dataFormat": {
        "ref": "MyInputDataType1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyPigActivity4",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyOutputData3"
      },
      "pipelineLogUri": "s3://example-bucket/path/",
      "name": "MyPigActivity4",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "type": "PigActivity",
      "dependsOn": {
        "ref": "MyPigActivity3"
      },
      "output": {
        "ref": "MyOutputData4"
      },
      "script": "B = LIMIT ${input1} 1; ${output1} = FOREACH B GENERATE one;",
      "stage": "true"
    }
  ]
}
```

```

    "id": "MyPigActivity3",
    "scheduleType": "CRON",
    "schedule": {
      "ref": "MyEmrResourcePeriod"
    },
    "input": {
      "ref": "MyOutputData2"
    },
    "pipelineLogUri": "s3://example-bucket/path",
    "name": "MyPigActivity3",
    "runsOn": {
      "ref": "MyEmrResource"
    },
    "script": "B = LIMIT ${input1} 2; ${output1} = FOREACH B GENERATE Fifth;",
    "type": "PigActivity",
    "dependsOn": {
      "ref": "MyPigActivity2"
    },
    "output": {
      "ref": "MyOutputData3"
    },
    "stage": "true"
  },
  {
    "id": "MyOutputData2",
    "schedule": {
      "ref": "MyEmrResourcePeriod"
    },
    "name": "MyOutputData2",
    "directoryPath": "s3://example-bucket/PigActivityOutput2",
    "dataFormat": {
      "ref": "MyOutputDataType2"
    },
    "type": "S3DataNode"
  },
  {
    "id": "MyOutputData1",
    "schedule": {
      "ref": "MyEmrResourcePeriod"
    },
    "name": "MyOutputData1",
    "directoryPath": "s3://example-bucket/PigActivityOutput1",
    "dataFormat": {
      "ref": "MyOutputDataType1"
    }
  }

```



```

    },
    "type": "S3DataNode"
  },
  {
    "id": "MyInputDataType1",
    "name": "MyInputDataType1",
    "column": [
      "First STRING",
      "Second STRING",
      "Third STRING",
      "Fourth STRING",
      "Fifth STRING",
      "Sixth STRING",
      "Seventh STRING",
      "Eighth STRING",
      "Ninth STRING",
      "Tenth STRING"
    ],
    "inputRegex": "^(\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+) (\\\\\\\\S+)",
    "type": "Regex"
  },
  {
    "id": "MyEmrResource",
    "region": "us-east-1",
    "schedule": {
      "ref": "MyEmrResourcePeriod"
    },
    "keyPair": "example-keypair",
    "masterInstanceType": "m1.small",
    "enableDebugging": "true",
    "name": "MyEmrResource",
    "actionOnTaskFailure": "continue",
    "type": "EmrCluster"
  },
  {
    "id": "MyOutputDataType4",
    "name": "MyOutputDataType4",
    "column": "one STRING",
    "type": "CSV"
  },
  {
    "id": "MyOutputData4",
    "schedule": {

```

```

    "ref": "MyEmrResourcePeriod"
  },
  "directoryPath": "s3://example-bucket/PigActivityOutput3",
  "name": "MyOutputData4",
  "dataFormat": {
    "ref": "MyOutputDataType4"
  },
  "type": "S3DataNode"
},
{
  "id": "MyOutputDataType1",
  "name": "MyOutputDataType1",
  "column": [
    "First STRING",
    "Second STRING",
    "Third STRING",
    "Fourth STRING",
    "Fifth STRING",
    "Sixth STRING",
    "Seventh STRING",
    "Eighth STRING"
  ],
  "columnSeparator": "*",
  "type": "Custom"
},
{
  "id": "MyOutputData3",
  "username": "__",
  "schedule": {
    "ref": "MyEmrResourcePeriod"
  },
  "insertQuery": "insert into #{table} (one) values (?)",
  "name": "MyOutputData3",
  "*password": "__",
  "runsOn": {
    "ref": "MyEmrResource"
  },
  "connectionString": "jdbc:mysql://example-database-instance:3306/example-database",
  "selectQuery": "select * from #{table}",
  "table": "example-table-name",
  "type": "MySQLDataNode"
},
{

```

```

    "id": "MyOutputDataType2",
    "name": "MyOutputDataType2",
    "column": [
      "Third STRING",
      "Fourth STRING",
      "Fifth STRING",
      "Sixth STRING",
      "Seventh STRING",
      "Eighth STRING"
    ],
    "type": "TSV"
  },
  {
    "id": "MyPigActivity2",
    "scheduleType": "CRON",
    "schedule": {
      "ref": "MyEmrResourcePeriod"
    },
    "input": {
      "ref": "MyOutputData1"
    },
    "pipelineLogUri": "s3://example-bucket/path",
    "name": "MyPigActivity2",
    "runsOn": {
      "ref": "MyEmrResource"
    },
    "dependsOn": {
      "ref": "MyPigActivity1"
    },
    "type": "PigActivity",
    "script": "B = LIMIT ${input1} 3; ${output1} = FOREACH B GENERATE Third, Fourth,
    Fifth, Sixth, Seventh, Eighth;",
    "output": {
      "ref": "MyOutputData2"
    },
    "stage": "true"
  },
  {
    "id": "MyEmrResourcePeriod",
    "startDateTime": "2013-05-20T00:00:00",
    "name": "MyEmrResourcePeriod",
    "period": "1 day",
    "type": "Schedule",
    "endDateTime": "2013-05-21T00:00:00"
  }
}

```

```

    },
    {
      "id": "MyPigActivity1",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyInputData1"
      },
      "pipelineLogUri": "s3://example-bucket/path",
      "scriptUri": "s3://example-bucket/script/pigTestScript.q",
      "name": "MyPigActivity1",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "scriptVariable": [
        "column1=First",
        "column2=Second",
        "three=3"
      ],
      "type": "PigActivity",
      "output": {
        "ref": "MyOutputData1"
      },
      "stage": "true"
    }
  ]
}

```

O conteúdo de `pigTestScript.q` é o seguinte.

```

B = LIMIT ${input1} $three; ${output1} = FOREACH B GENERATE $column1, $column2, Third,
Fourth, Fifth, Sixth, Seventh, Eighth;

```

Sintaxe

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|-------------------------------------|
| <code>schedule</code> | Esse objeto é invocado durante a execução de um intervalo de programação. Os usuários | Objeto de referênci a, por exemplo, |


| Campos de invocação de objetos | Descrição | Tipo de slot |
|--|--|---|
| | <p>precisam especificar uma referência de programação para outro objeto de modo a definir a ordem de execução de dependência desse objeto. Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando "agenda": {"ref": "DefaultSchedule"}. Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), os usuários poderão criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.</p> | <pre>"schedule": {"ref": "myScheduleId" }</pre> |
| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
| script | O script do Pig a ser executado. | String |
| scriptUri | A localização do script Pig a ser executado (por exemplo, s3://scriptLocation). | String |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|--|---|
| runsOn | EMRCluster no qual isso PigActivity é executado. | Objeto de referência, por exemplo, "runsOn": {"ref": "myEmrCluster Id"} |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, será ignorado.workerGroup | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|--|---|
| attemptStatus | O status mais recente da atividade remota. | String |
| attemptTimeout | O tempo limite para a conclusão do trabalho remoto. Se definida, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| dependsOn | Especifica a dependência em outro objeto executável. | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId"} |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| input | A fonte de dados de entrada. | Objeto de referência, por exemplo, "input": |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------------------|---|--|
| | | <code>{"ref": "myDataNode Id"}</code> |
| <code>lateAfterTimeout</code> | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como <code>ondemand</code> . | Período |
| <code>maxActiveInstances</code> | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| <code>maximumRetries</code> | A quantidade máxima de novas tentativas após uma falha. | Inteiro |
| <code>onFail</code> | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, <code>"onFail": {"ref": "myActionId"}</code> |
| <code>onLateAction</code> | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, <code>"onLateAction": {"ref": "myActionId"}</code> |
| <code>onSuccess</code> | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, <code>"onSuccess": {"ref": "myActionId"}</code> |
| <code>output</code> | A fonte de dados de saída. | Objeto de referência, por exemplo, <code>"output": {"ref": "myDataNode Id"}</code> |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|---|---|
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| pipelineLogUri | O Amazon S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| postActivityTaskConfig | Script de configuração pós-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, por exemplo, "postActivityTaskConfig": {"ref": "myShellScriptConfigId"} |
| preActivityTaskConfig | Script de configuração pré-atividade a ser executado. Isso consiste em um script URI de shell no Amazon S3 e uma lista de argumentos. | Objeto de referência, por exemplo, "preActivityTaskConfig": {"ref": "myShellScriptConfigId"} |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "pre-condition": {"ref": "myPreconditionId"} |
| reportProgressTimeout | O tempo limite para as chamadas sucessivas de trabalho remoto para reportProgress . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|---|--|--------------|
| <code>resizeClusterBeforeRunning</code> | <p>Redimensione o cluster antes de executar esta atividade para acomodar nós de dados do DynamoDB especificados como entradas ou saídas.</p> <div data-bbox="472 447 1149 1003" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Se sua atividade usa a <code>DynamoDB</code> <code>ataNode</code> como um nó de dados de entrada ou saída, e se você definir o <code>comoTRUE</code>, AWS Data Pipeline começa <code>resizeClusterBeforeRunning</code> a usar tipos de <code>m3.xlarge</code> instância. Isso substitui suas escolhas de tipo de instância por <code>m3.xlarge</code>, o que pode aumentar seus custos mensais.</p> </div> | Booleano |
| <code>resizeClusterMaxInstâncias</code> | Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento. | Inteiro |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| <code>scheduleType</code> | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Programação com estilo de séries temporais significa que as instâncias são programadas no final de cada intervalo, e Programação com estilo Cron significa que as instâncias são programadas no início de cada intervalo. Uma programação sob demanda permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão e deverá ser a única <code>scheduleType</code> especificada para objetos no pipeline. Para usar pipelines sob demanda, basta chamar a <code>ActivatePipeline</code> operação para cada execução subsequente. Os valores são: <code>cron</code> , <code>ondemand</code> e <code>timeseries</code> . | Enumeração |
| <code>scriptVariable</code> | Os argumentos a serem transmitidos para o script do Pig. Você pode usar <code>scriptVariable</code> com <code>script</code> ou <code>scriptUri</code> . | String |
| <code>stage</code> | Determina se a preparação está ativada e permite que seu script Pig tenha acesso às tabelas de dados preparados, como <code>\$ {INPUT1}</code> e <code>\$ {}</code> . <code>OUTPUT1</code> | Booleano |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | Os registros de EMR etapas da Amazon estão disponíveis somente em tentativas de EMR atividade. | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceid | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término programado para o objeto. | DateTime |
| @scheduledStartTime | Horário de início programado para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |
| Campos do sistema | Descrição | Tipo de slot |
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [ShellCommandActivity](#)
- [EmrActivity](#)

RedshiftCopyActivity

Copia uma tabela do DynamoDB ou Amazon S3 para o Amazon Redshift. Você pode carregar dados em uma nova tabela ou mesclar dados em uma tabela existente de maneira fácil.

Esta é uma visão geral de um caso de uso no qual usar `RedshiftCopyActivity`:

1. Comece usando AWS Data Pipeline para armazenar seus dados no Amazon S3.
2. Use `RedshiftCopyActivity` para mover os dados da Amazon RDS e da Amazon EMR para o Amazon Redshift.

Isso permite que você carregue seus dados no Amazon Redshift, onde pode analisá-los.

3. Use [SqlActivity](#) para realizar SQL consultas nos dados que você carregou no Amazon Redshift.

Além disso, `RedshiftCopyActivity` permite que você trabalhe com um `S3DataNode`, já que ele oferece suporte a um arquivo manifesto. Para obter mais informações, consulte [S3 DataNode](#).

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

Para garantir a conversão de formatos, este exemplo usa `EMPTYASNULL` parâmetros de conversão `IGNOREBLANKLINES` especiais em `commandOptions`. Para obter informações, consulte [Parâmetros de conversão de dados](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

```
{
  "id" : "S3ToRedshiftCopyActivity",
  "type" : "RedshiftCopyActivity",
  "input" : { "ref": "MyS3DataNode" },
  "output" : { "ref": "MyRedshiftDataNode" },
  "insertMode" : "KEEP_EXISTING",
  "schedule" : { "ref": "Hour" },
  "runsOn" : { "ref": "MyEc2Resource" },
  "commandOptions": ["EMPTYASNULL", "IGNOREBLANKLINES"]
}
```

A definição de pipeline de exemplo a seguir mostra uma atividade que usa o modo de inserção `APPEND`:

```
{
  "objects": [
    {
      "id": "CSVId1",
      "name": "DefaultCSV1",
      "type": "CSV"
    },
    {
      "id": "RedshiftDatabaseId1",
      "databaseName": "dbname",
      "username": "user",
      "name": "DefaultRedshiftDatabase1",
      "*password": "password",
    }
  ]
}
```

```

    "type": "RedshiftDatabase",
    "clusterId": "redshiftclusterId"
  },
  {
    "id": "Default",
    "scheduleType": "timeseries",
    "failureAndRerunMode": "CASCADE",
    "name": "Default",
    "role": "DataPipelineDefaultRole",
    "resourceRole": "DataPipelineDefaultResourceRole"
  },
  {
    "id": "RedshiftDataNodeId1",
    "schedule": {
      "ref": "ScheduleId1"
    },
    "tableName": "orders",
    "name": "DefaultRedshiftDataNode1",
    "createTableSql": "create table StructuredLogs (requestBeginTime CHAR(30)
PRIMARY KEY DISTKEY SORTKEY, requestEndTime CHAR(30), hostname CHAR(100), requestDate
varchar(20));",
    "type": "RedshiftDataNode",
    "database": {
      "ref": "RedshiftDatabaseId1"
    }
  },
  {
    "id": "Ec2ResourceId1",
    "schedule": {
      "ref": "ScheduleId1"
    },
    "securityGroups": "MySecurityGroup",
    "name": "DefaultEc2Resource1",
    "role": "DataPipelineDefaultRole",
    "logUri": "s3://myLogs",
    "resourceRole": "DataPipelineDefaultResourceRole",
    "type": "Ec2Resource"
  },
  {
    "id": "ScheduleId1",
    "startDateTime": "yyyy-mm-ddT00:00:00",
    "name": "DefaultSchedule1",
    "type": "Schedule",
    "period": "period",

```

```

    "endTime": "yyyy-mm-ddT00:00:00"
  },
  {
    "id": "S3DataNodeId1",
    "schedule": {
      "ref": "ScheduleId1"
    },
    "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
    "name": "DefaultS3DataNode1",
    "dataFormat": {
      "ref": "CSVId1"
    },
    "type": "S3DataNode"
  },
  {
    "id": "RedshiftCopyActivityId1",
    "input": {
      "ref": "S3DataNodeId1"
    },
    "schedule": {
      "ref": "ScheduleId1"
    },
    "insertMode": "APPEND",
    "name": "DefaultRedshiftCopyActivity1",
    "runsOn": {
      "ref": "Ec2ResourceId1"
    },
    "type": "RedshiftCopyActivity",
    "output": {
      "ref": "RedshiftDataNodeId1"
    }
  }
]
}

```

APPEND A operação adiciona itens a uma tabela, independentemente das chaves principais ou de classificação. Por exemplo, se você tiver a tabela a seguir, poderá anexar um registro com o mesmo ID e o valor de usuário.

| ID(PK) | USER |
|--------|------|
| 1 | aaa |
| 2 | bbb |

Você pode anexar um registro com o mesmo ID e valor de usuário:

| ID(PK) | USER |
|--------|------|
| 1 | aaa |
| 2 | bbb |
| 1 | aaa |

Note

Se uma operação APPEND é interrompida e realizada novamente, a nova execução resultante do pipeline pode acrescentar linhas desde o início. Isso pode causar uma duplicação. Por isso, você deve estar ciente desse comportamento, especialmente se houver alguma lógica que conta o número de linhas.

Para ver um tutorial, consulte [Copiar dados para o Amazon Redshift usando AWS Data Pipeline](#).

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| insertMode | <p>Determina o que AWS Data Pipeline acontece com os dados preexistentes na tabela de destino que se sobrepõem às linhas nos dados a serem carregados.</p> <p>Os valores válidos são: KEEP_EXISTING , OVERWRITE_EXISTING , TRUNCATE e APPEND.</p> <p>KEEP_EXISTING adiciona novas linhas à tabela deixando quaisquer linhas existentes sem modificações.</p> <p>KEEP_EXISTING e OVERWRITE_EXISTING usam as chaves primária, de classificação e de distribuição para identificar quais linhas de entrada correspondem a linhas</p> | Enumeração |

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| | <p>existentes. Consulte Atualizar e inserir novos dados no Guia do desenvolvedor de banco de dados do Amazon Redshift.</p> <p>TRUNCATE exclui todos os dados na tabela de destino antes de gravar os novos dados.</p> <p>APPEND adiciona todos os registros ao final da tabela do Redshift. APPEND não requer uma chave de distribuição primária ou uma chave de classificação de modo que itens que podem ser possíveis duplicatas podem ser anexados.</p> | |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|--|
| schedule | <p>Esse objeto é invocado durante a execução de um intervalo de programação.</p> <p>Especifique uma referência de programação para outro objeto para definir a ordem de execução de dependência desse objeto.</p> <p>Na maioria dos casos, recomendamos colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Por exemplo, você pode definir uma programação explicitamente no objeto especificando "schedule": {"ref": "DefaultSchedule"} .</p> <p>Se a programação principal do seu pipeline contiver programações aninhadas, crie um objeto pai que tenha uma referência de programação.</p> | <p>Objeto de referência, como: "schedule": {"ref": "myScheduleId"}</p> |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--|---|---|
| | Para obter mais informações sobre configurações opcionais de programação de exemplo, consulte Programação . | |
| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
| runsOn | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceId" } |
| workerGroup | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um runsOn valor e workerGroup existir, ele será workerGroup ignorado. | String |
| Campos opcionais | Descrição | Tipo de slot |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se definida, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| commandOptions | Pega parâmetros para passar para o nó de dados do Amazon Redshift durante a operação COPY. Para obter informações sobre parâmetros, consulte COPY o Amazon Redshift Database Developer Guide. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|--------------|
| | <p>À medida que carrega a tabela, COPY tenta converter implicitamente as strings no tipo de dados da coluna de destino. Além das conversões de dados padrão que são realizadas automaticamente, se você receber erros ou tiver outras necessidades de conversão, especifique parâmetros de conversão adicionais. Para obter informações, consulte Parâmetros de conversão de dados no Guia do desenvolvedor de banco de dados do Amazon Redshift.</p> <p>Se um formato de dados é associado ao nó de dados de entrada ou saída, os parâmetros fornecidos são ignorados.</p> <p>Como a operação de cópia usa COPY para inserir dados em uma tabela de preparação e, em seguida, usa um comando INSERT para copiar os dados da tabela de preparação para a tabela de destino, alguns parâmetros COPY não se aplicam, como a capacidade do comando COPY para permitir a compactação automática da tabela. Se a compactação for necessária, adicione detalhes de codificação de coluna na instrução CREATE TABLE.</p> <p>Além disso, em alguns casos, quando é preciso descarregar os dados do cluster do Amazon Redshift e criar arquivos no Amazon S3, a <code>RedshiftCopyActivity</code> depende da operação UNLOAD do Amazon Redshift.</p> <p>Para melhorar o desempenho ao copiar e descarregar, especifique o parâmetro <code>PARALLEL OFF</code> do comando UNLOAD. Para obter informações sobre parâmetros, consulte</p> | |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--|
| | UNLOAD do Amazon Redshift Database Developer Guide. | |
| dependsOn | Especifique a dependência em outro objeto executável. | Objeto de referência: "dependsOn": {"ref": "myActivityId"} |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| input | O nó de dados de entrada. A fonte de dados pode ser o Amazon S3, o DynamoDB ou o Amazon Redshift. | Objeto de referência: "input": {"ref": "myDataNodeId"} |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência: "onFail": {"ref": "myActionId"} |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência: "onLateAction": { "ref": "myActionId" } |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência: "onSuccess": { "ref": "myActionId" } |
| output | O nó de dados de saída. A localização de saída pode ser o Amazon S3 ou o Amazon Redshift. | Objeto de referência: "output": { "ref": "myDataNodeId" } |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência: "parent": { "ref": "myBaseObjectId" } |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência: "precondition": { "ref": "myPreconditionId" } |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|---|--------------|
| queue (fila) | <p>Corresponde à configuração <code>query_group</code> no Amazon Redshift, que permite atribuir e priorizar atividades simultâneas com base em sua colocação em filas.</p> <p>O Amazon Redshift limita o número de conexões simultâneas a 15. Para obter mais informações, consulte Atribuição de consultas a filas no Amazon RDS Database Developer Guide.</p> | String |
| reportProgressTimeout | <p>Tempo limite para as chamadas sucessivas de trabalho remoto para <code>reportProgress</code>.</p> <p>Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente.</p> | Período |
| retryDelay | <p>A duração do tempo limite entre duas novas tentativas.</p> | Período |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------------|--|--------------|
| <code>scheduleType</code> | <p>Permite que você especifique a programação para objetos no pipeline. Os valores são: <code>cron</code>, <code>ondemand</code> e <code>timeseries</code> .</p> <p>A programação <code>timeseries</code> significa que as instâncias são programadas no final de cada intervalo.</p> <p>A programação <code>Cron</code> significa que as instâncias são programadas no início de cada intervalo.</p> <p>Uma programação <code>ondemand</code> permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente.</p> <p>Para usar pipelines <code>ondemand</code>, chame a operação <code>ActivatePipeline</code> para cada execução subsequente.</p> <p>Se você usar uma programação <code>ondemand</code>, deverá especificá-la no objeto padrão, e este deverá ser o único <code>scheduleType</code> especificado para objetos no pipeline.</p> | Enumeração |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------------|---|--------------|
| <code>transformSql</code> | <p>A expressão SQL <code>SELECT</code> usada para transformar os dados de entrada.</p> <p>Execute a expressão <code>transformSql</code> na tabela chamada <code>staging</code>.</p> <p>Ao copiar dados do DynamoDB ou do Amazon S3, o AWS Data Pipeline cria uma tabela chamada "staging" e, inicialmente, carrega dados nesta tabela. Os dados dessa tabela são usados para atualizar a tabela de destino.</p> <p>O esquema de saída de <code>transformSql</code> deve corresponder ao esquema da tabela de destinos finais.</p> <p>Se você especificar a <code>transformSql</code> opção, uma segunda tabela de preparação será criada a partir da SQL instrução especificada. Os dados na segunda tabela de preparação são, então, atualizados na tabela de destino final.</p> | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-------------------------------|--|---|
| <code>@activeInstances</code> | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência: <code>"activeInstances": {"ref": "myRunnableObjectId"}</code> |
| <code>@actualEndTime</code> | Hora em que a execução deste objeto foi concluída. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--|
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência: "cascadeFailedOn": {"ref": "myRunnableObjectId"} |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstance | ID do último objeto da instância concluído. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência: "waitingOn": {"ref": "myRunnableObjectId"} |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto. Denota seu lugar no ciclo de vida. Por exemplo, objetos de componentes dão origem a objetos de instância, que executam objetos de tentativa. | String |

ShellCommandActivity

Executa um comando ou script. Você pode usar `ShellCommandActivity` para executar séries temporais ou tarefas programadas parecidas com Cron.

Quando o `stage` campo é definido como verdadeiro e usado com um `S3DataNode`, `ShellCommandActivity` suporta o conceito de armazenamento temporário de dados, o que significa que você pode mover dados do Amazon S3 para um local de estágio, como a EC2 Amazon ou seu ambiente local, realizar trabalhos nos dados usando scripts e o. e movê-los de volta para `ShellCommandActivity` o Amazon S3.

Nesse caso, quando o comando `shell` está conectado a uma entrada `S3DataNode`, os scripts `shell` operam diretamente nos dados usando `${INPUT1_STAGING_DIR}`, `${INPUT2_STAGING_DIR}` e outros campos, referindo aos campos de entrada `ShellCommandActivity`.

Da mesma forma, a saída do comando de `shell` pode ser preparada em um diretório de saída para ser automaticamente enviada ao Amazon S3, referenciada por `${OUTPUT1_STAGING_DIR}`, `${OUTPUT2_STAGING_DIR}` e assim por diante.

Essas expressões podem passar como argumentos de linha de comando para o comando de `shell` para que você possa usá-las na lógica de transformação de dados.

`ShellCommandActivity` retorna códigos de erro e strings no estilo do Linux. Se `ShellCommandActivity` resulta em um erro, o `error` retornado é um valor diferente de zero.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "CreateDirectory",
  "type" : "ShellCommandActivity",
  "command" : "mkdir new-directory"
}
```

