AWS Data Pipeline

Guia do desenvolvedor

Versão da API 2012-10-29

AWS Data Pipeline: Guia do desenvolvedor

Table of Contents

O que é o AWS Data Pipeline?

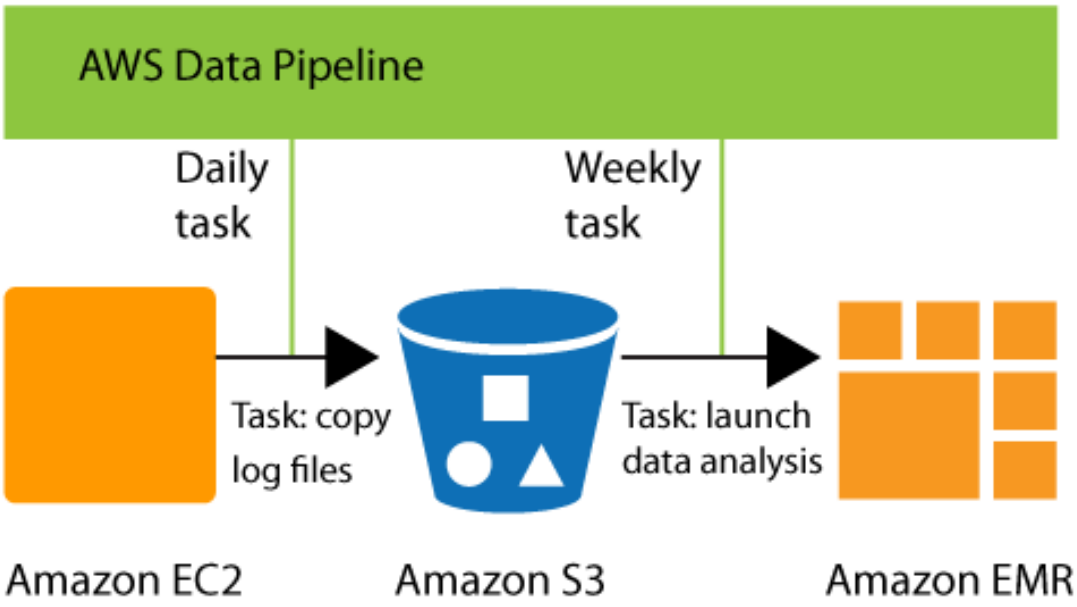
> **Note**
>
> AWS Data Pipelineo serviço está em modo de manutenção e nenhum novo recurso ou expansão de região está planejado. Para saber mais e descobrir como migrar suas cargas de trabalho existentes, consulte. Migração de cargas de trabalho do AWS Data Pipeline (p. 2)

O AWS Data Pipeline é um serviço web que você pode usar para automatizar a movimentação e a transformação de dados. Com o AWS Data Pipeline, você pode definir fluxos de trabalho dirigidos por dados para que as tarefas possam ser dependentes da conclusão bem-sucedida das tarefas anteriores. Você define os parâmetros das transformações dos seus dados, e o AWS Data Pipeline impõe a lógica configurada.

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

- Uma definição de pipeline especifica a lógica de negócios do seu gerenciamento de dados. Para obter mais informações, consulte Sintaxe do arquivo de definição de pipeline (p. 54).
- Um pipeline agenda e executa tarefas criando instâncias do Amazon EC2 para realizar as 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 pesquisa as tarefas e, em seguida, executa essas tarefas. 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 em recursos criados pelas definições do seu pipeline. Você pode escrever um aplicativo executor de tarefas personalizado ou usar o aplicativo Executor de Tarefas fornecido pelo. AWS Data Pipeline Para obter mais informações, consulte Executores de tarefas (p. 13).

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 depois executar um cluster semanal do Amazon EMR (Amazon EMR) sobre esses registros para gerar relatórios de tráfego. AWS Data Pipelineagenda as tarefas diárias para copiar dados e a tarefa semanal para iniciar o cluster do Amazon EMR. AWS Data Pipelinetambém garante que o Amazon EMR aguarde que os dados do último dia sejam enviados para o Amazon S3 antes de iniciar sua análise, mesmo que haja um atraso imprevisto no upload dos registros.