Sintaxe

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|---|
| schedule | <p>Esse objeto é invocado durante a execução de um intervalo <code>schedule</code>.</p> <p>Para definir a ordem de execução de dependência desse objeto, especifique uma referência <code>schedule</code> a outro objeto.</p> <p>Para atender a esse requisito, defina explicitamente um <code>schedule</code> no objeto, por exemplo, especificando <code>"schedule": {"ref": "DefaultSchedule"}</code> .</p> <p>Na maioria dos casos, é melhor colocar a referência <code>schedule</code> no objeto de pipeline padrão para que todos os objetos herdem essa programação. Se o pipeline consiste em uma árvore de programações (programações aninhadas na programação principal), crie um objeto pai que tenha uma referência de programação.</p> <p>Para distribuir a carga, AWS Data Pipeline cria objetos físicos um pouco antes do previsto, mas os executa dentro do cronograma.</p> <p>Para obter mais informações sobre o exemplo de configurações opcionais de programação,</p> | Objeto de referência, por exemplo, <code>"agenda": {"ref": "myScheduleId"}</code> |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--|---|---|
| | consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html . | |
| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
| command | O comando a ser executado. Use \$ para fazer referência aos parâmetros posicionais e <code>scriptArgument</code> para especificar os parâmetros para o comando. Este valor e quaisquer parâmetros associados precisam funcionar no ambiente do qual você está executando o Task Runner. | String |
| scriptUri | Um URI caminho do Amazon S3 para um arquivo ser baixado e executado como um comando shell. Especifique somente um campo <code>scriptUri</code> ou <code>command</code> . <code>scriptUri</code> não pode usar parâmetros, portanto, em vez disso, use <code>command</code> . | String |
| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
| runsOn | O recurso computacional para executar a atividade ou o comando, por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, "runsOn": {"ref": "myResourceId" } |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|--|--------------|
| workerGroup | Usado para tarefas de roteamento. Se você fornecer um valor de <code>runsOn</code> e <code>workerGroup</code> existir, será ignorado. <code>workerGroup</code> | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--|
| attemptStatus | O status mais recente da atividade remota. | String |
| attemptTimeout | O tempo limite para conclusão do trabalho remoto. Se definido, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| dependsOn | Especifica uma dependência em outro objeto executável. | Objeto de referência, por exemplo, "dependsOn": {"ref": "myActivityId" } |
| failureAndRerunMode | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| input | O local dos dados de entrada. | Objeto de referência, por exemplo, "input": {"ref": "myDataNodeId" } |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como <code>ondemand</code> . | Período |

| Campos opcionais | Descrição | Tipo de slot |
|--------------------|---|--|
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | A quantidade máxima de novas tentativas após uma falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi programado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| output | O local dos dados de saída. | Objeto de referência, por exemplo, "output": {"ref": "myDataNodeId"} |
| parent | O pai do objeto atual do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| pipelineLogUri | O Amazon S3URI, como 's3://BucketName/Key/' para fazer upload de registros para o pipeline. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------------------|--|--|
| <code>precondition</code> | Opcionalmente define uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "pré-condição": {"ref": "myPreconditionId"} |
| <code>reportProgressTimeout</code> | O tempo limite para chamadas sucessivas para <code>reportProgress</code> por atividades remotas. Se configurada, as atividades remotas sem progresso para o período especificado poderão ser consideradas como interrompidas e serão executadas novamente. | Período |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------------|---|--------------|
| <code>scheduleType</code> | <p>Permite que você especifique se os objetos na definição do pipeline devem ser programados no início ou no final do intervalo.</p> <p>Os valores possíveis são: <code>cron</code>, <code>ondemand</code> e <code>timeseries</code> .</p> <p>Se definido como <code>timeseries</code> , as instâncias são programadas no final de cada intervalo.</p> <p>Se definido como <code>Cron</code>, as instâncias são programadas no início de cada intervalo.</p> <p>Se definido como <code>ondemand</code>, você pode executar um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente . Se você usar uma programação <code>ondemand</code>, deverá especificá-la no objeto padrão como o único <code>scheduleType</code> para objetos no pipeline. Para usar pipelines <code>ondemand</code>, chame a operação <code>ActivatePipeline</code> para cada execução subsequente.</p> | Enumeração |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| <code>scriptArgument</code> | Uma matriz JSON formatada de strings para passar para o comando especificado pelo comando. Por exemplo, se o comando for <code>echo \$1 \$2</code> , especifique <code>scriptArgument</code> como <code>"param1", "param2"</code> . Para vários argumentos e parâmetros, passe o <code>scriptArgument</code> da seguinte forma: <code>"scriptArgument": "arg1", "scriptArgument": "param1", "scriptArgument": "arg2", "scriptArgument": "param2"</code> . O <code>scriptArgument</code> só pode ser usado com <code>command</code> . Usá-lo com <code>scriptUri</code> causa um erro. | String |
| <code>stage</code> | Determina se a preparação está ou não ativada e permite que os comandos shell tenham acesso às variáveis de dados preparados, como <code>\${INPUT1_STAGING_DIR}</code> e <code>\${OUTPUT1_STAGING_DIR}</code> . | Booleano |
| <code>stderr</code> | O caminho do que recebe mensagens de erro do sistema redirecionadas do comando. Se você usar o campo <code>runsOn</code> , ele precisará ser um caminho do Amazon S3 devido à natureza transitória do recurso que está executando sua atividade. No entanto, se você especificar o campo <code>workerGroup</code> , poderá usar um caminho de arquivo local. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| stdout | O caminho do Amazon S3 que recebe saídas redirecionadas do comando. Se você usar o campo <code>runsOn</code> , ele precisará ser um caminho do Amazon S3 devido à natureza transitória do recurso que está executando sua atividade. No entanto, se você especificar o campo <code>workerGroup</code> , poderá usar um caminho de arquivo local. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--|
| @activeInstances | A lista dos objetos da instância ativa programados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id"} |
| @actualEndTime | O horário em que a execução desse objeto foi concluída. | DateTime |
| @actualStartTime | O horário em que a execução desse objeto foi iniciada. | DateTime |
| cancellationReason | O <code>cancellationReason</code> se esse objeto foi cancelado. | String |
| @cascadeFailedOn | A descrição da cadeia de dependências que causou a falha no objeto. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id"} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| emrStepLog | Os registros de EMR etapas da Amazon estão disponíveis somente em tentativas de EMR atividade na Amazon. | String |
| errorId | O <code>errorId</code> se esse objeto apresentou falha. | String |
| errorMessage | O <code>errorMessage</code> se esse objeto apresentou falha. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | O horário em que a execução do objeto foi concluída. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis sobre tentativas de atividades EMR baseadas na Amazon. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceid | O ID do último objeto de instância que entrou em um estado concluído. | String |
| @healthStatusUpdated Hora | O horário em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome de host do cliente que pegou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @latestCompletedRun Hora | O horário da última execução concluída. | DateTime |
| @latestRunTime | O horário da última execução programada. | DateTime |
| @nextRunTime | O horário da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez em que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | O horário de término programado para o objeto. | DateTime |
| @scheduledStartTime | O horário de início programado para o objeto. | DateTime |
| @status | O status do objeto. | String |
| @version | A AWS Data Pipeline versão usada para criar o objeto. | String |
| @waitingOn | A descrição da lista de dependências pelas quais esse objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | O erro ao descrever o objeto malformatado. | String |
| @pipelineId | O ID do pipeline ao qual esse objeto pertence. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @sphere | O local de um objeto no ciclo de vida. Objetos de componentes dão origem a objetos de instância, que executam objetos de tentativa. | String |

Consulte também

- [CopyActivity](#)
- [EmrActivity](#)

SqlActivity

Executa uma SQL consulta (script) em um banco de dados.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MySqlActivity",
  "type" : "SqlActivity",
  "database" : { "ref": "MyDatabaseID" },
  "script" : "SQLQuery" | "scriptUri" : s3://scriptBucket/query.sql,
  "schedule" : { "ref": "MyScheduleID" },
}
```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|---|
| banco de dados | O banco de dados no qual executar o SQL script fornecido. | Objeto de referência, por exemplo, "banco de dados": {"ref": "myDatabaseId" } |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|---|---|
| schedule | <p>Esse objeto é invocado durante a execução de um intervalo de programação. Você deve especificar uma referência de programação para outro objeto para definir a ordem de execução de dependência desse objeto. Você pode definir uma programação explicitamente no objeto, por exemplo, especificando <code>"schedule": {"ref": "DefaultSchedule"}</code> .</p> <p>Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação.</p> <p>Se o pipeline tiver uma árvore de programações aninhada na programação principal, crie um objeto pai que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.</p> | Objeto de referência, por exemplo, <code>"agenda": {"ref": "myScheduleId"}</code> |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|--|--------------|
| script | O SQL script a ser executado. Você deve especificar o script <code>scriptUri</code> . Quando o script é armazenado no Amazon S3, o script não é avaliado como uma expressão. Especific | String |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|---|--|--------------|
| | ar vários valores para <code>scriptArgument</code> é útil quando o script é armazenado no Amazon S3. | |
| <code>scriptUri</code> | A URI especificando a localização de um SQL script a ser executado nessa atividade. | String |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|---|--|--|
| <code>runsOn</code> | O recurso computacional para executar a atividade ou o comando. Por exemplo, uma EC2 instância da Amazon ou um EMR cluster da Amazon. | Objeto de referência, por exemplo, <code>"runsOn": {"ref": "myResourceId" }</code> |
| <code>workerGroup</code> | O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de <code>runsOn</code> e <code>workerGroup</code> existir, será ignorado. <code>workerGroup</code> | String |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|---|------------------------------------|
| <code>attemptStatus</code> | Status mais recente da atividade remota. | String |
| <code>attemptTimeout</code> | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| <code>dependsOn</code> | Especifique a dependência em outro objeto executável. | Objeto de referência, por exemplo, |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|---|
| | | "dependsOn": {"ref": "myActivityId" } |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| input | Local dos dados de entrada. | Objeto de referência, por exemplo, "input": {"ref": "myDataNode Id" } |
| lateAfterTimeout | O período desde o início programado do pipeline no qual a execução do objeto deve começar. | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId" } |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não tiver sido programado ou ainda não tiver sido concluído no período de tempo desde o início programado do pipeline, conforme especificado por 'lateAfterTimeout'. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId" } |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId" } |
| output | Local dos dados de saída. Isso só é útil para fazer referência a partir de um script (por exemplo <code>#{output.tablename}</code>) e para criar a tabela de saída definindo 'createTableSql' no nó de dados de saída. A saída da SQL consulta não é gravada no nó de dados de saída. | Objeto de referência, por exemplo, "output": {"ref": "myDataNodeId" } |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId" } |
| pipelineLogUri | O S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| precondition | Se desejar, você pode definir uma condição. Um nó de dados não é marcado como "READY" até que todas as condições prévias tenham sido atendidas. | Objeto de referência, por exemplo, "precondition": {"ref": "myPreconditionId" } |
| queue (fila) | [Apenas para o Amazon Redshift] Corresponde à configuração <code>query_group</code> no Amazon Redshift, que permite atribuir e priorizar atividades simultâneas com base em sua colocação em filas. O Amazon Redshift limita o número de conexões simultâneas a 15. Para obter mais informações, consulte Atribuir consultas a filas no Guia do desenvolvedor de banco de dados do Amazon Redshift. | String |

| Campos opcionais | Descrição | Tipo de slot |
|-------------------------------------|--|--------------|
| <code>reportProgressTime out</code> | Tempo limite para trabalho remoto: chamadas sucessivas para <code>reportProgress</code> . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |
| <code>scheduleType</code> | <p>O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou no final do intervalo. Os valores são: <code>cron</code>, <code>ondemand</code> e <code>timeseries</code>.</p> <p>A programação <code>timeseries</code> significa que as instâncias são programadas no final de cada intervalo.</p> <p>A programação <code>cron</code> significa que as instâncias são programadas no início de cada intervalo.</p> <p>Uma programação <code>ondemand</code> permite que você execute um pipeline uma vez por ativação. Isso significa que você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação <code>ondemand</code>, ela precisará ser especificada no objeto padrão, além de ser a única <code>scheduleType</code> especificada para objetos no pipeline. Para usar pipelines <code>ondemand</code>, chame a operação <code>ActivatePipeline</code> para cada execução subsequente.</p> | Enumeração |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| scriptArgument | Uma lista de variáveis do script. Além disso, você pode colocar expressões diretamente no campo do script. Vários valores para scriptArgument são úteis quando o script é armazenado no Amazon S3. Exemplo: <code># {format (@scheduledStartTime, "YY-MM-DD HH:MM:SS")\n# {format (plusPeriod(@, "1 dia")scheduledStartTime, "YY-MM-DD HH:MM:SS")</code> | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstance | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Recursos

A seguir estão os objetos AWS Data Pipeline de recursos:

Objetos

- [Ec2Resource](#)
- [EmrCluster](#)
- [HttpProxy](#)

Ec2Resource

Uma EC2 instância da Amazon que executa o trabalho definido por uma atividade de pipeline.

AWS Data Pipeline agora é compatível com a EC2 instância Amazon, que usa um método orientado IMDSv2 por sessão para lidar melhor com a autenticação ao recuperar informações de metadados das instâncias. Uma sessão inicia e encerra uma série de solicitações que o software executado em uma EC2 instância da Amazon usa para acessar os metadados e as credenciais da instância EC2 Amazon armazenados localmente. O software inicia uma sessão com uma simples HTTP PUT solicitação paraIMDSv2. IMDSv2retorna um token secreto para o software executado na EC2 instância da Amazon, que usará o token como senha para fazer solicitações de metadados e credenciais. IMDSv2

Note

IMDSv2Para usar em sua EC2 instância Amazon, você precisará modificar as configurações, pois o padrão não AMI é compatível comIMDSv2. Você pode especificar uma nova AMI versão que pode ser recuperada por meio do seguinte SSM parâmetro:`/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs`.

Para obter informações sobre EC2 instâncias padrão da Amazon que são AWS Data Pipeline criadas se você não especificar uma instância, consulte [Instâncias do Amazon EC2 padrão por região da AWS](#).

Exemplos

EC2-Clássico

⚠ Important

Somente AWS contas criadas antes de 4 de dezembro de 2013 oferecem suporte à plataforma EC2 -Classic. Se você tiver uma dessas contas, poderá ter a opção de criar EC2Resource objetos para um pipeline em uma rede EC2 -Classic em vez de umaVPC. É altamente recomendável que você crie recursos para todos os seus pipelines emVPCs. Além disso, se você tiver recursos existentes no EC2 -Classic, recomendamos que você os migre para um. VPC

O objeto de exemplo a seguir inicia uma EC2 instância em EC2 -Classic, com alguns campos opcionais definidos.

```
{
  "id" : "MyEC2Resource",
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroups" : [
    "test-group",
    "default"
  ],
  "keyPair" : "my-key-pair"
}
```

EC2-VPC

O objeto de exemplo a seguir inicia uma EC2 instância em uma instância não padrãoVPC, com alguns campos opcionais definidos.

```
{
  "id" : "MyEC2Resource",
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroupIds" : [
    "sg-12345678",
  ]
}
```

```

    "sg-12345678"
  ],
  "subnetId": "subnet-12345678",
  "associatePublicIpAddress": "true",
  "keyPair" : "my-key-pair"
}

```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| resourceRole | A IAM função que controla os recursos que a EC2 instância da Amazon pode acessar. | String |
| role (perfil) | A IAM função AWS Data Pipeline usada para criar a EC2 instância. | String |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|--|
| schedule | <p>Esse objeto é invocado durante a execução de um intervalo de programação.</p> <p>Para definir a ordem de execução de dependência desse objeto, especifique uma referência de programação para outro objeto. Você pode fazer isso por meio de uma das seguintes maneiras:</p> <ul style="list-style-type: none"> Para garantir que todos os objetos no pipeline herdem a programação, defina uma programação no objeto explicitamente: <code>"schedule": {"ref": "DefaultSchedule"}</code> . Na maioria dos casos, é útil colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. | Objeto de referência. Por exemplo <code>"schedule": {"ref": "myScheduleId"}</code> |

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|--------------|
| | <ul style="list-style-type: none"> Se o pipeline tiver programações aninhadas na programação principal, você poderá criar um objeto pai que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html. | |
| Campos opcionais | Descrição | Tipo de slot |
| actionOnResourceFalha | A ação executada após uma falha de recurso para este recurso. Os valores válidos são "retryall" e "retrynone" . | String |
| actionOnTaskFalha | A ação executada após uma falha de tarefa para este recurso. Os valores válidos são "continue" ou "terminate" . | String |
| associatePublicIpEndereço | Indica se um endereço IP público deve ou não ser atribuído à instância. Se a instância estiver na Amazon EC2 ou na AmazonVPC, o valor padrão será true. Caso contrário, o valor padrão será false. | Booleano |
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | O tempo limite para a conclusão do trabalho remoto. Se definido, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--|
| availabilityZone | A zona de disponibilidade na qual iniciar a EC2 instância da Amazon. | String |
| disableIMDSv1 | O valor padrão é falso e ativa tanto IMDSv1 IMDSv2 e. Se você defini-lo como verdadeiro, ele desativará IMDSv1 e fornecerá apenas IMDSv2s | Booleano |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| httpProxy | O host proxy que os clientes usam para se conectar aos AWS serviços. | Objeto de referência. a. Por exemplo: "httpProxy": { "ref": "myHttpProxyId" } |
| imageId | O ID do AMI a ser usado para a instância. Por padrão, AWS Data Pipeline usa o tipo de HVM AMI virtualização. Os específicos AMI IDs usados são baseados em uma região. Você pode substituir o padrão AMI especificando o HVM AMI de sua escolha. Para obter mais informações sobre AMI tipos, consulte Tipos de AMI virtualização Linux e Como encontrar um Linux AMI no Guia do EC2 usuário da Amazon. | String |
| initTimeout | A quantidade de tempo de espera antes da inicialização do recurso. | Período |
| instanceCount | Suspenso. | Inteiro |
| instanceType | O tipo de EC2 instância da Amazon a ser iniciada. | String |

| Campos opcionais | Descrição | Tipo de slot |
|--------------------|---|---|
| keyPair | O nome do par de chaves. Se você iniciar uma EC2 instância da Amazon sem especificar um par de chaves, não poderá fazer login nela. | String |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | A quantidade máxima de novas tentativas após uma falha. | Inteiro |
| minInstanceCount | Suspenso. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência. Por exemplo <pre>"onFail": {"ref": "myActionId"}</pre> |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi programado ou ainda está em execução. | Objeto de referência. Por exemplo <pre>"onLateAction": {"ref": "myActionId"}</pre> |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência. Por exemplo: <pre>"onSuccess": {"ref": "myActionId"}</pre> |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|--|--|
| parent | O pai do objeto atual a partir do qual os slots são herdados. | Objeto de referência. a. Por exemplo: <code>"parent": {"ref": "myBaseObjectId"}</code> |
| pipelineLogUri | O Amazon S3 URI (como <code>'s3://BucketName/Key/'</code>) para fazer o upload de registros para o pipeline. | String |
| região | O código da região na qual a EC2 instância da Amazon deve ser executada. Por padrão, a instância é executada na mesma região que o pipeline. Você pode executar a instância na mesma região como um conjunto de dados dependente. | Enumeração |
| reportProgressTimeout | O tempo limite para as chamadas sucessivas de trabalho remoto para <code>reportProgress</code> . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e serão executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |
| runAsUser | O usuário que executará TaskRunner o. | String |
| runsOn | Esse campo não é permitido neste objeto. | Objeto de referência. a. Por exemplo: <code>"runsOn": {"ref": "myResourceId"}</code> |

| Campos opcionais | Descrição | Tipo de slot |
|-------------------------------|--|--------------|
| <code>scheduleType</code> | <p>O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início do intervalo, no final do intervalo ou sob demanda.</p> <p>Os valores são:</p> <ul style="list-style-type: none">• <code>timeseries</code> . As instâncias são programadas no final de cada intervalo.• <code>cron</code>. As instâncias são programadas no início de cada intervalo.• <code>ondemand</code>. Permite que você execute um pipeline uma vez por ativação. Você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação sob demanda, ela deverá ser especificada no objeto padrão, além de ser o único <code>scheduleType</code> especificado para objetos no pipeline. Para usar pipelines sob demanda, chame a operação <code>ActivatePipeline</code> para cada execução subsequente. | Enumeração |
| <code>securityGroupIds</code> | O IDs de um ou mais grupos de EC2 segurança da Amazon a serem usados para as instâncias no pool de recursos. | String |
| <code>securityGroups</code> | Um ou mais grupos EC2 de segurança da Amazon para usar nas instâncias no pool de recursos. | String |
| <code>spotBidPrice</code> | O valor máximo por hora para sua instância spot em dólares, que é um valor decimal entre 0 e 20,00, exclusivos. | String |

| Campos opcionais | Descrição | Tipo de slot |
|--------------------------|---|--------------|
| subnetId | O ID da EC2 sub-rede da Amazon na qual iniciar a instância. | String |
| terminateAfter | O número de horas após o qual encerrar o recurso. | Período |
| useOnDemandOnLastAttempt | Na última tentativa de solicitar uma instância spot, faça um pedido para instâncias sob demanda em vez de uma instância spot. Isso garante que, se todas as tentativas anteriores falharam, a última tentativa não será interrompida. | Booleano |
| workerGroup | Esse campo não é permitido neste objeto. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência a. Por exemplo: "activeInstances": { "ref": "myRunnable ObjectId" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | O cancellationReason se esse objeto foi cancelado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| @cascadeFailedOn | Descrição da cadeia de dependências na qual o objeto apresentou falha. | Objeto de referência. a. Por exemplo: "cascadeFailedOn": { "ref": "myRunnable ObjectId" } |
| emrStepLog | Os registros de etapas estão disponíveis somente nas tentativas de EMR atividade da Amazon. | String |
| errorId | O ID do erro se esse objeto apresentou falha. | String |
| errorMessage | A mensagem de erro se esse objeto apresentou falha. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @failureReason | O motivo da falha de recurso. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis sobre tentativas de atividades na AmazonEMR. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstance | ID do último objeto da instância concluído. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez em que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | O horário de término programado para o objeto. | DateTime |
| @scheduledStartTime | O horário de início programado para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência. Por exemplo: "waitingOn": { "ref": "myRunnableObjectId" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | O local de um objeto no ciclo de vida. Objetos de componentes dão origem a objetos de instância, que executam objetos de tentativa. | String |

EmrCluster

Representa a configuração de um EMR cluster da Amazon. Esse objeto é usado por [EmrActivity](#) e [HadoopActivity](#) para iniciar um cluster.

Conteúdo

- [Programadores](#)
- [Versões EMR de lançamento da Amazon](#)
- [EMRPermissões da Amazon](#)
- [Sintaxe](#)
- [Exemplos](#)
- [Consulte também](#)

Programadores

Os programadores fornecem uma maneira de especificar a alocação de recursos e a priorização de trabalhos dentro de um cluster Hadoop. Administradores ou usuários podem escolher um programador para várias classes de usuários e aplicativos. Um programador pode usar filas para alocar recursos para usuários e aplicativos. Você configura essas filas ao criar o cluster. Em seguida, você pode configurar a prioridade de certos tipos de trabalhos e usuários. Com isso, é possível usar recursos de cluster de maneira eficiente enquanto mais de um usuário envia trabalhos ao cluster. Existem três tipos de programadores disponíveis:

- [FairScheduler](#)— Tentativas de programar recursos uniformemente por um período significativo de tempo.

- [CapacityScheduler](#)— Usa filas para permitir que administradores de cluster atribuam usuários a filas de prioridade e alocação de recursos variáveis.
- Default – Usado pelo cluster e pode ser configurado pelo seu site.

Versões EMR de lançamento da Amazon

Uma EMR versão da Amazon é um conjunto de aplicativos de código aberto do ecossistema de big data. Cada versão inclui diferentes aplicativos, componentes e recursos de big data que você seleciona para que a Amazon EMR instale e configure ao criar um cluster. Especificar a versão usando o rótulo da versão. Os rótulos de versão estão no formato `emr-x.x.x`. Por exemplo, `emr-5.30.0`. EMROs clusters da Amazon com base no rótulo de lançamento `emr-4.0.0` e, posteriormente, usam a `releaseLabel` propriedade para especificar o rótulo de lançamento de um `EmrCluster` objeto. Versões anteriores usam a propriedade `amiVersion`.

Important

Todos os EMR clusters da Amazon criados usando a versão 5.22.0 ou posterior usam o [Signature versão 4](#) para autenticar solicitações para o Amazon S3. Algumas versões anteriores usam o Signature versão 2. O suporte ao Signature versão 2 está sendo descontinuado. Para obter mais informações, consulte [Atualização do Amazon S3 – Período de defasagem do SigV2 estendido e modificado](#). É altamente recomendável que você use uma versão de EMR lançamento da Amazon compatível com a Signature Version 4. Para versões anteriores, começando com EMR 4.7.x, a versão mais recente da série foi atualizada para oferecer suporte à Signature Version 4. Ao usar uma EMR versão anterior, recomendamos que você use a versão mais recente da série. Além disso, evite versões anteriores à EMR 4.7.0.

Condições e limitações

Use a versão mais recente do Task Runner

Se você estiver usando um objeto `EmrCluster` autogerenciado com um rótulo de release, use o Task Runner mais atual. Para mais informações sobre o Task Runner, consulte [Trabalhar com o Task Runner](#). Você pode configurar valores de propriedade para todas as classificações de EMR configuração da Amazon. Para obter mais informações, consulte [Como configurar aplicativos no Amazon EMR Release Guide](#) [the section called “EmrConfiguration”](#), no e nas referências de [the section called “Propriedade”](#) objetos.

Support for IMDSv2

Anteriormente, somente AWS Data Pipeline suportadoIMDSv1. Agora, é AWS Data Pipeline compatível com IMDSv2 Amazon EMR 5.23.1, 5.27.1 e 5.32 ou posterior e Amazon 6.2 ou posterior. EMR IMDSv2usa um método orientado à sessão para lidar melhor com a autenticação ao recuperar informações de metadados das instâncias. Você deve configurar suas instâncias para fazer IMDSv2 chamadas criando recursos gerenciados pelo usuário usando TaskRunner -2.0.

Amazon EMR 5.32 ou posterior e Amazon EMR 6.x

As séries Amazon EMR 5.32 ou posterior e 6.x usam a versão 3.x do Hadoop, que introduziu mudanças significativas na forma como o classpath do Hadoop é avaliado em comparação com a versão 2.x do Hadoop. Bibliotecas comuns como Joda-Time foram removidas do classpath.

Se [EmrActivity](#) ou [HadoopActivity](#) executa um arquivo Jar que tem dependências em uma biblioteca que foi removida no Hadoop 3.x, a etapa falhará com o erro `java.lang.NoClassDefFoundError` ou `java.lang.ClassNotFoundException`. Isso pode acontecer com arquivos Jar que foram executados sem problemas usando as versões de lançamento EMR 5.x da Amazon.

Para corrigir o problema, você deve copiar as dependências do arquivo Jar para o classpath do Hadoop em um objeto `EmrCluster` antes de iniciar o `EmrActivity` ou o `HadoopActivity`. Fornecemos um script bash para isso. O script bash está disponível no seguinte local, onde *MyRegion* é a AWS região em que seu `EmrCluster` objeto é executado, por exemplous-west-2.

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh
```

A forma de executar o script depende se ele `EmrActivity` `HadoopActivity` é executado em um recurso gerenciado AWS Data Pipeline ou executado em um recurso autogerenciado.

Se você usa um recurso gerenciado por AWS Data Pipeline, adicione um `bootstrapAction` ao `EmrCluster` objeto. O `bootstrapAction` especifica o script e os arquivos Jar a serem copiados como argumentos. Você pode adicionar até 255 campos `bootstrapAction` por objeto `EmrCluster` e adicionar um campo `bootstrapAction` a um objeto `EmrCluster` que já tenha ações de bootstrap.

Para especificar esse script como uma ação de bootstrap, use a seguinte sintaxe, onde `JarFileRegion` é a região em que o arquivo Jar é salvo e cada *MyJarFile* é o caminho absoluto no Amazon S3 de um arquivo Jar a ser copiado para o classpath do Hadoop. Não especifique arquivos Jar que estão no classpath do Hadoop por padrão.

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh,JarFileRegion,MyJarFile1,MyJarFile2[, ...]
```

O exemplo a seguir especifica uma ação de bootstrap que copia dois arquivos Jar no Amazon S3: `my-jar-file.jar` e o `emr-dynamodb-tool-4.14.0-jar-with-dependencies.jar`. A Região usada no exemplo é `us-west-2`.

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
  "keyPair" : "my-key-pair",
  "masterInstanceType" : "m5.xlarge",
  "coreInstanceType" : "m5.xlarge",
  "coreInstanceCount" : "2",
  "taskInstanceType" : "m5.xlarge",
  "taskInstanceCount" : "2",
  "bootstrapAction" : ["s3://datapipeline-us-west-2/us-west-2/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh,us-west-2,s3://path/to/my-jar-file.jar,s3://dynamodb-dpl-us-west-2/emr-ddb-storage-handler/4.14.0/emr-dynamodb-tools-4.14.0-jar-with-dependencies.jar"]
}
```

Você precisa salvar e ativar o pipeline para que a alteração no novo `bootstrapAction` seja habilitada.

Se você usar um recurso autogerenciado, poderá baixar o script para a instância do cluster e executá-lo na linha de comando usando SSH. O script cria um diretório chamado `/etc/hadoop/conf/shellprofile.d` e um arquivo chamado `datapipeline-jars.sh` nesse diretório. Os arquivos jar fornecidos como argumentos de linha de comando são copiados para um diretório que o script cria chamado `/home/hadoop/datapipeline_jars`. Se seu cluster estiver configurado de forma diferente, modifique o script adequadamente após baixá-lo.

A sintaxe para executar o script na linha de comando é um pouco diferente de usar a `bootstrapAction` exibida no exemplo anterior. Use espaços ao invés de vírgulas entre os argumentos, conforme mostrado no exemplo a seguir.

```
./copy-jars-to-hadoop-classpath.sh us-west-2 s3://path/to/my-jar-file.jar s3://dynamodb-dpl-us-west-2/emr-ddb-storage-handler/4.14.0/emr-dynamodb-tools-4.14.0-jar-with-dependencies.jar
```

EMRPermissões da Amazon

Ao criar um IAM papel personalizado, considere cuidadosamente as permissões mínimas necessárias para que seu cluster execute seu trabalho. Certifique-se de conceder acesso aos recursos necessários, como arquivos no Amazon S3 ou dados no AmazonRDS, Amazon Redshift ou DynamoDB. Se você quiser definir `visibleToAllUsers` como "False", sua função precisará das permissões adequadas. `DataPipelineDefaultRole` não tem essas permissões. Você precisa fornecer uma união das funções `DefaultDataPipelineResourceRole` e `DataPipelineDefaultRole` como a função de objeto `EmrCluster` ou criar sua própria função para essa finalidade.

Sintaxe

| Campos de invocação de objetos | Descrição | Tipo de slot |
|--------------------------------|--|--|
| schedule | Esse objeto é invocado durante a execução de um intervalo de programação. Especifique uma referência de programação para outro objeto para definir a ordem de execução de dependência desse objeto. É possível satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, ao especificar <code>"schedule": {"ref": "DefaultSchedule"}</code> . Na maioria dos casos, é melhor colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. Como alternativa, se o pipeline tiver uma árvore de programações (outras programações dentro de uma programação principal), você poderá criar um objeto principal que tenha uma referência de programação. Para obter mais informações sobre o exemplo de configurações opcionais de programação, consulte https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html . | Objeto de referência. a. Por exemplo: <pre>"schedule": {"ref": "myScheduleId"}</pre> |

| Campos opcionais | Descrição | Tipo de slot |
|----------------------------------|--|--------------|
| actionOnResourceFalha | A ação executada após uma falha de recurso para este recurso. Os valores válidos são "retryall", que tentará executar todas as tarefas para o cluster pela duração especificada e "retrynone". | String |
| actionOnTaskFalha | A ação executada após uma falha de tarefa para este recurso. Os valores válidos são "continuar", que significa que não encerrar o cluster, e "encerrar". | String |
| additionalMasterSecurityGroupIds | O identificador de grupos de segurança principais adicionais do EMR cluster, que segue o formato sg-01XXXX6a. Para obter mais informações, consulte Grupos de segurança EMR adicionais da Amazon no Guia EMR de gerenciamento da Amazon. | String |
| additionalSlaveSecurityGroupIds | O identificador de grupos de segurança escravos adicionais do EMR cluster, que segue o formuláriosg-01XXXX6a. | String |
| amiVersion | A versão Amazon Machine Image (AMI) que a Amazon EMR usa para instalar os nós do cluster. Para obter mais informações, consulte o Guia EMR de gerenciamento da Amazon . | String |
| applications | Aplicativos a serem instalados no cluster com argumentos separados por vírgula. Por padrão, o Hive e o Pig estão instalados. Esse parâmetro é aplicável somente para a Amazon EMR versão 4.0 e posterior. | String |
| attemptStatus | O status mais recente da atividade remota. | String |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------------|---|---|
| <code>attemptTimeout</code> | Tempo limite para conclusão do trabalho remoto. Se definida, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| <code>availabilityZone</code> | A zona de disponibilidade na qual o cluster será executado. | String |
| <code>bootstrapAction</code> | Uma ação para ser executada quando o cluster é iniciado. Você pode especificar argumentos separados por vírgula. Para especificar várias ações, até 255, adicione vários campos <code>bootstrapAction</code> . O comportamento padrão é iniciar o cluster sem quaisquer ações de bootstrap. | String |
| <code>configuration</code> | Configuração para o EMR cluster da Amazon. Esse parâmetro é aplicável somente para a Amazon EMR versão 4.0 e posterior. | Objeto de referência. Por exemplo: <pre>"configuration":{"ref":"myEmrConfigurationId"}</pre> |
| <code>coreInstanceBidPreço</code> | O preço spot máximo que você está disposto a pagar pelas EC2 instâncias da Amazon. Se um preço de oferta for especificado, a Amazon EMR usará instâncias spot para o grupo de instâncias. Especificado em USD. | String |
| <code>coreInstanceCount</code> | O número de nós core a serem usados no cluster. | Inteiro |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|--|--|
| coreInstanceType | O tipo de EC2 instância da Amazon a ser usada para os nós principais. Consulte Instâncias do Amazon EC2 compatíveis com clusters do Amazon EMR . | String |
| coreGroupConfiguration | A configuração do grupo de instâncias principais do EMR cluster da Amazon. Esse parâmetro é aplicável somente para a Amazon EMR versão 4.0 e posterior. | Objeto de referência. Por exemplo "configuration": {"ref": "myEmrConfigurationId"} |
| coreEbsConfiguration | A configuração dos EBS volumes da Amazon que serão anexados a cada um dos nós principais do grupo principal no EMR cluster da Amazon. Para obter mais informações, consulte Tipos de instância que suportam a EBS otimização no Guia EC2 do usuário da Amazon. | Objeto de referência. Por exemplo "coreEbsConfiguration": {"ref": "myEbsConfiguration"} |
| customAmild | Aplica-se somente à EMR versão 5.7.0 e posterior da Amazon. Especifica a AMI ID de uma personalização AMI a ser usada quando a Amazon EMR provisiona EC2 instâncias da Amazon. Ele também pode ser usado em vez de ações de bootstrap para personalizar as configurações dos nós do cluster. Para obter mais informações, consulte o tópico a seguir no Guia EMR de gerenciamento da Amazon. Usando um personalizado AMI | String |

| Campos opcionais | Descrição | Tipo de slot |
|--|--|--|
| <code>EbsBlockDeviceConfig</code> | <p>A configuração de um dispositivo de EBS bloco solicitado pela Amazon associado ao grupo de instâncias. Inclui um número especificado dos volumes que serão associados a cada instância no grupo de instâncias. Inclui <code>volumesPerInstance</code> e <code>volumeSpecification</code>, em que:</p> <ul style="list-style-type: none"> <code>volumesPerInstance</code> é o número de EBS volumes com uma configuração de volume específica que serão associados a cada instância no grupo de instâncias. <code>volumeSpecification</code> são as especificações de EBS volume da Amazon, como tipo e tamanho do volume em Gigabytes (GiB) IOPS, que serão solicitadas para o EBS volume anexado a uma instância EC2 no cluster da Amazon. EMR | Objeto de referência. Por exemplo <code>"EbsBlockDeviceConfig": {"ref": "myEbsBlockDeviceConfig"}</code> |
| <code>emrManagedMasterSecurityGroup</code> | O identificador do grupo de segurança principal do EMR cluster da Amazon, que segue a forma <code>sg-01XXXX6a</code> . Para obter mais informações, consulte Configurar grupos de segurança no Amazon EMR Management Guide. | String |
| <code>emrManagedSlaveSecurityGroup</code> | O identificador do grupo de segurança escravo do EMR cluster da Amazon, que segue o formulário <code>sg-01XXXX6a</code> . | String |
| <code>enableDebugging</code> | Permite a depuração no cluster da Amazon. EMR | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|--|---|
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| hadoopSchedulerType | O tipo de programador do cluster. Os tipos válidos são: PARALLEL_FAIR_SCHEDULING , PARALLEL_CAPACITY_SCHEDULING e DEFAULT_SCHEDULER . | Enumeração |
| httpProxy | O host proxy que os clientes usam para se conectar aos AWS serviços. | Objeto de referência, por exemplo, "httpProxy": {"ref": "myHttpProxy Id"} |
| initTimeout | A quantidade de tempo de espera antes da inicialização do recurso. | Período |
| keyPair | O par de EC2 chaves da Amazon a ser usado para fazer login no nó principal do EMR cluster da Amazon. | String |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| masterInstanceBidPreço | O preço spot máximo que você está disposto a pagar pelas EC2 instâncias da Amazon. É um valor decimal entre 0 e 20,00, exclusivos. Especificado emUSD. Definir esse valor ativa as Instâncias Spot para o nó principal EMR do cluster Amazon. Se um preço de oferta for especificado, a Amazon EMR usará instâncias spot para o grupo de instâncias. | String |

| Campos opcionais | Descrição | Tipo de slot |
|--------------------------|--|--|
| masterInstanceType | O tipo de EC2 instância da Amazon a ser usada para o nó principal. Consulte Instâncias do Amazon EC2 compatíveis com clusters do Amazon EMR . | String |
| masterGroupConfiguration | A configuração do grupo de instâncias mestre EMR do cluster da Amazon. Esse parâmetro é aplicável somente para a Amazon EMR versão 4.0 e posterior. | Objeto de referência. Por exemplo "configuration": {"ref": "myEmrConfigurationId"} |
| masterEbsConfiguration | A configuração dos EBS volumes da Amazon que serão anexados a cada um dos nós principais no grupo principal no EMR cluster da Amazon. Para obter mais informações, consulte Tipos de instância que suportam a EBS otimização no Guia EC2 do usuário da Amazon. | Objeto de referência. Por exemplo "masterEbsConfiguration": {"ref": "myEbsConfiguration"} |
| maxActiveInstances | O número máximo de instâncias ativas simultâneas de um componente. Novas execuções não contam para o número de instâncias ativas. | Inteiro |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência. Por exemplo: "onFail": {"ref": "myActionId"} |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------|---|--|
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência. Por exemplo: "onLateAction": { "ref": "myActionId" } |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, como "onSuccess": { "ref": "myActionId" } |
| parent | Pai do objeto atual a partir do qual os slots são herdados. | Objeto de referência. Por exemplo: "parent": { "ref": "myBaseObjectId" } |
| pipelineLogUri | O Amazon S3 URI (como 's3://BucketName/Key/ ') para carregar registros para o pipeline. | String |
| região | O código da região em que o EMR cluster da Amazon deve ser executado. Por padrão, o cluster é executado na mesma região que o pipeline. Você pode executar um cluster na mesma região como um conjunto de dados dependente. | Enumeração |
| releaseLabel | Rótulo de lançamento para o EMR cluster. | String |
| reportProgressTime out | Tempo limite para as chamadas sucessivas de trabalho remoto para reportProgress . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------------|---|--|
| resourceRole | A IAM função AWS Data Pipeline usada para criar o EMR cluster da Amazon. A função padrão é <code>DataPipelineDefaultRole</code> . | String |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |
| role (perfil) | A IAM função foi passada para a Amazon EMR para criar EC2 nós. | String |
| runsOn | Esse campo não é permitido neste objeto. | Objeto de referência. Por exemplo: <pre>"runsOn": {"ref": "myResourceId"}</pre> |
| securityConfiguration | O identificador da configuração de EMR segurança que será aplicada ao cluster. Esse parâmetro é aplicável somente para a Amazon EMR versão 4.8.0 e posterior. | String |
| serviceAccessSecurityGroupId | O identificador do grupo de segurança de acesso ao serviço do EMR cluster da Amazon. | String. Segue a forma <code>sg-01XXXX6a</code> . Por exemplo: <code>sg-1234abcd</code> . |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------------|--|--------------|
| <code>scheduleType</code> | O tipo de programação permite que você especifique se os objetos na sua definição de pipeline devem ser programados no início ou final do intervalo. Os valores são: <code>cron</code> , <code>ondemand</code> e <code>timeseries</code> . A programação <code>timeseries</code> significa que as instâncias são programadas no final de cada intervalo. A programação <code>cron</code> significa que as instâncias são programadas no início de cada intervalo. Uma programação <code>ondemand</code> permite que você execute um pipeline uma vez por ativação. Você não precisa clonar nem recriar o pipeline para executá-lo novamente. Se você usar uma programação <code>ondemand</code> , ela precisará ser especificada no objeto padrão, além de ser a única <code>scheduleType</code> especificada para objetos no pipeline. Para usar pipelines <code>ondemand</code> , chame a operação <code>ActivatePipeline</code> para cada execução subsequente. | Enumeração |
| <code>subnetId</code> | O identificador da sub-rede na qual iniciar o EMR cluster da Amazon. | String |
| <code>supportedProducts</code> | Um parâmetro que instala software de terceiros em um EMR cluster da Amazon, por exemplo, uma distribuição terceirizada do Hadoop. | String |
| <code>taskInstanceBidPreço</code> | O preço spot máximo que você está disposto a pagar pelas EC2 instâncias. Um valor decimal entre 0 e 20,00, exclusive. Especificado emUSD. Se um preço de oferta for especificado, a Amazon EMR usará instâncias spot para o grupo de instâncias. | String |

| Campos opcionais | Descrição | Tipo de slot |
|----------------------------|---|---|
| taskInstanceCount | O número de nós de tarefas a serem usados para o EMR cluster da Amazon. | Inteiro |
| taskInstanceType | O tipo de EC2 instância da Amazon a ser usada para nós de tarefas. | String |
| taskGroupConfigur ation | A configuração do grupo de instâncias de tarefas de EMR cluster da Amazon. Esse parâmetro é aplicável somente para a Amazon EMR versão 4.0 e posterior. | Objeto de referênci a. Por exemplo "configur ation": {"ref": "myEmrCon figurationId"} |
| taskEbsConfiguration | A configuração dos EBS volumes da Amazon que serão anexados a cada um dos nós de tarefas no grupo de tarefas no EMR cluster da Amazon. Para obter mais informações, consulte Tipos de instância que suportam a EBS otimização no Guia EC2 do usuário da Amazon. | Objeto de referênci a. Por exemplo "taskEbsC onfigur ation": {"ref": "myEbsCon figuration"} |
| terminateAfter | Encerrar o recurso após tantas horas. | Inteiro |

| Campos opcionais | Descrição | Tipo de slot |
|--------------------------|--|---|
| VolumeSpecification | <p>As especificações de EBS volume da Amazon, como tipo e tamanho do volume em Gigabytes (GiB) que serão solicitadas para o volume da Amazon EBS anexado a uma instância da Amazon EC2 no cluster da Amazon. IOPS EMR O nó pode ser um nó core, principal ou de tarefa.</p> <p>VolumeSpecification inclui:</p> <ul style="list-style-type: none"> • <code>iops()</code> Inteiro. O número de operações de E/S por segundo (IOPS) que o EBS volume da Amazon suporta, por exemplo, 1000. Para obter mais informações, consulte Características de EBS E/S no Guia do EC2 usuário da Amazon. • <code>sizeinGB()</code> . Inteiro. O tamanho do EBS volume da Amazon, em gibibytes (GiB), por exemplo, 500. Para obter informações sobre combinações válidas de tipos de volume e tamanhos de disco rígido, consulte Tipos de EBS volume no Guia EC2 do usuário da Amazon. • <code>volumeType</code> . String. O tipo de EBS volume da Amazon, por exemplo, gp2. Os tipos de volume suportados incluem gp2, io1, ST1, SC1 padrão e outros. Para obter mais informações, consulte Tipos de EBS volume no Guia EC2 do usuário da Amazon. | <p>Objeto de referência. Por exemplo</p> <pre> "VolumeSpecification": {"ref": "myVolumeSpecification"} </pre> |
| useOnDemandOnLastAttempt | <p>Na última tentativa de solicitar um recurso, faça um pedido para instâncias sob demanda em vez de instâncias spot. Isso garante que, se todas as tentativas anteriores falharam, a última tentativa não será interrompida.</p> | <p>Booleano</p> |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|---|---|
| workerGroup | Campo não é permitido neste objeto. | String |
| Campos de tempo de execução | Descrição | Tipo de slot |
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependências na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | Os registros de etapas estão disponíveis somente nas tentativas de EMR atividade da Amazon. | String |
| errorId | O ID do erro se esse objeto apresentou falha. | String |
| errorMessage | A mensagem de erro se esse objeto apresentou falha. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| @failureReason | O motivo da falha de recurso. | String |
| @finishedTime | A hora em que esse objeto terminou a execução. | DateTime |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis sobre tentativas de atividades na AmazonEMR. | String |
| @healthStatus | O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto. | String |
| @healthStatusFromInstanceid | ID do último objeto da instância concluído. | String |
| @healthStatusUpdated Hora | Hora em que o status de integridade foi atualizado pela última vez. | DateTime |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| @lastDeactivatedTime | A hora em que esse objeto foi desativado pela última vez. | DateTime |
| @latestCompletedRun Hora | Hora da última execução concluída. | DateTime |
| @latestRunTime | Hora da última execução programada. | DateTime |
| @nextRunTime | Hora da próxima execução a ser programada. | DateTime |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | O local de um objeto no ciclo de vida. Objetos de componentes dão origem a objetos de instância, que executam objetos de tentativa. | String |

Exemplos

Veja a seguir exemplos desse tipo de objeto.

Conteúdo

- [Inicie um EMR cluster da Amazon com hadoopVersion](#)

- [Inicie um EMR cluster da Amazon com a etiqueta de lançamento emr-4.x ou superior](#)
- [Instale software adicional em seu EMR cluster da Amazon](#)
- [Desativar a criptografia do lado do servidor em versões 3.x](#)
- [Desativar a criptografia do lado do servidor em versões 4.x](#)
- [Configure o Hadoop KMS ACLs e crie zonas de criptografia no HDFS](#)
- [Especificar IAM funções personalizadas](#)
- [Use o EmrCluster recurso em AWS SDK para Java](#)
- [Configurar um EMR cluster da Amazon em uma sub-rede privada](#)
- [Anexe EBS volumes aos nós do cluster](#)

Inicie um EMR cluster da Amazon com hadoopVersion

Example

O exemplo a seguir inicia um EMR cluster da Amazon usando a AMI versão 1.0 e o Hadoop 0.20.

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
  "hadoopVersion" : "0.20",
  "keyPair" : "my-key-pair",
  "masterInstanceType" : "m3.xlarge",
  "coreInstanceType" : "m3.xlarge",
  "coreInstanceCount" : "10",
  "taskInstanceType" : "m3.xlarge",
  "taskInstanceCount" : "10",
  "bootstrapAction" : ["s3://Region.elasticmapreduce/bootstrap-actions/configure-hadoop, arg1, arg2, arg3", "s3://Region.elasticmapreduce/bootstrap-actions/configure-hadoop/configure-other-stuff, arg1, arg2"]
}
```

Inicie um EMR cluster da Amazon com a etiqueta de lançamento emr-4.x ou superior

Example

O exemplo a seguir inicia um EMR cluster da Amazon usando o `releaseLabel` campo mais novo:

```
{
  "id" : "MyEmrCluster",
```

```

"type" : "EmrCluster",
"keyPair" : "my-key-pair",
"masterInstanceType" : "m3.xlarge",
"coreInstanceType" : "m3.xlarge",
"coreInstanceCount" : "10",
"taskInstanceType" : "m3.xlarge",
"taskInstanceCount": "10",
"releaseLabel": "emr-4.1.0",
"applications": ["spark", "hive", "pig"],
"configuration": {"ref":"myConfiguration"}
}

```

Instale software adicional em seu EMR cluster da Amazon

Example

`EmrCluster` fornece o `supportedProducts` campo que instala software de terceiros em um EMR cluster da Amazon, por exemplo, ele permite que você instale uma distribuição personalizada do Hadoop, como o MapR. Ele aceita uma lista de argumentos separados por vírgulas para os softwares de terceiros lerem e operarem. O exemplo a seguir mostra como usar o campo `supportedProducts` de `EmrCluster` para criar um cluster de edição MapR M3 personalizado com o Karmasphere Analytics instalado e executar um objeto `EmrActivity` nele.