Índice

Migração de cargas de trabalho do AWS Data Pipeline

AWSlançou 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, AWS Step Functions para ajudar a orquestrar componentes de AWS serviços 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 migrar de AWS Data Pipeline para opções alternativas. A opção escolhida depende da sua carga de trabalho atual. AWS Data Pipeline Você pode migrar casos de uso típicos do AWS Data Pipeline AWS Step Functions ou do 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. Ele inclui ferramentas para criação, execução de trabalhos e orquestração de fluxos de trabalho. Com o AWS Glue,

você pode detectar 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 está procurando um serviço de integração de dados sem servidor que ofereça suporte a 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 confidenciais.
- 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 sobre o Apache Spark).
- Você precisa de uma única plataforma 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 exportar uma tabela do DynamoDB para o Amazon S3, e você está procurando o mesmo modelo de propósito.
- Sua carga de trabalho não depende de um aplicativo específico do ecossistema Hadoop, como o Apache Hive.
- Sua carga de trabalho não exige a orquestração de servidores locais.

AWScobra uma taxa horária, cobrada por segundo, para rastreadores (descoberta de dados) e trabalhos de ETL (processamento e carregamento de dados). AWS Glue O Studio é um mecanismo integrado de orquestração de AWS Glue recursos e é oferecido sem custo adicional. Saiba mais sobre preços em AWS GluePreços.

Migração de cargas de trabalho para AWS Step Functions

AWSO 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 com 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 canais de processamento de dados, lidar com erros e trabalhar com os limites de limitação dos 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 aplicativos que exigem interação humana.

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

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

- Você está procurando um serviço de orquestração de fluxo de trabalho altamente disponível e sem servidor.
- Você está procurando uma solução econômica que cobre a 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 de baixo código que venha com um designer drag-and-drop visual para criação de fluxo de trabalho e não exija o aprendizado de novos conceitos de programação.

- Você está procurando um serviço que ofereça integrações com mais de 250 outros AWS serviços, abrangendo mais de 11.000 açõesout-of-the-box, além de permitir integrações com atividades e não AWS serviços personalizados.

AWS Data PipelineTanto o Step Functions quanto o Step 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. AWS Data Pipeline

Para migrar atividades em recursos AWS Data Pipeline gerenciados, você pode usar a integração do serviço AWS SDK no Step Functions para automatizar o provisionamento e a limpeza de recursos.

Para migrar atividades em servidores locais, instâncias EC2 gerenciadas pelo usuário ou um cluster EMR gerenciado pelo usuário, você pode instalar um agente SSM na instância. Você pode iniciar o comando por meio do AWSSystems Manager Run Command from Step Functions. Você também pode iniciar a máquina de estado a partir do cronograma definido na Amazon EventBridge.

AWSO Step Functions tem dois tipos de fluxos de trabalho: fluxos de trabalho padrão e fluxos de trabalho expressos. Para fluxos de trabalho padrão, você é cobrado com base no número de transições de estado necessárias para executar seu aplicativo. Para fluxos de trabalho expressos, você é cobrado com base no número de solicitações do seu fluxo de trabalho e sua duração. Saiba mais sobre preços em AWSStep Functions Pricing.

Migração de cargas de trabalho para o Amazon MWAA

O Amazon MWAA (Managed Workflows for Apache Airflow) é um serviço de orquestração gerenciado para o Apache Airflow que facilita a configuração e a operação de pipelines de dados de ponta a ponta na nuvem em grande escala. O Apache Airflow é uma ferramenta de código aberto usada para criar, agendar e monitorar programaticamente sequências de processos e tarefas chamadas de “fluxos de trabalho”. Com o Amazon MWAA, você pode usar a linguagem de programação Airflow e Python para criar fluxos de trabalho sem precisar gerenciar a infraestrutura subjacente para 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 queAWS Data Pipeline, o Amazon MWAA é um serviço totalmente gerenciado fornecido pela. AWS Embora você precise aprender vários conceitos novos específicos desses serviços, não é necessário gerenciar a infraestrutura, aplicar patches, gerenciar atualizações de versão do sistema operacional ou similares.