```

{
  "id": "MyEmrActivity",
  "type": "EmrActivity",
  "schedule": {"ref": "ResourcePeriod"},
  "runsOn": {"ref": "MyEmrCluster"},
  "postStepCommand": "echo Ending job >> /mnt/var/log/stepCommand.txt",
  "preStepCommand": "echo Starting job > /mnt/var/log/stepCommand.txt",
  "step": "/home/hadoop/contrib/streaming/hadoop-streaming.jar, -input, s3n://
elasticmapreduce/samples/wordcount/input, -output, \
  hdfs:///output32113/, -mapper, s3n://elasticmapreduce/samples/wordcount/
wordSplitter.py, -reducer, aggregate"
},
{
  "id": "MyEmrCluster",
  "type": "EmrCluster",
  "schedule": {"ref": "ResourcePeriod"},
  "supportedProducts": ["mapr, --edition, m3, --version, 1.2, --key1, value1", "karmasphere-
enterprise-utility"],
  "masterInstanceType": "m3.xlarge",
  "taskInstanceType": "m3.xlarge"
}

```

```
}
```

Desativar a criptografia do lado do servidor em versões 3.x

Example

Uma `EmrCluster` atividade com uma versão 2.x do Hadoop criada por AWS Data Pipeline habilita a criptografia do lado do servidor por padrão. Se você quiser desativar a criptografia do lado do servidor, precisará especificar uma ação de bootstrap na definição de objeto do cluster.

O exemplo a seguir cria uma atividade `EmrCluster` com criptografia do lado do servidor desativada:

```
{
  "id": "NoSSEEmrCluster",
  "type": "EmrCluster",
  "hadoopVersion": "2.x",
  "keyPair": "my-key-pair",
  "masterInstanceType": "m3.xlarge",
  "coreInstanceType": "m3.large",
  "coreInstanceCount": "10",
  "taskInstanceType": "m3.large",
  "taskInstanceCount": "10",
  "bootstrapAction": ["s3://Region.elasticmapreduce/bootstrap-actions/configure-
hadoop,-e, fs.s3.enableServerSideEncryption=false"]
}
```

Desativar a criptografia do lado do servidor em versões 4.x

Example

Você precisa desativar a criptografia do lado do servidor usando um objeto `EmrConfiguration`.

O exemplo a seguir cria uma atividade `EmrCluster` com criptografia do lado do servidor desativada:

```
{
  "name": "ReleaseLabelCluster",
  "releaseLabel": "emr-4.1.0",
  "applications": ["spark", "hive", "pig"],
  "id": "myResourceId",
  "type": "EmrCluster",
  "configuration": {
    "ref": "disableSSE"
  }
}
```



```

},
{
  "name": "disableSSE",
  "id": "disableSSE",
  "type": "EmrConfiguration",
  "classification": "emrfs-site",
  "property": [{
    "ref": "enableServerSideEncryption"
  }]
},
{
  "name": "enableServerSideEncryption",
  "id": "enableServerSideEncryption",
  "type": "Property",
  "key": "fs.s3.enableServerSideEncryption",
  "value": "false"
}

```

Configure o Hadoop KMS ACLs e crie zonas de criptografia no HDFS

Example

Os objetos a seguir são criados ACLs para o Hadoop KMS e criam zonas de criptografia e chaves de criptografia correspondentes em: HDFS

```

{
  "name": "kmsAcls",
  "id": "kmsAcls",
  "type": "EmrConfiguration",
  "classification": "hadoop-kms-acls",
  "property": [
    {"ref": "kmsBlacklist"},
    {"ref": "kmsAcl"}
  ]
},
{
  "name": "hdfsEncryptionZone",
  "id": "hdfsEncryptionZone",
  "type": "EmrConfiguration",
  "classification": "hdfs-encryption-zones",
  "property": [
    {"ref": "hdfsPath1"},
    {"ref": "hdfsPath2"}
  ]
}

```

```
]
},
{
  "name": "kmsBlacklist",
  "id": "kmsBlacklist",
  "type": "Property",
  "key": "hadoop.kms.blacklist.CREATE",
  "value": "foo,myBannedUser"
},
{
  "name": "kmsAcl",
  "id": "kmsAcl",
  "type": "Property",
  "key": "hadoop.kms.acl.ROLLOVER",
  "value": "myAllowedUser"
},
{
  "name": "hdfsPath1",
  "id": "hdfsPath1",
  "type": "Property",
  "key": "/myHDFSPath1",
  "value": "path1_key"
},
{
  "name": "hdfsPath2",
  "id": "hdfsPath2",
  "type": "Property",
  "key": "/myHDFSPath2",
  "value": "path2_key"
}
}
```

Especificar IAM funções personalizadas

Example

Por padrão, AWS Data Pipeline passa `DataPipelineDefaultRole` como função de EMR serviço da Amazon e `DataPipelineDefaultResourceRole` como perfil de EC2 instância da Amazon para criar recursos em seu nome. No entanto, você pode criar uma função de EMR serviço personalizada da Amazon e um perfil de instância personalizado e usá-los em vez disso. AWS Data Pipeline deve ter permissões suficientes para criar clusters usando o papel personalizado e você deve adicionar AWS Data Pipeline como uma entidade confiável.

O objeto de exemplo a seguir especifica funções personalizadas para o EMR cluster da Amazon:

```
{
  "id": "MyEmrCluster",
  "type": "EmrCluster",
  "hadoopVersion": "2.x",
  "keyPair": "my-key-pair",
  "masterInstanceType": "m3.xlarge",
  "coreInstanceType": "m3.large",
  "coreInstanceCount": "10",
  "taskInstanceType": "m3.large",
  "taskInstanceCount": "10",
  "role": "emrServiceRole",
  "resourceRole": "emrInstanceProfile"
}
```

Use o `EmrCluster` recurso em AWS SDK para Java

Example

O exemplo a seguir mostra como usar `EmrCluster` e `EmrActivity` criar um cluster Amazon EMR 4.x para executar uma etapa do Spark usando o Java: SDK

```
public class dataPipelineEmr4 {

    public static void main(String[] args) {

        AWSCredentials credentials = null;
        credentials = new ProfileCredentialsProvider("/path/to/
        AwsCredentials.properties", "default").getCredentials();
        DataPipelineClient dp = new DataPipelineClient(credentials);
        CreatePipelineRequest createPipeline = new
        CreatePipelineRequest().withName("EMR4SDK").withUniqueId("unique");
        CreatePipelineResult createPipelineResult = dp.createPipeline(createPipeline);
        String pipelineId = createPipelineResult.getPipelineId();

        PipelineObject emrCluster = new PipelineObject()
            .withName("EmrClusterObj")
            .withId("EmrClusterObj")
            .withFields(
                new Field().withKey("releaseLabel").withStringValue("emr-4.1.0"),
                new Field().withKey("coreInstanceCount").withStringValue("3"),
                new Field().withKey("applications").withStringValue("spark"),
                new Field().withKey("applications").withStringValue("Presto-Sandbox"),
                new Field().withKey("type").withStringValue("EmrCluster"),
```

```
new Field().withKey("keyPair").withStringValue("myKeyName"),
new Field().withKey("masterInstanceType").withStringValue("m3.xlarge"),
new Field().withKey("coreInstanceType").withStringValue("m3.xlarge")
);

PipelineObject emrActivity = new PipelineObject()
    .withName("EmrActivityObj")
    .withId("EmrActivityObj")
    .withFields(
        new Field().withKey("step").withStringValue("command-runner.jar,spark-submit,--
executor-memory,1g,--class,org.apache.spark.examples.SparkPi,/usr/lib/spark/lib/spark-
examples.jar,10"),
        new Field().withKey("runsOn").withRefValue("EmrClusterObj"),
        new Field().withKey("type").withStringValue("EmrActivity")
    );

PipelineObject schedule = new PipelineObject()
    .withName("Every 15 Minutes")
    .withId("DefaultSchedule")
    .withFields(
        new Field().withKey("type").withStringValue("Schedule"),
        new Field().withKey("period").withStringValue("15 Minutes"),
        new Field().withKey("startAt").withStringValue("FIRST_ACTIVATION_DATE_TIME")
    );

PipelineObject defaultObject = new PipelineObject()
    .withName("Default")
    .withId("Default")
    .withFields(
        new Field().withKey("failureAndRerunMode").withStringValue("CASCADE"),
        new Field().withKey("schedule").withRefValue("DefaultSchedule"),
        new
Field().withKey("resourceRole").withStringValue("DataPipelineDefaultResourceRole"),
        new Field().withKey("role").withStringValue("DataPipelineDefaultRole"),
        new Field().withKey("pipelineLogUri").withStringValue("s3://myLogUri"),
        new Field().withKey("scheduleType").withStringValue("cron")
    );

List<PipelineObject> pipelineObjects = new ArrayList<PipelineObject>();

pipelineObjects.add(emrActivity);
pipelineObjects.add(emrCluster);
pipelineObjects.add(defaultObject);
pipelineObjects.add(schedule);
```

```

PutPipelineDefinitionRequest putPipelineDefintion = new PutPipelineDefinitionRequest()
    .withPipelineId(pipelineId)
    .withPipelineObjects(pipelineObjects);

PutPipelineDefinitionResult putPipelineResult =
dp.putPipelineDefinition(putPipelineDefintion);
System.out.println(putPipelineResult);

ActivatePipelineRequest activatePipelineReq = new ActivatePipelineRequest()
    .withPipelineId(pipelineId);
ActivatePipelineResult activatePipelineRes = dp.activatePipeline(activatePipelineReq);

    System.out.println(activatePipelineRes);
    System.out.println(pipelineId);

}

}

```

Configurar um EMR cluster da Amazon em uma sub-rede privada

Example

Este exemplo inclui uma configuração que inicia o cluster em uma sub-rede privada em um VPC. Para obter mais informações, consulte [Launch Amazon EMR Clusters VPC in a no](#) Amazon EMR Management Guide. Essa configuração é opcional. Você pode usá-la em qualquer pipeline que usa um objeto `EmrCluster`.

Para iniciar um EMR cluster da Amazon em uma sub-rede privada, especifique `SubnetId`, `emrManagedMasterSecurityGroupId`, `emrManagedSlaveSecurityGroupId`, e `serviceAccessSecurityGroupId` em sua `EmrCluster` configuração.

```

{
  "objects": [
    {
      "output": {
        "ref": "S3BackupLocation"
      },
      "input": {
        "ref": "DDBSourceTable"
      },
      "maximumRetries": "2",

```

```

    "name": "TableBackupActivity",
    "step": "s3://dynamodb-emr-#{myDDBRegion}/emr-ddb-storage-handler/2.1.0/emr-
ddb-2.1.0.jar,org.apache.hadoop.dynamodb.tools.DynamoDbExport,#{output.directoryPath},#{input.t
    "id": "TableBackupActivity",
    "runsOn": {
      "ref": "EmrClusterForBackup"
    },
    "type": "EmrActivity",
    "resizeClusterBeforeRunning": "false"
  },
  {
    "readThroughputPercent": "#{myDDBReadThroughputRatio}",
    "name": "DDBSourceTable",
    "id": "DDBSourceTable",
    "type": "DynamoDBDataNode",
    "tableName": "#{myDDBTableName}"
  },
  {
    "directoryPath": "#{myOutputS3Loc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-
mm-ss')}",
    "name": "S3BackupLocation",
    "id": "S3BackupLocation",
    "type": "S3DataNode"
  },
  {
    "name": "EmrClusterForBackup",
    "coreInstanceCount": "1",
    "taskInstanceCount": "1",
    "taskInstanceType": "m4.xlarge",
    "coreInstanceType": "m4.xlarge",
    "releaseLabel": "emr-4.7.0",
    "masterInstanceType": "m4.xlarge",
    "id": "EmrClusterForBackup",
    "subnetId": "#{mySubnetId}",
    "emrManagedMasterSecurityGroupId": "#{myMasterSecurityGroup}",
    "emrManagedSlaveSecurityGroupId": "#{mySlaveSecurityGroup}",
    "serviceAccessSecurityGroupId": "#{myServiceAccessSecurityGroup}",
    "region": "#{myDDBRegion}",
    "type": "EmrCluster",
    "keyPair": "user-key-pair"
  },
  {
    "failureAndRerunMode": "CASCADE",
    "resourceRole": "DataPipelineDefaultResourceRole",

```

```

    "role": "DataPipelineDefaultRole",
    "pipelineLogUri": "#{myPipelineLogUri}",
    "scheduleType": "ONDEMAND",
    "name": "Default",
    "id": "Default"
  }
],
"parameters": [
  {
    "description": "Output S3 folder",
    "id": "myOutputS3Loc",
    "type": "AWS::S3::ObjectKey"
  },
  {
    "description": "Source DynamoDB table name",
    "id": "myDDBTableName",
    "type": "String"
  },
  {
    "default": "0.25",
    "watermark": "Enter value between 0.1-1.0",
    "description": "DynamoDB read throughput ratio",
    "id": "myDDBReadThroughputRatio",
    "type": "Double"
  },
  {
    "default": "us-east-1",
    "watermark": "us-east-1",
    "description": "Region of the DynamoDB table",
    "id": "myDDBRegion",
    "type": "String"
  }
],
"values": {
  "myDDBRegion": "us-east-1",
  "myDDBTableName": "ddb_table",
  "myDDBReadThroughputRatio": "0.25",
  "myOutputS3Loc": "s3://s3_path",
  "mySubnetId": "subnet_id",
  "myServiceAccessSecurityGroup": "service access security group",
  "mySlaveSecurityGroup": "slave security group",
  "myMasterSecurityGroup": "master security group",
  "myPipelineLogUri": "s3://s3_path"
}

```

```
}
```

Anexe EBS volumes aos nós do cluster

Example

Você pode anexar EBS volumes a qualquer tipo de nó no EMR cluster dentro do seu pipeline. Para anexar EBS volumes aos nós `scoreEbsConfiguration`, use `masterEbsConfiguration`, e `TaskEbsConfiguration` em sua `EmrCluster` configuração.

Este exemplo do EMR cluster da Amazon usa EBS volumes da Amazon para seus nós mestre, de tarefas e principais. Para obter mais informações, consulte [EBS Volumes da Amazon na Amazon EMR](#) no Guia EMR de gerenciamento da Amazon.

Essas configurações são opcionais. Você pode usá-las em qualquer pipeline que usa um objeto `EmrCluster`.

No pipeline, clique na configuração do `EmrCluster` objeto, escolha `EBSConfiguração principal`, `EBSConfiguração principal` ou `EBSConfiguração da tarefa` e insira os detalhes da configuração semelhantes ao exemplo a seguir.

```
{
  "objects": [
    {
      "output": {
        "ref": "S3BackupLocation"
      },
      "input": {
        "ref": "DDBSourceTable"
      },
      "maximumRetries": "2",
      "name": "TableBackupActivity",
      "step": "s3://dynamodb-emr-#{myDDBRegion}/emr-ddb-storage-handler/2.1.0/emr-ddb-2.1.0.jar,org.apache.hadoop.dynamodb.tools.DynamoDbExport,#{output.directoryPath},#{input.t",
      "id": "TableBackupActivity",
      "runsOn": {
        "ref": "EmrClusterForBackup"
      },
      "type": "EmrActivity",
      "resizeClusterBeforeRunning": "false"
    },
    {
```



```

    "readThroughputPercent": "#{myDDBReadThroughputRatio}",
    "name": "DDBSourceTable",
    "id": "DDBSourceTable",
    "type": "DynamoDBDataNode",
    "tableName": "#{myDDBTableName}"
  },
  {
    "directoryPath": "#{myOutputS3Loc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-mm-ss')}",
    "name": "S3BackupLocation",
    "id": "S3BackupLocation",
    "type": "S3DataNode"
  },
  {
    "name": "EmrClusterForBackup",
    "coreInstanceCount": "1",
    "taskInstanceCount": "1",
    "taskInstanceType": "m4.xlarge",
    "coreInstanceType": "m4.xlarge",
    "releaseLabel": "emr-4.7.0",
    "masterInstanceType": "m4.xlarge",
    "id": "EmrClusterForBackup",
    "subnetId": "#{mySubnetId}",
    "emrManagedMasterSecurityGroupId": "#{myMasterSecurityGroup}",
    "emrManagedSlaveSecurityGroupId": "#{mySlaveSecurityGroup}",
    "region": "#{myDDBRegion}",
    "type": "EmrCluster",
    "coreEbsConfiguration": {
      "ref": "EBSConfiguration"
    },
    "masterEbsConfiguration": {
      "ref": "EBSConfiguration"
    },
    "taskEbsConfiguration": {
      "ref": "EBSConfiguration"
    },
    "keyPair": "user-key-pair"
  },
  {
    "name": "EBSConfiguration",
    "id": "EBSConfiguration",
    "ebsOptimized": "true",
    "ebsBlockDeviceConfig" : [
      { "ref": "EbsBlockDeviceConfig" }
    ]
  }

```

```
    ],
    "type": "EbsConfiguration"
  },
  {
    "name": "EbsBlockDeviceConfig",
    "id": "EbsBlockDeviceConfig",
    "type": "EbsBlockDeviceConfig",
    "volumesPerInstance" : "2",
    "volumeSpecification" : {
      "ref": "VolumeSpecification"
    }
  },
  {
    "name": "VolumeSpecification",
    "id": "VolumeSpecification",
    "type": "VolumeSpecification",
    "sizeInGB": "500",
    "volumeType": "io1",
    "iops": "1000"
  },
  {
    "failureAndRerunMode": "CASCADE",
    "resourceRole": "DataPipelineDefaultResourceRole",
    "role": "DataPipelineDefaultRole",
    "pipelineLogUri": "#{myPipelineLogUri}",
    "scheduleType": "ONDEMAND",
    "name": "Default",
    "id": "Default"
  }
],
"parameters": [
  {
    "description": "Output S3 folder",
    "id": "myOutputS3Loc",
    "type": "AWS::S3::ObjectKey"
  },
  {
    "description": "Source DynamoDB table name",
    "id": "myDDBTableName",
    "type": "String"
  },
  {
    "default": "0.25",
    "watermark": "Enter value between 0.1-1.0",
```

```
    "description": "DynamoDB read throughput ratio",
    "id": "myDDBReadThroughputRatio",
    "type": "Double"
  },
  {
    "default": "us-east-1",
    "watermark": "us-east-1",
    "description": "Region of the DynamoDB table",
    "id": "myDDBRegion",
    "type": "String"
  }
],
"values": {
  "myDDBRegion": "us-east-1",
  "myDDBTableName": "ddb_table",
  "myDDBReadThroughputRatio": "0.25",
  "myOutputS3Loc": "s3://s3_path",
  "mySubnetId": "subnet_id",
  "mySlaveSecurityGroup": "slave security group",
  "myMasterSecurityGroup": "master security group",
  "myPipelineLogUri": "s3://s3_path"
}
}
```

Consulte também

- [EmrActivity](#)

HttpProxy

HttpProxy permite que você configure seu próprio proxy e faça com que o Task Runner acesse o AWS Data Pipeline serviço por meio dele. Você não precisa configurar um Task Runner em execução com essas informações.

Exemplo de uma HttpProxy entrada TaskRunner

A seguinte definição do pipeline mostra um objeto HttpProxy:

```
{
  "objects": [
    {
      "schedule": {
```

```
    "ref": "Once"
  },
  "pipelineLogUri": "s3://myDPLogUri/path",
  "name": "Default",
  "id": "Default"
},
{
  "name": "test_proxy",
  "hostname": "hostname",
  "port": "port",
  "username": "username",
  "*password": "password",
  "windowsDomain": "windowsDomain",
  "type": "HttpProxy",
  "id": "test_proxy",
},
{
  "name": "ShellCommand",
  "id": "ShellCommand",
  "runsOn": {
    "ref": "Resource"
  },
  "type": "ShellCommandActivity",
  "command": "echo 'hello world' "
},
{
  "period": "1 day",
  "startDateTime": "2013-03-09T00:00:00",
  "name": "Once",
  "id": "Once",
  "endDateTime": "2013-03-10T00:00:00",
  "type": "Schedule"
},
{
  "role": "dataPipelineRole",
  "httpProxy": {
    "ref": "test_proxy"
  },
  "actionOnResourceFailure": "retrynone",
  "maximumRetries": "0",
  "type": "Ec2Resource",
  "terminateAfter": "10 minutes",
  "resourceRole": "resourceRole",
  "name": "Resource",
```

```

    "actionOnTaskFailure": "terminate",
    "securityGroups": "securityGroups",
    "keyPair": "keyPair",
    "id": "Resource",
    "region": "us-east-1"
  }
],
"parameters": []
}

```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| hostname | Host do proxy que os clientes usarão para se conectar aos AWS Serviços. | String |
| porta | Porta do host proxy que os clientes usarão para se conectar aos AWS Serviços. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |
| *password | Senha para o proxy. | String |
| s3 NoProxy | Desative o HTTP proxy ao se conectar ao Amazon S3 | Booleano |
| username | Nome do usuário para o proxy. | String |
| windowsDomain | O nome de domínio do Windows para NTLM Proxy. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|--------------|
| windowsWorkgroup | O nome do grupo de trabalho do Windows para NTLM Proxy. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Precondições

A seguir estão os objetos de AWS Data Pipeline pré-condição:

Objetos

- [DynamoDBData Existe](#)
- [DynamoDBTable Existe](#)
- [Existe](#)
- [S3 KeyExists](#)
- [S3 PrefixNotEmpty](#)

- [ShellCommandPrecondition](#)

DynamoDBData Existe

Uma condição para verificar se os dados existem em uma tabela do DynamoDB.

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| role (perfil) | Especifica a função a ser usada para executar a condição. | String |
| tableName | Tabela do DynamoDB para verificação. | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--------------|
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|---|--|
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| preconditionTimeout | O período inicial após o qual a condição é marcada como "com falha" se ainda não tiver sido atendida. | Período |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress. Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| currentRetryCount | O número de vezes que a condição foi testada nesta tentativa. | String |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| lastRetryTime | Última vez em que a condição foi testada nessa tentativa. | String |
| nó | O nó para o qual esta condição está sendo realizada. | Objeto de referência, por exemplo, "node": {"ref": "myRunnableObject Id"} |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id"} |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

D ynamoDBTable Existe

Uma pré-condição para verificar se a tabela do DynamoDB existe.

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| role (perfil) | Especifica a função a ser usada para executar a pré-condição. | String |
| tableName | Tabela do DynamoDB para verificação. | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--------------|
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|---|--|
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como <code>ondemand</code> . | Período |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| preconditionTimeout | O período inicial após o qual a condição é marcada como "com falha" se ainda não tiver sido atendida. | Período |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para <code>reportProgress</code> . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------------|---|--|
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |
| Campos de tempo de execução | Descrição | Tipo de slot |
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id"} |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id"} |
| currentRetryCount | O número de vezes que a pré-condição foi testada nesta tentativa. | String |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| lastRetryTime | Última vez em que a pré-condição foi testada nessa tentativa. | String |
| nó | O nó para o qual esta pré-condição está sendo realizada. | Objeto de referência, por exemplo, "node": {"ref": " myRunnabl eObject Id "} |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Existe

Verifica se existe um objeto de nó de dados.

Note

Recomendamos que você use as condições gerenciadas pelo sistema. Para obter mais informações, consulte [Precondições](#).

Exemplo

Veja a seguir um exemplo deste tipo de objeto. O objeto InputData faz referência a esse objeto, Ready, e a outro objeto que você definir no mesmo arquivo de definição de pipeline. CopyPeriod é um objeto Schedule.

```
{
  "id" : "InputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://example-bucket/InputData/#{@scheduledStartTime.format('YYYY-MM-dd-hh:mm')}.csv",
  "precondition" : { "ref" : "Ready" }
},
{
  "id" : "Ready",
  "type" : "Exists"
}
```

Sintaxe

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--------------|
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|---|--|
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| preconditionTimeout | O período inicial após o qual a condição é marcada como "com falha" se ainda não tiver sido atendida. | Período |
| reportProgressTimeout | Tempo limite para trabalho remoto: chamadas sucessivas para reportProgress. Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| nó | O nó para o qual esta condição está sendo realizada. | Objeto de referência, por exemplo, "node": {"ref": "myRunnableObject Id"} |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id"} |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| | aos objetos de instância que executam os objetos de tentativa. | |

Consulte também

- [ShellCommandPrecondition](#)

S3 KeyExists

Verifica se existe uma chave em um nó de dados do Amazon S3.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. A condição será acionada quando a chave, `s3://mybucket/mykey`, referenciada pelo parâmetro `s3Key`, existir.

```
{
  "id" : "InputReady",
  "type" : "S3KeyExists",
  "role" : "test-role",
  "s3Key" : "s3://mybucket/mykey"
}
```

Você também pode usar `S3KeyExists` como uma condição no segundo pipeline que aguarda a conclusão do primeiro pipeline. Para fazer isso:

1. Grave um arquivo no Amazon S3 ao final da conclusão do primeiro pipeline.
2. Crie uma condição `S3KeyExists` no segundo pipeline.

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| role (perfil) | Especifica a função a ser usada para executar a condição. | String |

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|-----------------------|--------------|
| s3Key | A chave do Amazon S3. | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|--|--|
| attemptStatus | Status mais recente da atividade remota. | String |
| attemptTimeout | Tempo limite antes de tentar concluir o trabalho remoto mais uma vez. Se configurada, uma atividade remota não concluída dentro do prazo definido após a inicialização poderá ser executada novamente. | Período |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maximumRetries | Número máximo de tentativas que são iniciadas em caso de falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |

| Campos opcionais | Descrição | Tipo de slot |
|-----------------------|--|--|
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| preconditionTimeout | O período inicial após o qual a condição é marcada como "com falha" se ainda não tiver sido atendida. | Período |
| reportProgressTimeout | Tempo limite para as chamadas sucessivas de trabalho remoto para <code>reportProgress</code> . Se configurada, as atividades remotas sem progresso para o período especificado poderão ser consideradas como interrompidas e serão executadas novamente. | Período |
| retryDelay | A duração do tempo limite entre duas tentativas sucessivas. | Período |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObjectId"} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id "} |
| currentRetryCount | O número de vezes que a pré-condição foi testada nesta tentativa. | String |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| lastRetryTime | Última vez em que a pré-condição foi testada nessa tentativa. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| nó | O nó para o qual esta condição está sendo realizada. | Objeto de referência, por exemplo, "node": {"ref": "myRunnableObject Id"} |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id"} |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| | aos objetos de instância que executam os objetos de tentativa. | |

Consulte também

- [ShellCommandPrecondition](#)

S3 PrefixNotEmpty

Uma condição prévia para verificar se os objetos do Amazon S3 com o prefixo fornecido (representado como URI a) estão presentes.

Exemplo

Veja a seguir um exemplo desse tipo de objeto usando campos obrigatórios, opcionais e de expressão.

```
{
  "id" : "InputReady",
  "type" : "S3PrefixNotEmpty",
  "role" : "test-role",
  "s3Prefix" : "#{node.filePath}"
}
```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|--|--------------|
| role (perfil) | Especifica a função a ser usada para executar a condição. | String |
| s3Prefix | O prefixo do Amazon S3 para verificar a existência de objetos. | String |

| Campos opcionais | Descrição | Tipo de slot |
|----------------------------------|---|--|
| <code>attemptStatus</code> | Status mais recente da atividade remota. | String |
| <code>attemptTimeout</code> | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| <code>failureAndRerunModo</code> | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| <code>lateAfterTimeout</code> | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como <code>ondemand</code> . | Período |
| <code>maximumRetries</code> | Quantidade máxima de novas tentativas com falha. | Inteiro |
| <code>onFail</code> | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| <code>onLateAction</code> | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| <code>onSuccess</code> | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| <code>parent</code> | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------------------|--|---|
| | | <code>{"ref": " myBaseObject Id "}</code> |
| <code>preconditionTimeout</code> | O período inicial após o qual a condição é marcada como "com falha" se ainda não tiver sido atendida. | Período |
| <code>reportProgressTimeout</code> | Tempo limite para trabalho remoto: chamadas sucessivas para <code>reportProgress</code> Se definidas , as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |

| Campos de tempo de execução | Descrição | Tipo de slot |
|---------------------------------|---|---|
| <code>@activeInstances</code> | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {" ref": " myRunnableObject Id "} |
| <code>@actualEndTime</code> | Hora em que a execução deste objeto foi concluída. | DateTime |
| <code>@actualStartTime</code> | Hora em que a execução deste objeto foi iniciada. | DateTime |
| <code>cancellationReason</code> | <code>cancellationReason</code> Se esse objeto foi cancelado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id"} |
| currentRetryCount | O número de vezes que a precondição foi testada nesta tentativa. | String |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |
| lastRetryTime | Última vez em que a precondição foi testada nessa tentativa. | String |
| nó | O nó para o qual esta precondição está sendo realizada. | Objeto de referência, por exemplo, "node": {"ref": "myRunnableObject Id"} |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id" } |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [ShellCommandPrecondition](#)

ShellCommandPrecondition

Um comando shell do Unix/Linux que pode ser executado como uma condição.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "VerifyDataReadiness",
  "type" : "ShellCommandPrecondition",
  "command" : "perl check-data-ready.pl"
}
```

Sintaxe

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|---|--|--------------|
| command | O comando a ser executado. Este valor e quaisquer parâmetros associados precisam funcionar no ambiente do qual você está executando o Task Runner. | String |
| scriptUri | Um URI caminho do Amazon S3 para um arquivo ser baixado e executado como um comando shell. Somente um campo scriptUri ou campo de comando deve estar presente. scriptUri não pode usar parâmetros, use o comando em vez disso. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| attemptStatus | Status mais recente da atividade remota. | String |

| Campos opcionais | Descrição | Tipo de slot |
|---------------------|---|--|
| attemptTimeout | Tempo limite para conclusão do trabalho remoto. Se configurada, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente. | Período |
| failureAndRerunModo | Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente. | Enumeração |
| lateAfterTimeout | O tempo decorrido após o início do pipeline no qual o objeto deve ser concluído. Ele é acionado somente quando o tipo de programação não está definido como ondemand. | Período |
| maximumRetries | Quantidade máxima de novas tentativas com falha. | Inteiro |
| onFail | Uma ação a ser executada quando há falha no objeto atual. | Objeto de referência, por exemplo, "onFail": {"ref": "myActionId"} |
| onLateAction | Ações que devem ser acionadas se um objeto ainda não foi agendado ou não foi concluído. | Objeto de referência, por exemplo, "onLateAction": {"ref": "myActionId"} |
| onSuccess | Uma ação a ser executada quando o objeto atual é executado com êxito. | Objeto de referência, por exemplo, "onSuccess": {"ref": "myActionId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |

| Campos opcionais | Descrição | Tipo de slot |
|------------------------------------|--|--------------|
| <code>preconditionTimeout</code> | O período inicial após o qual a condição é marcada como "com falha" se ainda não tiver sido atendida. | Período |
| <code>reportProgressTimeout</code> | Tempo limite para trabalho remoto: chamadas sucessivas para <code>reportProgress</code> . Se definidas, as atividades remotas sem progresso para o período especificado podem ser consideradas como interrompidas e executadas novamente. | Período |
| <code>retryDelay</code> | A duração do tempo limite entre duas novas tentativas. | Período |
| <code>scriptArgument</code> | Argumento a ser passado para o script de shell | String |
| <code>stderr</code> | O caminho do Amazon S3 que recebe mensagens de erro do sistema redirecionadas do comando. Se você usar o campo <code>runsOn</code> , ele precisará ser um caminho do Amazon S3 devido à natureza transitória do recurso que está executando sua atividade. No entanto, se você especificar o campo <code>workerGroup</code> , poderá usar um caminho de arquivo local. | String |
| <code>stdout</code> | O caminho do Amazon S3 que recebe saídas redirecionadas do comando. Se você usar o campo <code>runsOn</code> , ele precisará ser um caminho do Amazon S3 devido à natureza transitória do recurso que está executando sua atividade. No entanto, se você especificar o campo <code>workerGroup</code> , poderá usar um caminho de arquivo local. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|---|
| @activeInstances | Lista dos objetos da instância ativa agendados no momento. | Objeto de referência, por exemplo, "activeInstances": {"ref": "myRunnableObject Id" } |
| @actualEndTime | Hora em que a execução deste objeto foi concluída. | DateTime |
| @actualStartTime | Hora em que a execução deste objeto foi iniciada. | DateTime |
| cancellationReason | cancellationReason Se esse objeto foi cancelado. | String |
| @cascadeFailedOn | Descrição da cadeia de dependência na qual o objeto apresentou falha. | Objeto de referência, por exemplo, "cascadeFailedOn": {"ref": "myRunnableObject Id" } |
| emrStepLog | EMRregistros de etapas disponíveis somente em tentativas de EMR atividade | String |
| errorId | errorId Se esse objeto falhar. | String |
| errorMessage | errorMessage Se esse objeto falhar. | String |
| errorStackTrace | O rastreamento de pilha com erro se esse objeto apresentou falha. | String |
| hadoopJobLog | Registros de tarefas do Hadoop disponíveis em tentativas de atividades EMR baseadas. | String |
| hostname | O nome do host do cliente que capturou a tentativa da tarefa. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|--|--|
| nó | O nó para o qual esta condição está sendo realizada. | Objeto de referência, por exemplo, "node": {"ref": "myRunnableObject Id"} |
| reportProgressTime | A última vez que a atividade remota relatou progresso. | DateTime |
| @scheduledEndTime | Horário de término da programação para o objeto. | DateTime |
| @scheduledStartTime | Horário de início da programação para o objeto. | DateTime |
| @status | O status deste objeto. | String |
| @version | A versão do pipeline com que o objeto foi criado. | String |
| @waitingOn | Descrição da lista de dependências em que este objeto está aguardando. | Objeto de referência, por exemplo, "waitingOn": {"ref": "myRunnableObject Id"} |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| | aos objetos de instância que executam os objetos de tentativa. | |

Consulte também

- [ShellCommandActivity](#)
- [Existe](#)

Bancos de dados

A seguir estão os objetos do AWS Data Pipeline banco de dados:

Objetos

- [JdbcDatabase](#)
- [RdsDatabase](#)
- [RedshiftDatabase](#)

JdbcDatabase

Define um JDBC banco de dados.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyJdbcDatabase",
  "type" : "JdbcDatabase",
  "connectionString" : "jdbc:redshift://hostname:portnumber/dbname",
  "jdbcDriverClass" : "com.amazon.redshift.jdbc41.Driver",
  "jdbcDriverJarUri" : "s3://redshift-downloads/drivers/RedshiftJDBC41-1.1.6.1006.jar",
  "username" : "user_name",
  "password" : "my_password"
}
```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| connectionString | A cadeia de JDBC conexão para acessar o banco de dados. | String |
| jdbcDriverClass | A classe do driver a ser carregada antes de estabelecer a JDBC conexão. | String |
| *password | A senha a ser informada. | String |
| username | O nome de usuário a ser informado ao se conectar com o banco de dados. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| databaseName | Nome do banco de dados lógico para se conectar | String |
| jdbcDriverJarUri | A localização no Amazon S3 do JAR arquivo do JDBC driver usado para se conectar ao banco de dados. AWSO Data Pipeline deve ter permissão para ler esse JAR arquivo. | String |
| jdbcProperties | Pares do formato A=B que serão definidos como propriedades nas JDBC conexões desse banco de dados. | String |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |
| Campos do sistema | Descrição | Tipo de slot |
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

RdsDatabase

Define um RDS banco de dados da Amazon.

Note

RdsDatabase não é compatível com Aurora. Em vez disso, use [the section called "JdbcDatabase"](#) para Aurora.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyRdsDatabase",
  "type" : "RdsDatabase",
  "region" : "us-east-1",
  "username" : "user_name",
  "*password" : "my_password",
```

```
"rdsInstanceId" : "my_db_instance_identifier"
}
```

Para o mecanismo da Oracle, o campo `jdbcDriverJarUri` é necessário, e você pode especificar o seguinte driver: <http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html>. Para o mecanismo do SQL servidor, o `jdbcDriverJarUri` campo é obrigatório e você pode especificar o seguinte driver: <https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774>. Para os SQL mecanismos My SQL e Postgre, o `jdbcDriverJarUri` campo é opcional.

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|---|--------------|
| *password | A senha a ser informada. | String |
| rdsInstanceId | A propriedade <code>DBInstanceIdentifier</code> da instância de banco de dados. | String |
| username | O nome de usuário a ser informado ao se conectar com o banco de dados. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|--------------|
| databaseName | Nome do banco de dados lógico para se conectar | String |
| jdbcDriverJarUri | A localização no Amazon S3 do JAR arquivo do JDBC driver usado para se conectar ao banco de dados. AWSO Data Pipeline deve ter permissão para ler esse JAR arquivo. Para os SQL mecanismos My SQL e Postgre, o driver padrão é usado se esse campo não for especificado, mas você pode substituir o padrão usando esse campo. Para os | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|---|
| | mecanismos Oracle e SQL Server, esse campo é obrigatório. | |
| jdbcProperties | Pares do formato A=B que serão definidos como propriedades nas JDBC conexões desse banco de dados. | String |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |
| região | O código da região na qual o banco de dados está. Por exemplo, us-east-1. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

RedshiftDatabase

Define um banco de dados do Amazon Redshift. O `RedshiftDatabase` representa as propriedades do banco de dados usado pelo seu pipeline.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyRedshiftDatabase",
  "type" : "RedshiftDatabase",
  "clusterId" : "myRedshiftClusterId",
  "username" : "user_name",
  "*password" : "my_password",
  "databaseName" : "database_name"
}
```

Por padrão, o objeto usa o driver Postgres, que exige o campo `clusterId`. Para usar o driver do Amazon Redshift, especifique a string de conexão do banco de dados do Amazon Redshift no console do Amazon Redshift (inicia com "jdbc:redshift:") no campo `connectionString`.

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|--|--------------|
| *password | A senha a ser informada. | String |
| username | O nome de usuário a ser informado ao se conectar com o banco de dados. | String |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|--|--------------|
| clusterId | O identificador fornecido pelo usuário quando o cluster do Amazon Redshift foi criado. Por | String |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|--|--------------|
| | exemplo, se o endpoint para o cluster do Amazon Redshift for mydb.example.us-east-1.redshift.amazonaws.com, o identificador correto será mydb. No console do Amazon Redshift, você pode obter este valor no identificador ou no nome do cluster. | |
| connectionString | O JDBC endpoint para se conectar a uma instância do Amazon Redshift de propriedade de uma conta diferente do pipeline. Não é possível especificar ambos connectionString e clusterId. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--|
| databaseName | Nome do banco de dados lógico para se conectar. | String |
| jdbcProperties | Pares do formato A=B a serem definidos como propriedades nas JDBC conexões desse banco de dados. | String |
| parent | Pai do objeto atual a partir do qual os slots são herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| região | O código da região na qual o banco de dados está. Por exemplo, us-east-1. | Enumeração |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Formatos de dados

A seguir estão os objetos AWS Data Pipeline de formato de dados:

Objetos

- [CSVFormato de dados](#)
- [Formato de dados personalizado](#)
- [ynameDBDataFormato D](#)
- [D ynameDBExport DataFormat](#)
- [RegEx Formato de dados](#)
- [TSVFormato de dados](#)

CSVFormato de dados

Um formato de dados delimitado por vírgulas em que o separador de colunas é a vírgula e o separador de registros é o caractere de nova linha.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyOutputDataType",
  "type" : "CSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintaxe

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| column | Nome da coluna com o tipo dos dados especificado por campo para os dados descritos por esse nó de dados. Por exemplo: hostname STRING Para vários valores, use nomes de colunas e tipos de dados separados por um espaço. | String |
| escapeChar | Um caractere, por exemplo "\", que instrui o analisador para ignorar o próximo caractere. | String |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Formato de dados personalizado

Um formato de dados personalizado definido pela combinação de um determinado separador de colunas, separador de registros e caractere de escape.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyOutputDataType",
  "type" : "Custom",
  "columnSeparator" : ",",
  "recordSeparator" : "\n",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|-----------------------------|---|---|
| columnSeparator | Um caractere que indica o fim de uma coluna em um arquivo de dados. | String |
| Campos opcionais | Descrição | Tipo de slot |
| column | Nome da coluna com o tipo dos dados especificado por campo para os dados descritos por esse nó de dados. Por exemplo: hostname STRING Para vários valores, use nomes de colunas e tipos de dados separados por um espaço. | String |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |
| recordSeparator | Um caractere que indica o fim de uma linha em um arquivo de dados, por exemplo "\n". Há suporte apenas para caracteres únicos. | String |
| Campos de tempo de execução | Descrição | Tipo de slot |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

ynamoDBDataFormato D

Aplica um esquema a uma tabela do DynamoDB para que ela possa ser acessada por uma consulta do Hive. O `DynamoDBDataFormat` é usado como um objeto `HiveActivity` e uma entrada e saída `DynamoDBDataNode`. O `DynamoDBDataFormat` exige que você especifique todas as colunas na consulta do Hive. Para ter mais flexibilidade de especificar certas colunas em uma consulta do Hive ou receber suporte para o Amazon S3, consulte [DynamoDBExport DataFormat](#).

Note

Os booleanos do tipos DynamoDB não são mapeados para os tipos booleanos do Hive. No entanto, é possível mapear valores de 0 ou 1 inteiros do DynamoDB para os tipos booleanos do Hive.

Exemplo

O exemplo a seguir mostra como usar `DynamoDBDataFormat` para atribuir um esquema a uma entrada `DynamoDBDataNode`, permitindo que um objeto `HiveActivity` acesse os dados por colunas nomeadas e copie os dados para uma saída `DynamoDBDataNode`.

```
{
  "objects": [
    {
      "id" : "Exists.1",
      "name" : "Exists.1",
      "type" : "Exists"
    }
  ]
}
```

```
},
{
  "id" : "DataFormat.1",
  "name" : "DataFormat.1",
  "type" : "DynamoDBDataFormat",
  "column" : [
    "hash STRING",
    "range STRING"
  ]
},
{
  "id" : "DynamoDBDataNode.1",
  "name" : "DynamoDBDataNode.1",
  "type" : "DynamoDBDataNode",
  "tableName" : "$INPUT_TABLE_NAME",
  "schedule" : { "ref" : "ResourcePeriod" },
  "dataFormat" : { "ref" : "DataFormat.1" }
},
{
  "id" : "DynamoDBDataNode.2",
  "name" : "DynamoDBDataNode.2",
  "type" : "DynamoDBDataNode",
  "tableName" : "$OUTPUT_TABLE_NAME",
  "schedule" : { "ref" : "ResourcePeriod" },
  "dataFormat" : { "ref" : "DataFormat.1" }
},
{
  "id" : "EmrCluster.1",
  "name" : "EmrCluster.1",
  "type" : "EmrCluster",
  "schedule" : { "ref" : "ResourcePeriod" },
  "masterInstanceType" : "m1.small",
  "keyPair" : "$KEYPAIR"
},
{
  "id" : "HiveActivity.1",
  "name" : "HiveActivity.1",
  "type" : "HiveActivity",
  "input" : { "ref" : "DynamoDBDataNode.1" },
  "output" : { "ref" : "DynamoDBDataNode.2" },
  "schedule" : { "ref" : "ResourcePeriod" },
  "runsOn" : { "ref" : "EmrCluster.1" },
  "hiveScript" : "insert overwrite table ${output1} select * from ${input1} ;"
},
```

```

{
  "id" : "ResourcePeriod",
  "name" : "ResourcePeriod",
  "type" : "Schedule",
  "period" : "1 day",
  "startDateTime" : "2012-05-04T00:00:00",
  "endDateTime" : "2012-05-05T00:00:00"
}
]
}

```

Sintaxe

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--|
| column | O nome da coluna com o tipo dos dados especificado por campo para os dados descritos por esse nó de dados. Por exemplo, hostname STRING. Para vários valores, use nomes de colunas e tipos de dados separados por um espaço. | String |
| parent | O pai do objeto atual do qual os slots serão herdados. | Objeto de referência, como "parent": {"ref": "myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline usada para criar o objeto. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | O erro ao descrever o objeto malformatado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @pipelineId | O ID do pipeline ao qual esse objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

DynamoDBExport DataFormat

Aplica um esquema a uma tabela do DynamoDB para que ela possa ser acessada por uma consulta do Hive. Use `DynamoDBExportDataFormat` com um objeto `HiveCopyActivity` e a entrada e a saída `DynamoDBDataNode` ou `S3DataNode`. O `DynamoDBExportDataFormat` apresenta os seguintes benefícios:

- Fornece suporte tanto para o DynamoDB quanto para o Amazon S3
- Permite que você filtre dados por determinadas colunas na sua consulta do Hive
- Exporta todos os atributos do DynamoDB mesmo que você tenha um esquema esparsos

Note

Os booleanos dos tipos DynamoDB não são mapeados para os tipos booleanos do Hive. No entanto, é possível mapear valores de 0 ou 1 inteiros do DynamoDB para os tipos booleanos do Hive.

Exemplo

O exemplo a seguir mostra como usar `HiveCopyActivity` e `DynamoDBExportDataFormat` para copiar dados de um `DynamoDBDataNode` para outro ao aplicar filtros com base em um time stamp.

```
{
  "objects": [
    {
      "id" : "DataFormat.1",
      "name" : "DataFormat.1",

```

```

    "type" : "DynamoDBExportDataFormat",
    "column" : "timeStamp BIGINT"
  },
  {
    "id" : "DataFormat.2",
    "name" : "DataFormat.2",
    "type" : "DynamoDBExportDataFormat"
  },
  {
    "id" : "DynamoDBDataNode.1",
    "name" : "DynamoDBDataNode.1",
    "type" : "DynamoDBDataNode",
    "tableName" : "item_mapped_table_restore_temp",
    "schedule" : { "ref" : "ResourcePeriod" },
    "dataFormat" : { "ref" : "DataFormat.1" }
  },
  {
    "id" : "DynamoDBDataNode.2",
    "name" : "DynamoDBDataNode.2",
    "type" : "DynamoDBDataNode",
    "tableName" : "restore_table",
    "region" : "us_west_1",
    "schedule" : { "ref" : "ResourcePeriod" },
    "dataFormat" : { "ref" : "DataFormat.2" }
  },
  {
    "id" : "EmrCluster.1",
    "name" : "EmrCluster.1",
    "type" : "EmrCluster",
    "schedule" : { "ref" : "ResourcePeriod" },
    "masterInstanceType" : "m1.xlarge",
    "coreInstanceCount" : "4"
  },
  {
    "id" : "HiveTransform.1",
    "name" : "Hive Copy Transform.1",
    "type" : "HiveCopyActivity",
    "input" : { "ref" : "DynamoDBDataNode.1" },
    "output" : { "ref" : "DynamoDBDataNode.2" },
    "schedule" : { "ref" : "ResourcePeriod" },
    "runsOn" : { "ref" : "EmrCluster.1" },
    "filterSql" : "`timeStamp` > unix_timestamp(\"#{@scheduledStartTime}\", \"yyyy-MM-dd'T'HH:mm:ss\")"
  },

```

```

{
  "id" : "ResourcePeriod",
  "name" : "ResourcePeriod",
  "type" : "Schedule",
  "period" : "1 Hour",
  "startDateTime" : "2013-06-04T00:00:00",
  "endDateTime" : "2013-06-04T01:00:00"
}
]
}

```

Sintaxe

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|---|
| column | Nome da coluna com o tipo dos dados especificado por campo para os dados descritos por esse nó de dados. Ex: nome do host STRING | String |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

RegEx Formato de dados

Um formato de dados personalizado definido por uma expressão regular.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyInputDataType",
  "type" : "RegEx",
  "inputRegEx" : "([ ]*) ([ ]*) ([ ]*) (-|\\[[^\\]]*\\]) ([^ \\"]*|\"[^\"]*\"") (-|[0-9]*) (-|[0-9]*)?(?: ([^ \\"]*|\"[^\"]*\"") ([^ \\"]*|\"[^\"]*\""))?",
  "outputFormat" : "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s",
  "column" : [
    "host STRING",
    "identity STRING",
    "user STRING",
    "time STRING",
    "request STRING",
    "status STRING",
    "size STRING",
    "referer STRING",
    "agent STRING"
  ]
}
```

Sintaxe

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| column | Nome da coluna com o tipo dos dados especificado por campo para os dados descritos por esse nó de dados. Por exemplo: hostname STRING Para vários valores, use nomes de colunas e tipos de dados separados por um espaço. | String |
| inputRegex | A expressão regular para analisar um arquivo de entrada do S3. inputRegex fornece uma forma de recuperar colunas de dados relativamente não estruturados em um arquivo. | String |
| outputFormat | Os campos da coluna recuperados por inputRegex, mas referenciados como %1\$s %2\$s usando a sintaxe do formatador Java. | String |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

TSVFormato de dados

Um formato de dados delimitado por vírgulas em que o separador de colunas é o caractere de tabulação e o separador de registros é o caractere de nova linha.

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

```
{
  "id" : "MyOutputDataType",
  "type" : "TSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintaxe

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|--------------|
| column | Nome da coluna e o tipo dos dados descritos por esse nó de dados. Por exemplo "Name STRING" indica uma coluna chamada Name com campos para tipo de dados STRING. Separe vários pares de nome da coluna e | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|---|
| | tipo de dados com vírgulas (como exibido no exemplo). | |
| columnSeparator | O caractere que separa os campos em uma coluna de campos na próxima coluna. Assume '\t' como padrão. | String |
| escapeChar | Um caractere, por exemplo "\", que instrui o analisador para ignorar o próximo caractere. | String |
| parent | Pai do objeto atual a partir do qual os slots são herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |
| recordSeparator | O caractere que separa registros. Assume '\n' como padrão. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| | aos objetos de instância que executam os objetos de tentativa. | |

Ações

A seguir estão os objetos de AWS Data Pipeline ação:

Objetos

- [SnsAlarm](#)
- [Encerrar](#)

SnsAlarm

Envia uma mensagem de SNS notificação da Amazon quando uma atividade falha ou é concluída com sucesso.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Os valores de `node.input` e `node.output` são retirados do nó de dados ou da atividade que faz referência a este objeto no seu respectivo campo `onSuccess`.

```
{
  "id" : "SuccessNotify",
  "name" : "SuccessNotify",
  "type" : "SnsAlarm",
  "topicArn" : "arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic",
  "subject" : "COPY SUCCESS: #{node.@scheduledStartTime}",
  "message" : "Files were copied from #{node.input} to #{node.output}."
}
```


Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|--|--------------|
| message | O corpo do texto da SNS notificação da Amazon. | String |
| role (perfil) | A IAM função a ser usada para criar o SNS alarme da Amazon. | String |
| subject | A linha de assunto da mensagem de SNS notificação da Amazon. | String |
| topicArn | O SNS tópico da Amazon de destino ARN para a mensagem. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--|
| nó | O nó para o qual esta ação está sendo realizada. | Objeto de referência, por exemplo, "node": {"ref": " myRunnabl eObject Id "} |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Encerrar

Uma ação para acionar o cancelamento de uma atividade, recurso ou nó de dados pendente ou inacabado. AWS Data Pipeline tenta colocar a atividade, o recurso ou o nó de dados no CANCELLED estado se ele não começar pelo `lateAfterTimeout` valor.

Não é possível encerrar ações que incluem os recursos `onSuccess`, `OnFail` ou `onLateAction`.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. Neste exemplo, o campo `onLateAction` de `MyActivity` contém uma referência para a ação `DefaultAction1`. Ao fornecer uma ação para `onLateAction`, você também deve fornecer um valor `lateAfterTimeout` para indicar o período decorrido desde o início programado do pipeline, depois do qual a atividade será considerada como atrasada.

```
{
  "name" : "MyActivity",
  "id" : "DefaultActivity1",
  "schedule" : {
    "ref" : "MySchedule"
  },
  "runsOn" : {
    "ref" : "MyEmrCluster"
  },
  "lateAfterTimeout" : "1 Hours",
  "type" : "EmrActivity",
  "onLateAction" : {
    "ref" : "DefaultAction1"
  }
}
```

```

},
"step" : [
  "s3://myBucket/myPath/myStep.jar,firstArg,secondArg",
  "s3://myBucket/myPath/myOtherStep.jar,anotherArg"
]
},
{
  "name" : "TerminateTasks",
  "id" : "DefaultAction1",
  "type" : "Terminate"
}

```

Sintaxe

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| parent | Pai do objeto atual a partir do qual os slots são herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|---|
| nó | O nó para o qual esta ação está sendo realizada. | Objeto de referência, por exemplo "node": {"ref": " myRunnabl eObject Id "} |
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|--|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Schedule

Define o tempo de um evento programado, como quando uma atividade é executada.

Note

Quando o horário de início de um cronograma está no passado, AWS Data Pipeline preenche seu pipeline e começa a programar as execuções imediatamente, começando no horário de início especificado. Para testes/desenvolvimento, use um intervalo relativamente curto. Caso contrário, AWS Data Pipeline tente enfileirar e programar todas as execuções do seu pipeline para esse intervalo. AWS Data Pipeline tenta evitar preenchimentos acidentais se o componente da tubulação `scheduledStartTime` for anterior a 1 dia atrás, bloqueando a ativação da tubulação.

Exemplos

Veja a seguir um exemplo deste tipo de objeto. Ele define uma programação de hora em hora com início em 00:00:00 de 2012-09-01 e término em 00:00:00 de 2012-10-01. O primeiro período termina às 01:00:00 de 2012-09-01.

```
{
  "id" : "Hourly",
  "type" : "Schedule",
  "period" : "1 hours",
  "startDateTime" : "2012-09-01T00:00:00",
  "endDateTime" : "2012-10-01T00:00:00"
}
```

O pipeline a seguir é iniciado em `FIRST_ACTIVATION_DATE_TIME` e executado de hora em hora até 22:00:00 de 2014-04-25.

```
{
  "id": "SchedulePeriod",
  "name": "SchedulePeriod",
  "startAt": "FIRST_ACTIVATION_DATE_TIME",
  "period": "1 hours",
  "type": "Schedule",
  "endDateTime": "2014-04-25T22:00:00"
}
```

O pipeline a seguir tem início em `FIRST_ACTIVATION_DATE_TIME`, é executado de hora em hora e concluído após três ocorrências.

```
{
  "id": "SchedulePeriod",
  "name": "SchedulePeriod",
  "startAt": "FIRST_ACTIVATION_DATE_TIME",
  "period": "1 hours",
  "type": "Schedule",
  "occurrences": "3"
}
```

O pipeline a seguir tem início às 22:00:00 de 2014-04-25, é executado de hora em hora e concluído após três ocorrências.

```
{
  "id": "SchedulePeriod",
  "name": "SchedulePeriod",
  "startDateTime": "2014-04-25T22:00:00",
  "period": "1 hours",
  "type": "Schedule",
  "occurrences": "3"
}
```

Sob demanda usando o objeto Default

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
}
```

```
"role": "DataPipelineDefaultRole",
"scheduleType": "ondemand"
}
```

Sob demanda com objeto Schedule explícito

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
},
{
  "name": "DefaultSchedule",
  "type": "Schedule",
  "id": "DefaultSchedule",
  "period": "ONDEMAND_PERIOD",
  "startAt": "ONDEMAND_ACTIVATION_TIME"
},
```

Os exemplos a seguir demonstram como um objeto Schedule pode ser herdado do objeto Default, explicitamente definido para esse objeto ou fornecido por uma referência principal:

Schedule herdado do objeto Default

```
{
  "objects": [
    {
      "id": "Default",
      "failureAndRerunMode": "cascade",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "s3://myLogsbucket",
      "scheduleType": "cron",
      "schedule": {
        "ref": "DefaultSchedule"
      }
    },
    {
      "type": "Schedule",
      "id": "DefaultSchedule",
      "occurrences": "1",
      "period": "1 Day",

```

```
    "startAt": "FIRST_ACTIVATION_DATE_TIME"
  },
  {
    "id": "A_Fresh_NewEC2Instance",
    "type": "Ec2Resource",
    "terminateAfter": "1 Hour"
  },
  {
    "id": "ShellCommandActivity_HelloWorld",
    "runsOn": {
      "ref": "A_Fresh_NewEC2Instance"
    },
    "type": "ShellCommandActivity",
    "command": "echo 'Hello World!'"
  }
]
}
```

Schedule explícito no objeto

```
{
  "objects": [
    {
      "id": "Default",
      "failureAndRerunMode": "cascade",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "s3://myLogsbucket",
      "scheduleType": "cron"
    },
    {
      "type": "Schedule",
      "id": "DefaultSchedule",
      "occurrences": "1",
      "period": "1 Day",
      "startAt": "FIRST_ACTIVATION_DATE_TIME"
    },
    {
      "id": "A_Fresh_NewEC2Instance",
      "type": "Ec2Resource",
      "terminateAfter": "1 Hour"
    }
  ],
}
```

```

{
  "id": "ShellCommandActivity_HelloWorld",
  "runsOn": {
    "ref": "A_Fresh_NewEC2Instance"
  },
  "schedule": {
    "ref": "DefaultSchedule"
  },
  "type": "ShellCommandActivity",
  "command": "echo 'Hello World!'"
}
]
}

```

Schedule de uma referência principal

```

{
  "objects": [
    {
      "id": "Default",
      "failureAndRerunMode": "cascade",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "s3://myLogsbucket",
      "scheduleType": "cron"
    },
    {
      "id": "parent1",
      "schedule": {
        "ref": "DefaultSchedule"
      }
    },
    {
      "type": "Schedule",
      "id": "DefaultSchedule",
      "occurrences": "1",
      "period": "1 Day",
      "startAt": "FIRST_ACTIVATION_DATE_TIME"
    },
    {
      "id": "A_Fresh_NewEC2Instance",

```



```

    "type": "Ec2Resource",
    "terminateAfter": "1 Hour"
  },
  {
    "id": "ShellCommandActivity_HelloWorld",
    "runsOn": {
      "ref": "A_Fresh_NewEC2Instance"
    },
    "parent": {
      "ref": "parent1"
    },
    "type": "ShellCommandActivity",
    "command": "echo 'Hello World!'"
  }
]
}

```

Sintaxe

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|--|--------------|
| período | Com que frequência o pipeline deve ser executado. O formato é "N [minutes hours days weeks months]", em que N é um número seguido de um dos especificadores de tempo. Por exemplo, "15 minutes", executa o pipeline a cada 15 minutos. O período mínimo é de 15 minutos, e o máximo é de 3 anos. | Período |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|--|---|--------------|
| startAt | A data e a hora para iniciar as execuções programadas do pipeline. O valor válido é FIRST _ ACTIVATION _ DATE _ TIME, que | Enumeração |

| Grupo obrigatório (um dos seguintes é obrigatório) | Descrição | Tipo de slot |
|---|--|--------------|
| | foi descontinuado em favor da criação de um pipeline sob demanda. | |
| startTime | A data e a hora para iniciar as execuções programadas. Você deve usar um startTime ou startAt , mas não os dois. | DateTime |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|--|---|
| endTime | A data e a hora para terminar as execuções programadas. Deve ser uma data e hora posteriores ao valor de startTime ou startAt . O comportamento padrão é agendar as execuções até que o pipeline seja desligado. | DateTime |
| ocorrências | O número de vezes para executar o pipeline depois que ele é ativado. Você não pode usar ocorrências com endTime. | Inteiro |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|----------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @firstActivationTime | O tempo de criação do objeto. | DateTime |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Utilitários

Os seguintes objetos utilitários configuram outros objetos do pipeline:

Tópicos

- [ShellScriptConfig](#)
- [EmrConfiguration](#)
- [Propriedade](#)

ShellScriptConfig

Use com uma atividade para executar um script de shell para preActivityTask Config e Config postActivityTask. Esse objeto está disponível para [HadoopActivityHiveActivityHiveCopyActivity,, PigActivity](#). Você especifica um S3 URI e uma lista de argumentos para o script.

Exemplo

R ShellScriptConfig com argumentos:

```
{
  "id" : "ShellScriptConfig_1",
  "name" : "prescript",
  "type" : "ShellScriptConfig",
  "scriptUri": "s3://my-bucket/shell-cleanup.sh",
  "scriptArgument" : ["arg1","arg2"]
}
```

}

Sintaxe

Este objeto inclui os seguintes campos.

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| parent | Pai do objeto atual a partir do qual os slots são herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |
| scriptArgument | Uma lista de argumentos para uso com script de shell. | String |
| scriptUri | O script URI no Amazon S3 que deve ser baixado e executado. | String |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

EmrConfiguration

O EmrConfiguration objeto é a configuração usada para EMR clusters com versões 4.0.0 ou superiores. As configurações (como uma lista) são um parâmetro para a RunJobFlow API chamada. A configuração API da Amazon EMR usa uma classificação e propriedades. AWS Data Pipeline usa EmrConfiguration com objetos de propriedade correspondentes para configurar um [EmrCluster](#) aplicativo como Hadoop, Hive, Spark ou Pig em EMR clusters lançados em uma execução de pipeline. Como a configuração só pode ser alterada para novos clusters, você não pode fornecer um EmrConfiguration objeto para os recursos existentes. Para obter mais informações, consulte <https://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/>.

Exemplo

O seguinte objeto de configuração define as propriedades `io.file.buffer.size` e `fs.s3.block.size` em `core-site.xml`:

```
[
  {
    "classification": "core-site",
    "properties": {
      "io.file.buffer.size": "4096",
      "fs.s3.block.size": "67108864"
    }
  }
]
```

A definição do objeto de pipeline correspondente usa um EmrConfiguration objeto e uma lista de objetos de propriedade no `property` campo:

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    }
  ]
}
```

```
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      }],
      {
        "ref": "fs-s3-block-size"
      }
    ]
  },
  {
    "name": "io-file-buffer-size",
    "id": "io-file-buffer-size",
    "type": "Property",
    "key": "io.file.buffer.size",
    "value": "4096"
  },
  {
    "name": "fs-s3-block-size",
    "id": "fs-s3-block-size",
    "type": "Property",
    "key": "fs.s3.block.size",
    "value": "67108864"
  }
]
}
```

O exemplo a seguir é uma configuração aninhada usada para definir o ambiente Hadoop com a classificação `hadoop-env`:

```
[
  {
    "classification": "hadoop-env",
    "properties": {},
    "configurations": [
      {
        "classification": "export",
        "properties": {
          "YARN_PROXYSERVER_HEAPSIZE": "2396"
        }
      }
    ]
  }
]
```

```

    }
  }
]
}
]

```

Veja a seguir o objeto de definição de pipeline correspondente que usa essa configuração:

```

{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.0.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "hadoop-env"
      }
    },
    {
      "name": "hadoop-env",
      "id": "hadoop-env",
      "type": "EmrConfiguration",
      "classification": "hadoop-env",
      "configuration": {
        "ref": "export"
      }
    },
    {
      "name": "export",
      "id": "export",
      "type": "EmrConfiguration",
      "classification": "export",
      "property": {
        "ref": "yarn-proxyserver-heapsize"
      }
    },
    {
      "name": "yarn-proxyserver-heapsize",
      "id": "yarn-proxyserver-heapsize",
      "type": "Property",
      "key": "YARN_PROXYSERVER_HEAPSIZE",

```

```
    "value": "2396"
  },
]
}
```

O exemplo a seguir modifica uma propriedade específica do Hive para um cluster: EMR

```
{
  "objects": [
    {
      "name": "hivesite",
      "id": "hivesite",
      "type": "EmrConfiguration",
      "classification": "hive-site",
      "property": [
        {
          "ref": "hive-client-timeout"
        }
      ]
    },
    {
      "name": "hive-client-timeout",
      "id": "hive-client-timeout",
      "type": "Property",
      "key": "hive.metastore.client.socket.timeout",
      "value": "2400s"
    }
  ]
}
```

Sintaxe

Este objeto inclui os seguintes campos.

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|------------------------------------|--------------|
| classificação | Classificação para a configuração. | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|--|
| configuration | Subconfiguração para esta configuração. | Objeto de referência, por exemplo, "configuração": {"ref": "myEmrConfigurationId"} |
| parent | Pai do objeto atual a partir do qual os slots serão herdados. | Objeto de referência, por exemplo, "parent": {"ref": "myBaseObjectId"} |
| property | Propriedade de configuração. | Objeto de referência, por exemplo, "propriedade": {"ref": "myPropertyId"} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [EmrCluster](#)
- [Propriedade](#)
- [Guia de EMR lançamento da Amazon](#)

Propriedade

Uma única propriedade de valor-chave para uso com um objeto EmrConfiguration .

Exemplo

A definição de pipeline a seguir mostra um EmrConfiguration objeto e os objetos de propriedade correspondentes para iniciar um EmrCluster:

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      },
      {
        "ref": "fs-s3-block-size"
      }
    ]
  },
  {

```

```
    "name": "io-file-buffer-size",
    "id": "io-file-buffer-size",
    "type": "Property",
    "key": "io.file.buffer.size",
    "value": "4096"
  },
  {
    "name": "fs-s3-block-size",
    "id": "fs-s3-block-size",
    "type": "Property",
    "key": "fs.s3.block.size",
    "value": "67108864"
  }
]
}
```

Sintaxe

Este objeto inclui os seguintes campos.

| Campos obrigatórios | Descrição | Tipo de slot |
|---------------------|-----------|--------------|
| chave | chave | String |
| valor | valor | String |

| Campos opcionais | Descrição | Tipo de slot |
|------------------|---|---|
| parent | Pai do objeto atual a partir do qual os slots são herdados. | Objeto de referência, por exemplo, "parent": {"ref": " myBaseObject Id "} |

| Campos de tempo de execução | Descrição | Tipo de slot |
|-----------------------------|---|--------------|
| @version | A versão do pipeline com que o objeto foi criado. | String |

| Campos do sistema | Descrição | Tipo de slot |
|-------------------|---|--------------|
| @error | Erro ao descrever o objeto malformatado. | String |
| @pipelineId | ID do pipeline ao qual este objeto pertence. | String |
| @sphere | A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos objetos de instância que executam os objetos de tentativa. | String |

Consulte também

- [EmrCluster](#)
- [EmrConfiguration](#)
- [Guia de EMR lançamento da Amazon](#)

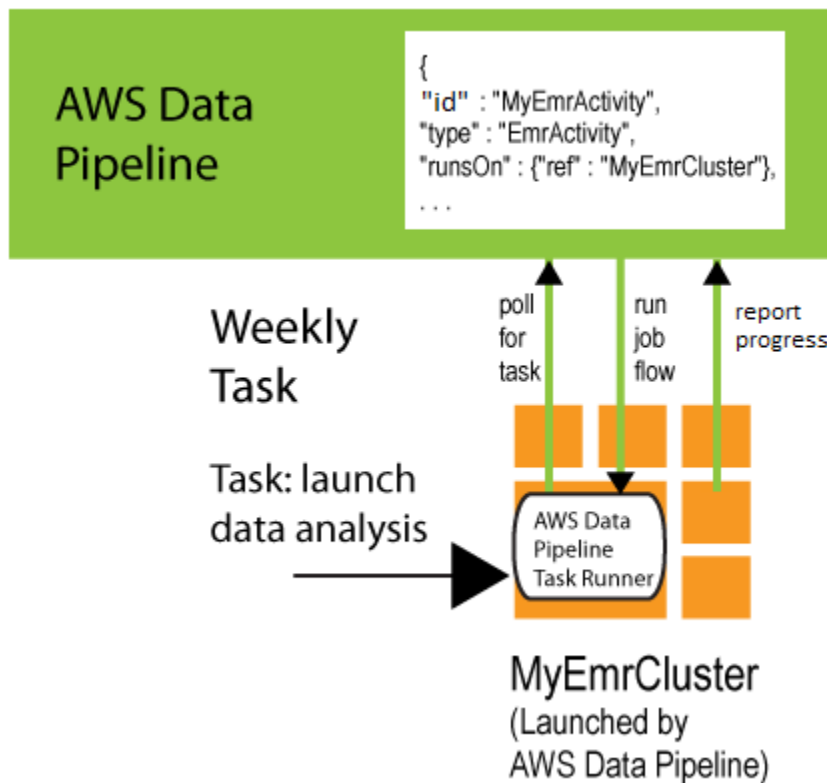
Trabalhar com o Task Runner

O Task Runner é um aplicativo de agente de tarefas que pesquisa AWS Data Pipeline tarefas agendadas e as executa em instâncias da AmazonEC2, EMR clusters da Amazon ou outros recursos computacionais, relatando o status à medida que o faz. Dependendo do seu aplicativo, você pode optar pelo seguinte:

- Permita AWS Data Pipeline instalar e gerenciar um ou mais aplicativos do Task Runner para você. Quando um pipeline é ativado, o padrão `Ec2Instance` ou `EmrCluster` objeto referenciado por um `runsOn` campo de atividade é criado automaticamente. AWS Data Pipeline cuida da instalação do Task Runner em uma EC2 instância ou no nó principal de um EMR cluster. Nesse padrão, AWS Data Pipeline pode fazer a maior parte do gerenciamento de instâncias ou clusters para você.
- Executar todo o pipeline ou partes dele nos recursos que você gerencia. Os recursos potenciais incluem uma EC2 instância da Amazon de longa duração, um EMR cluster da Amazon ou um servidor físico. Você pode instalar um executor de tarefas (que pode ser o Executor de Tarefas ou um agente de tarefas personalizado de sua autoria) praticamente em qualquer lugar, desde que ele possa se comunicar com o AWS Data Pipeline serviço web. Neste padrão, você tem controle quase completo sobre quais recursos são usados e como eles são gerenciados. Além disso, é necessário instalar e configurar o Task Runner manualmente. Para fazer isso, siga os procedimentos desta seção, conforme descrito em [Executar trabalho em recursos existentes usando o Task Runner](#).

Executor de tarefas em recursos AWS Data Pipeline gerenciados

Quando um recurso é iniciado e gerenciado pelo AWS Data Pipeline, o serviço web instala automaticamente o Task Runner nesse recurso para processar tarefas no pipeline. Você especifica um recurso computacional (uma EC2 instância da Amazon ou um EMR cluster da Amazon) para o `runsOn` campo de um objeto de atividade. Ao iniciar esse recurso, o AWS Data Pipeline instala o Task Runner nele e o configura para processar todos os objetos de atividade cujo campo de `runsOn` esteja definido para ele. Quando o recurso é AWS Data Pipeline encerrado, os logs do Task Runner são publicados em um local do Amazon S3 antes de serem encerrados.



Por exemplo, se você usar o `EmrActivity` em um pipeline e especificar um recurso `EmrCluster` no campo `runsOn`. Quando AWS Data Pipeline processa essa atividade, ele inicia um EMR cluster da Amazon e instala o Task Runner no nó principal. Em seguida, esse Task Runner processa as tarefas para atividades que têm o campo de `runsOn` definido para o objeto `EmrCluster`. O trecho a seguir de uma definição de pipeline mostra essa relação entre os dois objetos.

```
{
  "id" : "MyEmrActivity",
  "name" : "Work to perform on my data",
  "type" : "EmrActivity",
  "runsOn" : {"ref" : "MyEmrCluster"},
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : "s3://myBucket/myPath/myStep.jar,firstArg,secondArg",
  "step" : "s3://myBucket/myPath/myOtherStep.jar,anotherArg",
  "postStepCommand" : "scp localFiles remoteFiles",
  "input" : {"ref" : "MyS3Input"},
  "output" : {"ref" : "MyS3Output"}
},
{
  "id" : "MyEmrCluster",
  "name" : "EMR cluster to perform the work",
```

```
"type" : "EmrCluster",
"hadoopVersion" : "0.20",
"keypair" : "myKeyPair",
"masterInstanceType" : "m1.xlarge",
"coreInstanceType" : "m1.small",
"coreInstanceCount" : "10",
"taskInstanceType" : "m1.small",
"taskInstanceCount" : "10",
"bootstrapAction" : "s3://elasticmapreduce/libs/ba/configure-hadoop,arg1,arg2,arg3",
"bootstrapAction" : "s3://elasticmapreduce/libs/ba/configure-other-stuff,arg1,arg2"
}
```

Para obter mais informações e exemplos sobre como executar essa atividade, consulte [EmrActivity](#).

Se você tiver vários recursos AWS Data Pipeline gerenciados em um pipeline, o Task Runner será instalado em cada um deles e todos pesquisarão as tarefas a serem AWS Data Pipeline processadas.

Executar trabalho em recursos existentes usando o Task Runner

Você pode instalar o Task Runner nos recursos computacionais que você gerencia, como uma EC2 instância da Amazon, um servidor físico ou uma estação de trabalho. O Task Runner pode ser instalado em qualquer lugar, em qualquer hardware ou sistema operacional compatível, desde que possa se comunicar com o serviço AWS Data Pipeline web.

Essa abordagem pode ser útil quando, por exemplo, você deseja usar AWS Data Pipeline para processar dados armazenados no firewall da sua organização. Ao instalar o Task Runner em um servidor na rede local, você pode acessar o banco de dados local com segurança e, em seguida, pesquisar AWS Data Pipeline a próxima tarefa a ser executada. Quando AWS Data Pipeline termina o processamento ou exclui o pipeline, a instância do Task Runner permanece em execução no seu recurso computacional até que você a desligue manualmente. Os logs do Task Runner são mantidos depois que a execução do pipeline é concluída.

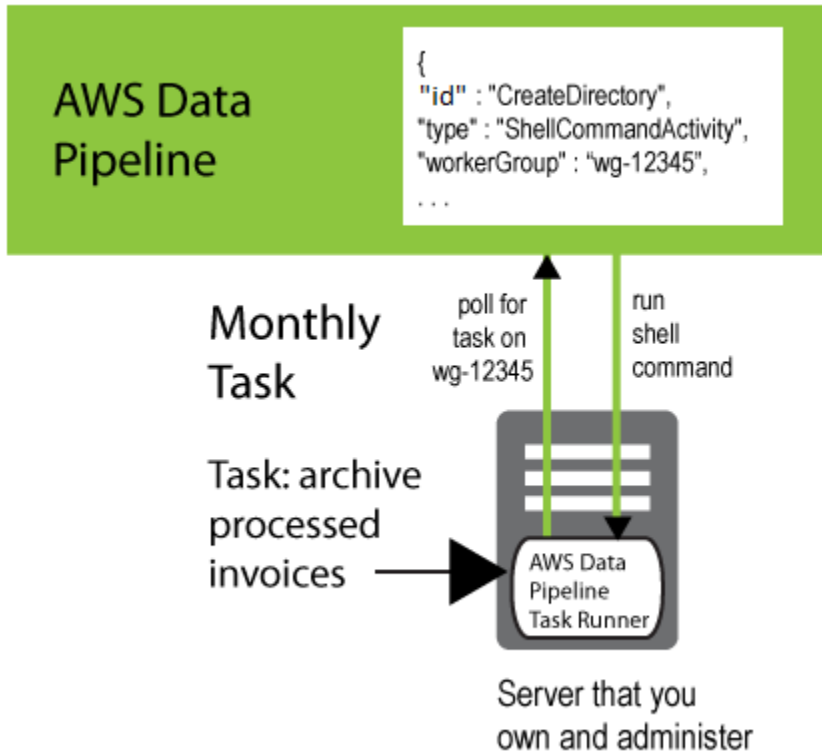
Para usar o Task Runner em um recurso que você gerencia, é necessário fazer download do Task Runner e instalá-lo no seu recurso computacional, seguindo os procedimentos nesta seção.

Note

Você só pode instalar o Task Runner no Linux ou no macOS. O Task Runner não é compatível com o sistema operacional Windows.

Para usar o Task Runner 2.0, a versão mínima necessária do Java é 1.7.

Para conectar um Task Runner que você instalou às atividades do pipeline que devem ser processadas, adicione um campo de `workerGroup` ao objeto e configure o Task Runner para pesquisar o valor do grupo desse operador. Você faz isso passando a string do grupo de trabalho como parâmetro (por exemplo, `--workerGroup=wg-12345`) ao executar o JAR arquivo Task Runner.



```
{
  "id" : "CreateDirectory",
  "type" : "ShellCommandActivity",
  "workerGroup" : "wg-12345",
  "command" : "mkdir new-directory"
}
```


Instalando o Task Runner

Esta seção explica como instalar e configurar o Task Runner e quais são os pré-requisitos. A instalação é um processo manual simples.

Para instalar o Task Runner

1. O Task Runner requer Java versões 1.6 ou 1.8. Para determinar se o Java está instalado e qual versão está sendo executada, use o seguinte comando:

```
java -version
```

Se você não tiver o Java 1.6 ou 1.8 instalado em seu computador, baixe uma dessas versões em <http://www.oracle.com/technetwork/java/index.html>. Faça download e instale o Java. Em seguida, vá para a próxima etapa.

2. Faça o download TaskRunner-1.0.jar de <https://s3.amazonaws.com/datapipeline-us-east-1/us-east-1/software/latest/TaskRunner/TaskRunner-1.0.jar> e copie-o em uma pasta no recurso de computação de destino. Para EMR clusters da Amazon executando EmrActivity tarefas, instale o Task Runner no nó principal do cluster.
3. Ao usar o Task Runner para se conectar ao serviço AWS Data Pipeline web para processar seus comandos, os usuários precisam de acesso programático a uma função que tenha permissões para criar ou gerenciar pipelines de dados. Para obter mais informações, consulte [Conceder acesso programático](#).
4. O Task Runner se conecta ao serviço AWS Data Pipeline web usando HTTPS. Se você estiver usando um AWS recurso, verifique se ele HTTPS está habilitado na tabela de roteamento e na ACL sub-rede apropriadas. Se você estiver usando um firewall ou proxy, verifique se a porta 443 está aberta.

(Opcional) Concedendo acesso ao Task Runner à Amazon RDS

A Amazon RDS permite que você controle o acesso às suas instâncias de banco de dados usando grupos de segurança de banco de dados (grupos de segurança de banco de dados). Um security group de banco de dados funciona como um firewall controlando o acesso da rede à sua Instância de banco de dados. Por padrão, o acesso à rede é desativado nas suas instâncias de banco de dados. Você deve modificar seus grupos de segurança de banco de dados para permitir que o Task Runner acesse suas RDS instâncias da Amazon. O Task Runner obtém RDS acesso à Amazon

a partir da instância em que é executado, portanto, as contas e os grupos de segurança que você adiciona à sua RDS instância da Amazon dependem de onde você instala o Task Runner.

Para conceder acesso ao Task Runner no EC2 -Classic

1. Abra o RDS console da Amazon.
2. No painel de navegação, selecione Instances e selecione sua instância de banco de dados.
3. Em Security and Network, selecione o security group. A página Security Groups é exibida com esse security group de banco de dados selecionado. Selecione o ícone de detalhes do security group de banco de dados.
4. Em Security Group Details, crie uma regra com Connection Type e Details apropriados. Esses campos dependem de onde o Task Runner está sendo executado, como descrito aqui:

- Ec2Resource

- Connection Type: EC2 Security Group

Detalhes: *my-security-group-name* (o nome do grupo de segurança que você criou para a EC2 instância)

- EmrResource

- Connection Type: EC2 Security Group

Detalhes: ElasticMapReduce-master

- Connection Type: EC2 Security Group

Detalhes: ElasticMapReduce-slave

- Seu ambiente local (nas instalações)

- Connection Type: CIDR/IP:

Detalhes: *my-ip-address* (o endereço IP do seu computador ou o intervalo de endereços IP da sua rede, se o computador estiver protegido por um firewall)

5. Clique em Add (Adicionar).

Para conceder acesso ao Task Runner em EC2 - VPC

1. Abra o RDS console da Amazon.
2. No painel de navegação, escolha Instances (Instâncias).

3. Selecione o ícone de detalhes da instância de banco de dados. Em Segurança e Rede, abra o link para o grupo de segurança, que leva você ao EC2 console da Amazon. Se você estiver usando o design antigo do console para security groups, mude para o novo design selecionando o ícone exibido na parte superior da página do console.
4. Na guia Entrada, selecione Editar, Adicionar regra. Especifique a porta do banco de dados que você usou quando iniciou a instância do banco de dados. A origem depende de onde o Task Runner está sendo executado, como descrito aqui:
 - `Ec2Resource`
 - `my-security-group-id` (o ID do grupo de segurança que você criou para a EC2 instância)
 - `EmrResource`
 - `master-security-group-id` (o ID do grupo ElasticMapReduce-master de segurança)
 - `slave-security-group-id` (o ID do grupo ElasticMapReduce-slave de segurança)
 - Seu ambiente local (nas instalações)
 - `ip-address` (o endereço IP do seu computador ou o intervalo de endereços IP da sua rede, se o computador estiver protegido por um firewall)
5. Clique em Salvar.

Iniciar o Task Runner

Em uma nova janela de prompt de comando configurada para o diretório em que você instalou o Task Runner, inicie o Task Runner com o comando a seguir.

```
java -jar TaskRunner-1.0.jar --config ~/credentials.json --workerGroup=myWorkerGroup --region=MyRegion --logUri=s3://mybucket/foldername
```

A opção `--config` aponta para o arquivo de credenciais.

A opção `--workerGroup` especifica o nome do grupo do operador, que deve ser o mesmo valor especificado no seu pipeline para que tarefas sejam processadas.

A opção `--region` especifica a região de serviço de onde as tarefas serão retiradas para execução.

A opção `--logUri` é usada para enviar seus logs compactados para um local no Amazon S3.

Quando o Task Runner está ativo, ele imprime o caminho do local onde os arquivos de log serão gravados na janela do terminal. Veja um exemplo a seguir.

```
Logging to /Computer_Name/.../output/logs
```

O Task Runner deve ser executado separadamente do seu shell de login. Se você estiver usando um aplicativo de terminal para se conectar ao seu computador, precisará de um utilitário, como o `nohup`, ou uma tela para impedir que o aplicativo Task Runner seja desligado quando você se desconectar. Para obter mais informações sobre as opções de linha de comando, consulte [Opções de configuração do Task Runner](#).

Verificando o registro do Task Runner

A maneira mais fácil de saber se o Task Runner está funcionando é verificar se ele está gravando arquivos de log. De hora em hora, o Task Runner grava arquivos de log no diretório, `output/logs`, sob o diretório em que ele está instalado. O nome do arquivo é `Task Runner.log.YYYY-MM-DD-HH`, onde `HH` é executado de 00 a 23, em UDT. Para economizar espaço de armazenamento, todos os arquivos de log com mais de oito horas são compactados com GZip.

Threads e pré-condições do Task Runner

O Task Runner usa um grupo de threads para cada uma das tarefas, atividades e pré-condições. A configuração padrão para `--tasks` é 2, o que significa que há dois threads alocados do pool de tarefas e cada thread pesquisa o AWS Data Pipeline serviço em busca de novas tarefas. Desse modo, `--tasks` é um atributo de ajuste de desempenho que pode ser usado para ajudar a otimizar o throughput do pipeline.

A lógica de nova tentativa do pipeline para pré-condições ocorre no Task Runner. Dois segmentos de pré-condição são alocados para pesquisar objetos de pré-condição. AWS Data Pipeline O Task Runner respeita o objeto `retryDelay` os `preconditionTimeout` campos de pré-condição que você define em condições prévias.

Em muitos casos, diminuir o tempo limite da pesquisa de pré-condição e o número de novas tentativas ajuda a melhorar o desempenho do seu aplicativo. Da mesma forma, os aplicativos com pré-condições prolongadas podem precisar ter o tempo limite e os valores de novas tentativas aumentados. Para obter mais informações objetos de pré-condição, consulte [Precondições](#).

Opções de configuração do Task Runner

Estas são as opções de configuração disponíveis na linha de comando quando você inicia o Task Runner.

| Parâmetro da linha de comando | Descrição |
|-------------------------------|---|
| <code>--help</code> | Ajuda da linha de comando. Exemplo: <code>Java -jar TaskRunner-1.0.jar --help</code> |
| <code>--config</code> | O caminho e o nome do seu arquivo <code>credentials.json</code> . |
| <code>--accessId</code> | <p>Seu ID de chave de AWS acesso para o Task Runner usar ao fazer solicitações.</p> <p>As opções <code>--accessID</code> e <code>--secretKey</code> oferecem uma alternativa para usar um arquivo <code>credentials.json</code>. Se um arquivo <code>credentials.json</code> também for fornecido, as opções <code>--accessID</code> e <code>--secretKey</code> terão prioridade.</p> |
| <code>--secretKey</code> | Sua chave AWS secreta para o Task Runner usar ao fazer solicitações. Para obter mais informações, consulte <code>--accessID</code> . |
| <code>--endpoint</code> | Um endpoint é URL aquele que é o ponto de entrada para um serviço web. O endpoint do AWS Data Pipeline serviço na região em que você está fazendo solicitações. Opcional. De modo geral, especificar uma região é suficiente e não há necessidade de definir o endpoint. Para obter uma lista de AWS Data Pipeline regiões e endpoints, consulte Regiões e endpoints do AWS Data Pipeline no. Referência geral da AWS |

| Parâmetro da linha de comando | Descrição |
|-------------------------------|---|
| <code>--workerGroup</code> | <p>O nome do grupo de operadores para o qual o Task Runner recupera o trabalho. Obrigatório.</p> <p>Quando o Task Runner pesquisa o serviço web, ele usa as credenciais que você forneceu e o valor de <code>workerGroup</code> para selecionar as tarefas a serem recuperadas (se houver alguma). Você pode usar qualquer nome significativo para você. No entanto, a string precisa corresponder com o Task Runner e suas respectivas atividades de pipeline. O nome do grupo de operadores está vinculado a uma região. Mesmo que haja nomes idênticos de grupos de operadores em outras regiões, o Task Runner sempre receberá tarefas da região especificada em <code>--region</code>.</p> |
| <code>--taskrunnerId</code> | <p>O ID do executor de tarefas a ser usado para informar o andamento. Opcional.</p> |
| <code>--output</code> | <p>O diretório do Task Runner para os arquivos de saída de log. Opcional. Os arquivos de log ficam armazenados em um diretório local até serem enviados ao Amazon S3. Essa opção substituirá o diretório padrão.</p> |
| <code>--region</code> | <p>A região da a ser usada. Embora seja opcional, recomendamos que você sempre configure a região. Se você não especificar a região, o Task Runner recuperará as tarefas da região de serviços padrão, <code>us-east-1</code>.</p> <p>Outras regiões com suporte são: <code>eu-west-1</code>, <code>ap-northeast-1</code>, <code>ap-southeast-2</code>, <code>us-west-2</code>.</p> |

| Parâmetro da linha de comando | Descrição |
|---------------------------------|---|
| <code>--logUri</code> | O caminho de destino do Amazon S3 para que o Task Runner faça o backup dos arquivos de log de hora em hora. Quando o Task Runner encerra, logs ativos do diretório local são enviados para a pasta de destino do Amazon S3. |
| <code>--proxyHost</code> | O host do proxy usado pelos clientes do Task Runner para se conectar aos AWS serviços. |
| <code>--proxyPort</code> | Porta do host proxy usada pelos clientes do Task Runner para se conectar aos AWS serviços. |
| <code>--proxyUsername</code> | O nome do usuário para o proxy. |
| <code>--proxyPassword</code> | A senha para o proxy. |
| <code>--proxyDomain</code> | O nome de domínio do Windows para NTLM Proxy. |
| <code>--proxyWorkstation</code> | O nome da estação de trabalho Windows para NTLM Proxy. |

Usar o Task Runner com um proxy

Se você estiver usando um host proxy, poderá especificar sua [configuração](#) ao invocar o Task Runner ou definir a variável de ambiente `_HTTPS_PROXY`. A variável de ambiente usada com o Task Runner aceita a mesma configuração usada para a [interface de linha de AWS comando](#).

Task Runner e Custom AMIs

Quando você especifica um `Ec2Resource` objeto para seu pipeline, AWS Data Pipeline cria uma EC2 instância para você, usando uma AMI que instala e configura o Task Runner para você. É necessário um tipo de instância compatível com PV. Como alternativa, você pode criar um

personalizado AMI com o Task Runner e, em seguida, especificar o ID AMI dele usando o `imageId` campo do `Ec2Resource` objeto. Para obter mais informações, consulte [Ec2Resource](#).

Um personalizado AMI deve atender aos seguintes requisitos AWS Data Pipeline para usá-lo com êxito no Task Runner:

- Crie o AMI na mesma região em que as instâncias serão executadas. Para obter mais informações, consulte [Creating Your Own AMI](#) no Guia EC2 do usuário da Amazon.
- Certifique-se de que o tipo de virtualização do AMI seja compatível com o tipo de instância que você planeja usar. Por exemplo, os tipos de instância I2 e G2 exigem um HVM AMI e os tipos de instância T1, C1, M1 e M2 exigem um PV. AMI Para obter mais informações, consulte [Tipos de AMI virtualização Linux](#) no Guia do EC2 usuário da Amazon.
- Instale o seguinte:
 - Linux
 - Bash
 - wget
 - unzip
 - Java 1.6 ou 1.8
 - cloud-init
- Crie e configure um usuário chamado `ec2-user`.

Solução de problemas

Quando você tem um problema com o AWS Data Pipeline, o sintoma mais comum é um pipeline não ser executado. Você pode usar os dados que o console e a CLI fornecem para identificar o problema e encontrar uma solução.

Índice

- [Localizar erros em pipelines](#)
- [Identificar o cluster do Amazon EMR que serve seu pipeline](#)
- [Interpretar detalhes de status do pipeline](#)
- [Localizar logs de erro](#)
- [Resolver problemas comuns](#)

Localizar erros em pipelines

O console do AWS Data Pipeline é uma ferramenta conveniente para monitorar visualmente o status dos pipelines e localiza com facilidade os erros relacionados a execuções de pipeline incompletas ou com falhas.

Para localizar erros de execuções incompletas ou com falhas com o console

1. Na página List Pipelines, se a coluna Status de qualquer uma de suas instâncias de pipeline exibe um status diferente de FINISHED, o pipeline está esperando que alguma pré-condição seja atendida ou ele apresentou falha e você precisa para solucionar o problema do pipeline.
2. Na página Listar pipelines, localize o pipeline de instância e selecione o triângulo à esquerda, para expandir os detalhes.
3. Na parte inferior desse painel, escolha View execution details. O painel Instance summary é exibido para mostrar os detalhes da instância selecionada.
4. No painel Resumo da instância, selecione o triângulo ao lado da instância para ver seus detalhes adicionais e selecione Detalhes, Mais... Se o status da instância selecionada for FAILED, a caixa de detalhes terá entradas para a mensagem de erro, o `errorStackTrace` e outras informações. Você pode salvar essas informações em um arquivo. Escolha OK.
5. No painel Instance summary, escolha Attempts, para ver os detalhes de cada linha de tentativa.

6. Para executar uma ação em sua instância incompleta ou com falha, marque a caixa de seleção ao lado da instância. Isso ativa as ações. Em seguida, selecione uma ação (Rerun | Cancel | Mark Finished).

Identificar o cluster do Amazon EMR que serve seu pipeline

Se um `EMRCluster` ou `EMRActivity` falhar e as informações de erro fornecidas pelo console do AWS Data Pipeline não forem claras, você poderá identificar o cluster do Amazon EMR que serve seu pipeline usando o console do Amazon EMR. Isso ajuda você a localizar os logs que o Amazon EMR fornece para obter mais detalhes sobre os erros que ocorrem.

Para obter informações de erro mais detalhadas do Amazon EMR

1. No console do AWS Data Pipeline, selecione o triângulo ao lado da instância do pipeline para expandir os detalhes da instância.
2. Escolha View execution details e, em seguida, o triângulo ao lado do componente.
3. Na coluna Details, escolha More.... A tela de informações é aberta listando os detalhes do componente. Localize e copie o valor `instanceParent` da tela, como:
`@EmrActivityId_xiFDD_2017-09-30T21:40:13`
4. Navegue até o console do Amazon EMR e pesquise um cluster com o valor correspondente `instanceParent` em seu nome e selecione Depurar.

Note

Para que o botão Debug funcione, a definição do seu pipeline deve configurar a opção `EmrActivity enableDebugging` como `true` e a opção `EmrLogUri` como um caminho válido.

5. Agora que você sabe qual cluster do Amazon EMR contém o erro que gera a falha do pipeline, siga as [Dicas de solução de problemas](#) no Guia do desenvolvedor do Amazon EMR.

Interpretar detalhes de status do pipeline

Os vários níveis de status exibidos no console e na CLI do AWS Data Pipeline indicam a condição de um pipeline e seus componentes. O status do pipeline é simplesmente uma visão geral de um pipeline. Para mais informações, veja o status dos componentes individuais do pipeline. Você pode

fazer isso clicando em um pipeline no console ou recuperando os detalhes do componente do pipeline usando a CLI.

Códigos de status

ACTIVATING

O componente ou recurso está sendo iniciado, como uma instância do EC2.

CANCELED

O componente foi cancelado por um usuário ou AWS Data Pipeline antes de ser executado. Isso pode acontecer automaticamente quando ocorre uma falha em um componente ou recurso diferente do qual esse componente depende.

CASCADE_FAILED

O componente ou recurso foi cancelado como em resposta a uma falha em cascata de uma de suas dependências, mas o componente provavelmente não era a fonte original da falha.

DEACTIVATING

O pipeline está sendo desativado.

FAILED

O componente ou recurso encontrou um erro e parou de funcionar. Quando há falha de um componente ou recurso, isso pode causar cancelamentos e falhas em cascata para outros componentes que dependem dele.

FINISHED

O componente concluiu o trabalho atribuído.

INACTIVE

O pipeline foi desativado.

PAUSED

O componente foi pausado e, no momento, não está executando seu trabalho.

PENDING

O pipeline está pronto para ser ativado pela primeira vez.

RUNNING

O recurso está sendo executado e pronto para receber trabalho.

SCHEDULED

O recurso está programado para ser executado.

SHUTTING_DOWN

O recurso está sendo encerrado após a conclusão bem-sucedida do trabalho.

SKIPPED

O componente pulou os intervalos de execução após a ativação do pipeline usando uma marca de tempo posterior à programação atual.

TIMEDOUT

O recurso excedeu o limite de `terminateAfter` e foi interrompido pelo AWS Data Pipeline. Depois que o recurso atinge esse status, AWS Data Pipeline ignora os valores de `actionOnResourceFailure`, `retryDelay` e `retryTimeout` para esse recurso. Esse status só é aplicável aos recursos.

VALIDATING

A definição de pipeline está sendo validada pelo AWS Data Pipeline.

WAITING_FOR_RUNNER

O componente está aguardando que o operador do cliente recupere um item de trabalho. O relacionamento entre componente e operador do cliente é controlado pelos campos `runsOn` ou `workerGroup` definidos por esse componente.

WAITING_ON_DEPENDENCIES

O componente está verificando se as pré-condições padrão e configuração pelo usuário foram atendidas antes de realizar seu trabalho.

Localizar logs de erro

Esta seção explica como encontrar os vários logs que o AWS Data Pipeline grava, que você pode usar para determinar a origem de determinadas falhas e erros.

Logs de pipeline

Recomendamos que você configure os pipelines para criar arquivos de log em um local persistente, como no exemplo a seguir, em que você usa o campo `pipelineLogUri` em um objeto `Default` do

pipeline para fazer com que todos os componentes usem um local do log do Amazon S3 por padrão (você pode substituir isso configurando um local do log em um componente específico do pipeline).

Note

O Task Runner armazena seus logs em um local diferente, por padrão, que pode estar indisponível quando o pipeline é concluído e a instância que executa o Task Runner é encerrada. Para obter mais informações, consulte [Verificando o registro do Task Runner](#).

Para configurar a localização do log usando a CLI do AWS Data Pipeline em um arquivo JSON do pipeline, comece o arquivo do pipeline com o seguinte texto:

```
{ "objects": [  
  {  
    "id":"Default",  
    "pipelineLogUri":"s3://mys3bucket/error_logs"  
  },  
  ...  
]
```

Depois de configurar um diretório de log de pipeline, o Task Runner cria uma cópia dos logs em seu diretório, com a mesma formatação e nomes de arquivos descritos na seção anterior sobre logs do Task Runner.

Logs de trabalho do Hadoop e Amazon EMR

Com qualquer atividade baseada em Hadoop, como [HadoopActivity](#), [HiveActivity](#) ou [PigActivity](#), você pode visualizar logs de trabalho do Hadoop no local retornados no slot de runtime, `hadoopJobLog`. O [EmrActivity](#) tem seus próprios recursos de log, e os logs são armazenados usando o local escolhido pelo Amazon EMR e retornados pelo slot de runtime, `emrStepLog`. Para obter mais informações, consulte [Visualizar arquivos de log](#) no Guia do Desenvolvedor do Amazon EMR.

Resolver problemas comuns

Este tópico fornece vários sintomas de problemas do AWS Data Pipeline e as etapas recomendadas para resolvê-los.

Índice

- [Pipeline preso em status pendente](#)

- [Componente de pipeline preso no status Waiting for Runner](#)
- [Componente de pipeline preso no status WAITING_ON_DEPENDENCIES](#)
- [A execução não inicia quando programada](#)
- [Os componentes do pipeline são executados na ordem errada](#)
- [O cluster do EMR falha com erro: o token de segurança incluído na solicitação é inválido](#)
- [Permissões insuficientes para acessar recursos](#)
- [Código de status: 400 Error Code: PipelineNotFoundException](#)
- [Criar um pipeline provoca um erro de token de segurança](#)
- [Não é possível ver detalhes do pipeline no console](#)
- [Erro no código de status do executor remoto: 404, AWS Service: Amazon S3](#)
- [Acesso negado – Não autorizado para executar a função datapipeline:](#)
- [AMIs do Amazon EMR mais antigas podem criar dados falsos em arquivos CSV grandes](#)
- [Aumentar limites do AWS Data Pipeline](#)

Pipeline preso em status pendente

Um pipeline que aparece travado com o status PENDING indica que ele ainda não foi ativado ou que a ativação falhou devido a um erro na definição do pipeline. Certifique-se de que você não recebeu nenhum erro quando enviou o pipeline usando a CLI do AWS Data Pipeline ou quando tentou salvar ou ativar o pipeline usando o console do AWS Data Pipeline. Além disso, verifique se o pipeline tem uma definição válida.

Para visualizar a definição do pipeline na tela usando a CLI:

```
aws datapipeline --get-pipeline-definition --pipeline-id df-EXAMPLE_PIPELINE_ID
```

Certifique-se de que a definição de pipeline foi concluída, verifique as chaves de fechamento, as vírgulas necessárias, as referências ausentes e outros erros de sintaxe. É melhor usar um editor de texto que pode validar visualmente a sintaxe de arquivos JSON.

Componente de pipeline preso no status Waiting for Runner

Se o pipeline está no estado SCHEDULED e uma ou mais tarefas aparecem presas no estado WAITING_FOR_RUNNER, assegure-se de que você configurou um valor válido para os campos runsOn ou workerGroup para essas tarefas. Se ambos os valores estão vazios ou ausentes, a

tarefa não pode ser iniciada porque não há associação entre a tarefa e um operador para executar as tarefas. Nesta situação, você definiu o trabalho, mas não definiu o computador que fará esse trabalho. Se aplicável, verifique se o valor `workerGroup` atribuído ao componente do pipeline tem exatamente o mesmo nome e caso do valor `workerGroup` que você configurou para Task Runner.

Note

Se você fornecer um valor de `runsOn` e `workerGroup` existir, `workerGroup` será ignorado.

Outra possível causa desse problema é que o endpoint e a chave de acesso fornecidas para Task Runner não são os mesmos que o console do AWS Data Pipeline ou computador em que as ferramentas da CLI da AWS Data Pipeline estão instaladas. Você pode criar novos pipelines, sem erros visíveis, mas o Task Runner consulta o local errado devido à diferença de credenciais, ou consulta o local correto com permissões insuficientes para identificar e executar o trabalho especificado pela definição do pipeline.

Componente de pipeline preso no status `WAITING_ON_DEPENDENCIES`

Se o pipeline está no estado `SCHEDULED` e uma ou mais tarefas aparecem presas no estado `WAITING_ON_DEPENDENCIES`, certifique-se de que as condições iniciais do seu pipeline foram atendidas. Se as condições do primeiro objeto na cadeia lógica não forem atendidas, nenhum dos objetos que dependem do primeiro objeto sairá do estado `WAITING_ON_DEPENDENCIES`.

Por exemplo, considere o trecho a seguir de uma definição de pipeline. Nesse caso, o objeto `InputData` tem uma condição `'Ready'` especificando que os dados devem existir antes que o objeto `InputData` seja concluído. Se os dados não existirem, o objeto `InputData` continuará no estado `WAITING_ON_DEPENDENCIES`, esperando os dados especificados pelo campo de caminho para se tornar disponível. Todos os objetos que têm a mesma dependência continuarão no estado `WAITING_ON_DEPENDENCIES` esperando que o objeto `InputData` alcance o estado `FINISHED`.

```
{
  "id": "InputData",
  "type": "S3DataNode",
  "filePath": "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
  "schedule":{"ref":"MySchedule"},
  "precondition": "Ready"
},
{
```

```
"id": "Ready",  
"type": "Exists"  
...
```

Além disso, verifique se seus objetos têm as permissões adequadas para acessar os dados. No exemplo acima, se as informações no campo de credenciais não tivessem permissões para acessar os dados especificados no campo do caminho, o objeto `InputData` ficaria preso no estado `WAITING_ON_DEPENDENCIES`, pois não poderia acessar os dados especificados pelo campo de caminho, mesmo se esses dados existissem.

Também é possível que um recurso comunicando-se com o Amazon S3 não tenha um endereço IP público associado a ele. Por exemplo, um `Ec2Resource` em uma sub-rede pública deve ter um endereço IP público associado a ela.

Por fim, em determinadas condições, instâncias de recursos podem atingir o estado `WAITING_ON_DEPENDENCIES` muito antes que suas atividades associadas sejam programadas para iniciar, o que pode oferecer a impressão de que o recurso ou a atividade não estão funcionando.

A execução não inicia quando programada

Verifique se você escolheu o tipo de programação correta que determina se sua tarefa começa no início do intervalo de programação (estilo Cron) ou no final do intervalo de programação (estilo de séries temporais).

Além disso, verifique se você especificou as datas corretamente em seus objetos de programação e se os valores `startDateTime` `endDateTime` estão no formato UTC, como neste exemplo:

```
{  
  "id": "MySchedule",  
  "startDateTime": "2012-11-12T19:30:00",  
  "endDateTime": "2012-11-12T20:30:00",  
  "period": "1 Hour",  
  "type": "Schedule"  
},
```

Os componentes do pipeline são executados na ordem errada

Você pode perceber que os horários de início e término dos seus componentes de pipeline são executados na ordem errada ou em uma sequência diferente da esperada. É importante

compreender que componentes de pipeline podem começar a ser executados simultaneamente se suas condições forem atendidas no tempo de inicialização. Em outras palavras, os componentes de pipeline não são executados sequencialmente por padrão. Se você precisa de uma determinada ordem de execução, deve controlar essa ordem com condições e campos `dependsOn`.

Verifique se você está usando o campo `dependsOn` preenchido com uma referência para os componentes corretos de pré-requisitos e se todos os ponteiros necessários entre os componentes estão presentes para alcançar a ordem que você precisa.

O cluster do EMR falha com erro: o token de segurança incluído na solicitação é inválido

Verifique suas funções do perfil do IAM, políticas e relações de confiança conforme descrito em [Funções do IAM para o AWS Data Pipeline](#).

Permissões insuficientes para acessar recursos

As permissões que você define em funções do perfil do IAM determinam se o AWS Data Pipeline pode acessar os clusters do EMR e as instâncias do EC2 para executar os pipelines. Além disso, o IAM; fornece o conceito de relacionamentos de confiança que vão além para permitir a criação dos recursos em seu nome. Por exemplo, quando você cria um pipeline que usa uma instância do EC2 para executar um comando para mover dados, o AWS Data Pipeline pode provisionar essa instância do EC2 para você. Se você encontrar problemas, especialmente aqueles que envolvem recursos que você pode acessar manualmente, mas o AWS Data Pipeline não pode, verifique suas funções do perfil do IAM, políticas e relacionamentos de confiança, como descrito em [Funções do IAM para o AWS Data Pipeline](#).

Código de status: 400 Error Code: PipelineNotFoundException

Este erro significa que as funções do IAM padrão podem não ter as permissões necessárias para que o AWS Data Pipeline funcione corretamente. Para obter mais informações, consulte [Funções do IAM para o AWS Data Pipeline](#).

Criar um pipeline provoca um erro de token de segurança

Você recebe o seguinte erro quando tenta criar um pipeline:

Falha ao criar pipeline com 'pipeline_name'. Erro: UnrecognizedClientException - O token de segurança incluído na solicitação é inválido.

Não é possível ver detalhes do pipeline no console

O filtro do pipeline do console do AWS Data Pipeline aplica-se à data de início programada de um pipeline, sem levar em consideração quando o pipeline foi enviado. É possível enviar um novo pipeline usando uma data de início programada que ocorre no passado, que o filtro de data padrão pode não exibir. Para ver os detalhes do pipeline, altere o filtro de data a fim de assegurar que a data de início programada do pipeline esteja no intervalo de datas do filtro.

Erro no código de status do executor remoto: 404, AWS Service: Amazon S3

Este erro significa que o Task Runner não pode acessar seus arquivos no Amazon S3. Verificar se:

- Suas credenciais estão definidas corretamente
- O bucket do Amazon S3 que você está tentando acessar existe
- Você está autorizado a acessar o bucket do Amazon S3

Acesso negado – Não autorizado para executar a função datapipeline:

Nos logs do Task Runner, você pode ver um erro semelhante ao seguinte:

- Código do status do ERRO: 403
- Serviço da AWS: DataPipeline
- Código de Erro da AWS: AccessDenied
- Mensagem de erro da AWS: Usuário: arn:aws:sts::XXXXXXXXXXXXX:federated-user/i-XXXXXXXX não está autorizado a executar: datapipeline: PollForTask.

Note

Nesta mensagem de erro, PollForTask pode ser substituído por nomes de outras permissões do AWS Data Pipeline.

Esta mensagem de erro indica que a função do perfil do IAM especificada precisa de permissões adicionais necessárias para interagir com o AWS Data Pipeline. Certifique-se de que sua política do perfil do IAM contenha as seguintes linhas, onde PollForTask é substituído pelo nome da permissão

que você deseja adicionar (use* para conceder todas as permissões). Para obter mais informações sobre como criar um novo perfil do IAM e aplicar uma política a ele, consulte [Gerenciar políticas do IAM](#) no guia Usar IAM.

```
{
  "Action": [ "datapipeline:PollForTask" ],
  "Effect": "Allow",
  "Resource": ["*"]
}
```

AMIs do Amazon EMR mais antigas podem criar dados falsos em arquivos CSV grandes

Em AMIs do Amazon EMR anteriores a 3.9 (3.8 e inferior), o AWS Data Pipeline usa um InputFormat personalizado para ler e gravar arquivos CSV para uso com trabalhos do MapReduce. Isso é usado quando o serviço prepara tabelas de e para o Amazon S3. Um problema com este InputFormat foi descoberto, em que a leitura de registros de grandes arquivos CSV pode resultar na produção de tabelas que não são corretamente copiadas. Este problema foi corrigido em versões posteriores do Amazon EMR. Use AMI do Amazon EMR 3.9 ou um Amazon EMR com versão 4.0.0 ou superior.

Aumentar limites do AWS Data Pipeline

Ocasionalmente, você pode exceder os limites de sistema específicos do AWS Data Pipeline. Por exemplo, o limite de pipeline padrão é de 20 pipelines com 50 objetos em cada um deles. Se você descobrir que vai precisar de mais pipelines do que o limite, considere mesclar vários pipelines para criar um número menor de pipelines com mais objetos em cada um deles. Para obter mais informações sobre os limites do AWS Data Pipeline, consulte [Limites do AWS Data Pipeline](#). No entanto, se você não conseguir contornar os limites usando a técnica de mesclar pipelines, solicite um aumento na sua capacidade usando este formulário: [Aumento de limite de pipeline de dados](#).

Limites do AWS Data Pipeline

Para garantir que haja capacidade para todos os usuários, o AWS Data Pipeline impõe limites para os recursos que você pode alocar e a taxa na qual você pode alocar recursos.

Índice

- [Limites da conta](#)
- [Limites de chamada do serviço web](#)
- [Considerações sobre escalabilidade](#)

Limites da conta

Os seguintes limites aplicam-se a uma única conta da AWS. Se precisar de capacidade adicional, você pode usar o [Formulário de solicitação da Central de suporte da Amazon Web Services](#) para aumentar sua capacidade.

| Atributo | Limite | Ajustável |
|--|--------|-----------|
| Número de pipelines | 100 | Sim |
| Número de objetos por pipeline | 100 | Sim |
| Número de instâncias ativas por objeto | 5 | Sim |
| Número de campos por objeto | 50 | Não |
| Número de UTF8 bytes por nome ou identificador | 256 | Não |
| Número de UTF8 bytes por campo | 10,240 | Não |

| Atributo | Limite | Ajustável |
|--|-----------------------------------|-----------|
| Número de UTF8 bytes por objeto | 15.360 (incluindo nomes de campo) | Não |
| Índice de criação de uma instância de um objeto | 1 por 5 minutos | Não |
| Novas tentativas de uma atividade de pipeline | 5 por tarefa | Não |
| Intervalo mínimo entre novas tentativas | 2 minutos | Não |
| Intervalo máximo de programação | 15 minutos | Não |
| Número máximo de sumarizações em um único objeto | 32 | Não |
| Número máximo de instâncias do EC2 por objeto do Ec2Resource | 1 | Não |

Limites de chamada do serviço web

O AWS Data Pipeline limita a taxa na qual você pode chamar a API de serviço web. Esses limites também se aplicam a agentes do AWS Data Pipeline que chamam a API de serviço web em seu nome, como o console, a CLI e o Task Runner.

Os seguintes limites aplicam-se a uma única conta da AWS. Isso significa que o uso total na conta, incluindo aquele por usuários do , não pode exceder esses limites.

A taxa de intermitência permite que você acumule chamadas de serviço web durante períodos de inatividade e use todas elas em um curto período. Por exemplo, CreatePipeline tem uma taxa regular de uma chamada a cada cinco segundos. Se você não chamar o serviço por 30 segundos, terá seis chamadas salvas. Em seguida, você pode chamar o serviço da web seis vezes em um segundo. Como esse preço está abaixo do limite de intermitência médio e mantém suas chamadas no limite de taxa regular, suas chamadas não são suspensas.

Se você exceder o limite de taxa e o limite de intermitência, a chamada de serviço web falha e retorna uma exceção de controle de utilização. A implementação padrão de um operador, Task Runner, tentará executar automaticamente as chamadas de API que falham com uma exceção do controle de utilização. O Task Runner tem um recuo para que as tentativas subsequentes de chamada da API ocorram em intervalos cada vez mais longos. Se você gravar um operador, recomendamos que implemente uma lógica semelhante de novas tentativas de trabalho.

Esses limites são aplicados em relação a uma conta individual da AWS.

| API | Limite de taxa regular | Limite de intermitência |
|-----------------------|-------------------------|-------------------------|
| ActivatePipeline | 1 chamada por segundo | 100 chamadas |
| CreatePipeline | 1 chamada por segundo | 100 chamadas |
| DeletePipeline | 1 chamada por segundo | 100 chamadas |
| DescribeObjects | 2 chamadas por segundo | 100 chamadas |
| DescribePipelines | 1 chamada por segundo | 100 chamadas |
| GetPipelineDefinition | 1 chamada por segundo | 100 chamadas |
| PollForTask | 2 chamadas por segundo | 100 chamadas |
| ListPipelines | 1 chamada por segundo | 100 chamadas |
| PutPipelineDefinition | 1 chamada por segundo | 100 chamadas |
| QueryObjects | 2 chamadas por segundo | 100 chamadas |
| ReportTaskProgress | 10 chamadas por segundo | 100 chamadas |
| SetTaskStatus | 10 chamadas por segundo | 100 chamadas |

| API | Limite de taxa regular | Limite de intermitência |
|----------------------------|------------------------|-------------------------|
| SetStatus | 1 chamada por segundo | 100 chamadas |
| ReportTaskRunnerHeartbeat | 1 chamada por segundo | 100 chamadas |
| ValidatePipelineDefinition | 1 chamada por segundo | 100 chamadas |

Considerações sobre escalabilidade

O AWS Data Pipeline pode ser dimensionado para acomodar uma grande quantidade de tarefas simultâneas, e você pode configurá-lo para criar automaticamente os recursos necessários para lidar com grandes cargas de trabalho. Esses recursos criados automaticamente são controlados por você e contam para os limites de recursos da sua conta da AWS. Por exemplo, se você configurar o AWS Data Pipeline para criar um cluster do Amazon EMR de 20 nós para processar dados, e sua conta da AWS tiver um limite de 20 instâncias do EC2, você poderá esgotar de maneira não intencional os seus recursos de alocação disponíveis. Por isso, considere essas restrições de recursos no seu projeto ou aumente os limites da sua conta.

Se precisar de capacidade adicional, você pode usar o [Formulário de solicitação da Central de suporte da Amazon Web Services](#) para aumentar sua capacidade.

Recursos da AWS Data Pipeline

Veja a seguir os recursos para ajudar você a usar o AWS Data Pipeline.

- [AWS Data Pipeline Informações sobre produto](#) – A principal página da web para obter informações sobre o AWS Data Pipeline.
- [AWS Data Pipeline Perguntas técnicas frequentes](#) – Abrange as 20 principais perguntas feitas pelos desenvolvedores sobre esse produto.
- [Notas de liberação](#) – Oferecem uma visão geral de alto nível da versão atual. Elas observam especificamente os novos recursos, correções e problemas conhecidos.
- [Fóruns de discussão do AWS Data Pipeline](#) – Um fórum comunitário para desenvolvedores discutirem questões técnicas relacionadas à Amazon Web Services.
- [Aulas e workshops](#) — Links para cursos de especialidades e baseados em perfil, bem como laboratórios autoguiados para ajudar a aperfeiçoar suas habilidades na AWS e a obter experiência prática.
- [Centro dos desenvolvedores da AWS](#) — Explore tutoriais, baixe ferramentas e informe-se sobre eventos para desenvolvedores da AWS.
- [Ferramentas do desenvolvedor da AWS](#) — Links para ferramentas de desenvolvedor, SDKs, toolkits de IDE e ferramentas da linha de comando para desenvolver e gerenciar aplicativos da AWS.
- [Centro de recursos de conceitos básicos](#) — Saiba como configurar a Conta da AWS, participar da comunidade da AWS e lançar seu primeiro aplicativo.
- [Tutoriais práticos](#) — Siga os tutoriais passo a passo para iniciar seu primeiro aplicativo na AWS.
- [Whitepapers da AWS](#) — Links para uma lista abrangente de whitepapers técnicos da AWS que abrangem tópicos como arquitetura, segurança e economia, elaborados pelos arquitetos de soluções da AWS ou por outros especialistas técnicos.
- [AWS Support Center](#): a central para criar e gerenciar seus casos do AWS Support. Também inclui links para outros recursos úteis, como fóruns, perguntas frequentes técnicas, status de integridade do serviço e AWS Trusted Advisor.
- [AWS Support](#) — A página Web principal para obter informações sobre o AWS Support, um canal de suporte de resposta rápida e com atendimento individual para ajudar a construir e a executar aplicativos na nuvem.

- [Entrar em contato](#) – Um ponto central de contato para consultas relativas a faturas da AWS, contas, eventos, uso abusivo e outros problemas.
- [Termos do site da AWS](#): informações detalhadas sobre nossos direitos autorais e marca registrada; sua conta, licença e acesso ao site, entre outros tópicos.

Histórico do documento

Esta documentação está associada à versão 2012-10-29 do. AWS Data Pipeline

| Alteração | Descrição | Data de lançamento |
|---|--|------------------------|
| AWS Data Pipeline não está mais disponível para novos clientes | AWS Data Pipeline não está mais disponível para novos clientes. Os clientes existentes do AWS Data Pipeline podem continuar usando o serviço normalmente. Saiba mais | 25 de julho de 2025 |
| Foi adicionada documentação para realizar determinados procedimentos usando AWS CLI o. Procedimentos relacionados ao AWS Data Pipeline console removidos. | Para mais informações, consulte Clonar o pipeline , Visualizar logs de pipeline e Crie pipelines a partir de modelos de Data Pipeline usando a CLI . | 26 de maio de 2023 |
| Foram adicionados mais conteúdo e amostras para AWS Data Pipeline migrar de outros serviços alternativos. | Atualizou o tópico para migrar AWS Data Pipeline para AWS Step Functions ou Amazon MWAA com mais informações sobre cada alternativa, mapeamentos conceituais entre os serviços e amostras. AWS Glue Para obter mais informações, consulte Migrando cargas de trabalho do AWS Data Pipeline . | 31 de março de 2023 |
| Foram adicionadas informações sobre AWS Data Pipeline o suporte doIMDSv2. | AWS Data Pipeline suportes IMDSv2 para a Amazon EMR e EC2 os recursos da Amazon. Para mais informações, consulte Proteção de dados no AWS Data Pipeline , EmrCluster e Ec2Resource . | 16 de dezembro de 2022 |
| Foi adicionado um tópico para AWS Data Pipeline migrar | Agora existem outros AWS serviços que oferecem aos clientes uma melhor experiência de integração de dados. Você pode migrar casos de uso típicos | 16 de dezembro de 2022 |

| Alteração | Descrição | Data de lançamento |
|---|---|-----------------------|
| de outros serviços alternativos. | AWS Data Pipeline para AWS Step Functions ou AmazonMWS. AWS Glue Para obter mais informações, consulte Migrando cargas de trabalho do AWS Data Pipeline . | |
| Atualizou as listas de EMR instâncias da Amazon EC2 e da Amazon compatíveis. Atualizou a lista IDs das HVM (Máquina Virtual de Hardware) AMIs usadas para as instâncias. | Atualizou as listas de EMR instâncias da Amazon EC2 e da Amazon compatíveis. Para obter mais informações, consulte Tipos de instância com suporte para as atividades de trabalho do pipeline . Atualizou a lista IDs das HVM (Máquina Virtual de Hardware) AMIs usadas para as instâncias. Para obter mais informações, consulte Sintaxe e pesquise <code>imageId</code> . | 9 de novembro de 2018 |

| Alteração | Descrição | Data de lançamento |
|--|---|--------------------------------|
| <p>Configuração adicionada para anexar EBS volumes da Amazon aos nós do cluster e para iniciar um EMR cluster da Amazon em uma sub-rede privada.</p> | <p>Adição de opções de configuração a um objeto <code>EMRCluster</code> . Você pode usar essas opções em pipelines que usam EMR clusters da Amazon.</p> <p>Use os <code>TaskEbsConfiguration</code> campos <code>coreEbsConfiguration</code> <code>masterEbsConfiguration</code> , e para configurar a vinculação dos EBS volumes da Amazon aos nós principais, principais e de tarefas no EMR cluster da Amazon. Para obter mais informações, consulte Anexe EBS volumes aos nós do cluster.</p> <p>Use os <code>ServiceAccessSecurityGroupId</code> campos <code>emrManagedMasterSecurityGroupId</code> <code>emrManagedSlaveSecurityGroupId</code> , e para configurar um EMR cluster da Amazon em uma sub-rede privada. Para obter mais informações, consulte Configurar um EMR cluster da Amazon em uma sub-rede privada.</p> <p>Para mais informações sobre sintaxe de <code>EMRCluster</code> , consulte EmrCluster.</p> | <p>19 de abril de 2018</p> |
| <p>Foi adicionada a lista de EMR instâncias da Amazon EC2 e da Amazon compatíveis.</p> | <p>Foi adicionada a lista de instâncias que são AWS Data Pipeline criadas por padrão, caso você não especifique um tipo de instância na definição do pipeline. Foi adicionada uma lista de EMR instâncias da Amazon EC2 e da Amazon compatíveis. Para obter mais informações, consulte Tipos de instância com suporte para as atividades de trabalho do pipeline.</p> | <p>22 de março de 2018</p> |
| <p>Adição do suporte aos pipelines sob demanda</p> | <ul style="list-style-type: none"> Mais suporte aos pipelines sob demanda, o que permite que você execute novamente um pipeline ao reativá-lo. | <p>22 de fevereiro de 2016</p> |

| Alteração | Descrição | Data de lançamento |
|--|---|------------------------|
| Suporte adicional para RDS bancos de dados | <ul style="list-style-type: none"> <code>rdsInstanceId</code>, <code>region</code> e <code>jdbcDriverJarUri</code> adicionados a RdsDatabase. <code>database</code> atualizado em SqlActivity para oferecer suporte a <code>RdsDatabase</code> também. | 17 de agosto de 2015 |
| JDBCSuporte adicional | <ul style="list-style-type: none"> <code>database</code> atualizado em SqlActivity para oferecer suporte a <code>JdbcDatabase</code> também. Adição de <code>jdbcDriverJarUri</code> a JdbcDatabase <code>initTimeout</code> adicionado a Ec2Resource e EmrCluster. Adição de <code>runAsUser</code> a Ec2Resource. | 7 de julho de 2015 |
| HadoopActivity, Availability Zone e Spot Support | <ul style="list-style-type: none"> Suporte adicionado para enviar trabalhos paralelos aos clusters do Hadoop. Para obter mais informações, consulte HadoopActivity. Capacidade de solicitar instâncias spot com Ec2Resource e EmrCluster. Capacidade de iniciar recursos <code>EmrCluster</code> em uma zona de disponibilidade específica. | 1 de junho de 2015 |
| Desativar pipelines | Suporte adicional à desativação de pipelines ativos. Para obter mais informações, consulte Desativar o pipeline . | 7 de abril de 2015 |
| Modelos e console atualizados | Adição de novos modelos. Atualizou o capítulo Introdução para usar o <code>ShellCommandActivity</code> modelo Introdução. Para obter mais informações, consulte Crie pipelines a partir de modelos de Data Pipeline usando a CLI . | 25 de novembro de 2014 |

| Alteração | Descrição | Data de lançamento |
|---|--|-------------------------|
| VPCapio | Foi adicionado suporte para lançar recursos em uma nuvem privada virtual (VPC). | 12 de março de 2014 |
| Suporte de região | Suporte adicionado para várias regiões de serviços. Além de us-east-1, AWS Data Pipeline é suportado em eu-west-1, ap-northeast-1, ap-southeast-2, us-west-2 e. | 20 de fevereiro de 2014 |
| Suporte a Amazon Redshift | Foi adicionado suporte para o Amazon Redshift em AWS Data Pipeline, incluindo um novo modelo de console (Copy to Redshift) e um tutorial para demonstrar o modelo. Para obter mais informações, consulte Copiar dados para o Amazon Redshift usando AWS Data Pipeline , RedshiftDataNode , RedshiftDatabase e RedshiftCopyActivity . | 6 de novembro de 2013 |
| PigActivity | Adicionado PigActivity, que fornece suporte nativo para o Pig. Para obter mais informações, consulte PigActivity . | 15 de outubro de 2013 |
| Modelo, atividade e formato de dados novos do console | Foi adicionado o novo CrossRegion modelo de console do DynamoDB Copy, incluindo o HiveCopyActivity novo e D. ynamoDBExport DataFormat | 21 de agosto de 2013 |
| Falhas e novas execuções em cascata | Foram adicionadas informações sobre falha AWS Data Pipeline em cascata e comportamento de reexecução. Para obter mais informações, consulte Falhas e novas execuções em cascata . | 8 de agosto de 2013 |
| Vídeo sobre a solução de problemas | Foi adicionado o vídeo de solução de problemas AWS Data Pipeline básicos. Para obter mais informações, consulte Solução de problemas . | 17 de julho de 2013 |

| Alteração | Descrição | Data de lançamento |
|---|--|-------------------------|
| Editar pipelines ativos | Mais informações adicionadas sobre como editar pipelines ativos e executar novamente os componentes do pipeline. Para obter mais informações, consulte Editar o pipeline . | 17 de julho de 2013 |
| Usar recursos em diferentes regiões | Mais informações adicionadas sobre como usar recursos em diferentes regiões. Para obter mais informações, consulte Usar um pipeline com recursos em várias regiões . | 17 de junho de 2013 |
| WAITINGStatus_ON_DEPENDENCIES | CHECKING_ PRECONDITIONS o status mudou para WAITING _ON_ DEPENDENCIES e adicionou o campo @ waitingOn runtime para objetos de pipeline. | 20 de maio de 2013 |
| ynamoDBDataFormato D | Modelo de ynamoDBData formato D adicionado. | 23 de abril de 2013 |
| Vídeo Processar logs da web e suporte a instâncias spot | Apresentou o vídeo “Process Web Logs with AWS Data PipelineEMR, Amazon e Hive” e o suporte para Amazon EC2 Spot Instances. | 21 de fevereiro de 2013 |
| | A versão inicial do Guia do AWS Data Pipeline Desenvolvedor. | 20 de dezembro de 2012 |