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

- Você está procurando um serviço gerenciado e altamente disponível para orquestrar fluxos de trabalho escritos em Python.
- Você quer fazer a transição para uma tecnologia de código aberto totalmente gerenciada e amplamente adotada, o Apache Airflow, para máxima portabilidade.
- Você precisa de uma única plataforma que possa lidar com todos os aspectos do seu pipeline de dados, incluindo ingestão, processamento, transferência, testes de integridade e verificações de qualidade.
- Você está procurando um serviço projetado para orquestração de pipelines de dados com recursos como interface de usuário avançada para observabilidade, reinicializações em caso de falhas em fluxos de trabalho, 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 gráficos acíclicos direcionados (DAGs) usando Python, então você também pode tratá-los como código-fonte. A estrutura extensível em Python do Airflow permite que você crie fluxos de trabalho conectando-se a praticamente qualquer tecnologia. Ele vem com uma interface de usuário avançada para visualização e monitoramento de 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. AWS Data Pipeline

AWScobra pelo tempo em que seu ambiente Airflow funciona, além de qualquer escalonamento automático adicional para fornecer mais capacidade de trabalho ou servidor web. Saiba mais sobre preços nos preços do Amazon Managed Workflows for Apache Airflow.

Mapeando os conceitos

A tabela a seguir contém o mapeamento dos principais conceitos usados pelos serviços. Isso ajudará as pessoas familiarizadas com o Data Pipeline a entender as funções de etapas e a terminologia do MWAA.

Data Pipeline	União	Step Functions	Amazon MWAA
Pipelines	Fluxos de trabalho	Fluxos de trabalho	Gráficos de acrílico direto
Definição de pipeline JSON	Definição de fluxo de trabalho ou esquemas baseados em Python	JSON da linguagem estadual da Amazon	Baseado em Python
Atividades	Trabalhos	Estados e tarefas	Tarefas (operadores e sensores)
Instâncias	O trabalho é executado	Execuções	O DAG é executado
Attempts	Tentativas de repetição	Catchers e retriers	Tenta novamente
Cronograma do pipeline	Agendar gatilhos	EventBridgeTarefas do agendador	Cron, horários, com reconhecimento de dados
Expressões e funções do pipeline	Biblioteca de Blueprint	Funções Step: funções intrínsecas e Lambda AWS	Estrutura extensível em Python

Amostras

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

Exemplos do AWS Glue

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

- Executando 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
- Movendo dados de e para o Amazon Redshift
- Acesso entre contas entre regiões às tabelas do DynamoDB

AWSExemplos do Step Functions

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

- Gerenciando um trabalho do Amazon EMR
- Executando um trabalho de processamento de dados no Amazon EMR Serverless
- Executando trabalhos do Hive/Pig/Hadoop
- Consultar grandes conjuntos de dados (Amazon Athena, Amazon S3,) AWS Glue
- Execução de fluxos de trabalho de ETL usando o Amazon Redshift
- AWS GlueOrquestrando rastreadores

Veja tutoriais adicionais e exemplos de projetos para usar o AWS Step Functions.

Amostras de MWAA da Amazon

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

- Executando um trabalho do Amazon EMR
- Criação de um plug-in personalizado para Apache Hive e Hadoop
- Copiar dados do Amazon S3 para o Redshift
- Executando um script do Shell em uma instância remota do EC2
- Orquestrando fluxos de trabalho híbridos (no local)

Veja outros tutoriais e exemplos de projetos para usar o Amazon MWAA.

Serviços relacionados

O AWS Data Pipeline trabalha com os serviços a seguir para armazenar dados.

- Amazon DynamoDB — Fornece um banco de dados NoSQL totalmente gerenciado com desempenho rápido e 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 pode ser dimensionado para grandes conjuntos de dados. Para obter mais informações, consulte o Guia do desenvolvedor do Amazon Relational Database Service.
- Amazon Redshift — Fornece um armazém de dados rápido, totalmente gerenciado e em escala de petabytes que torna fácil e econômico analisar uma grande quantidade de dados. Para obter mais informações, consulte o Guia do desenvolvedor de banco de dados do Amazon Redshift.
- Amazon S3 — Fornece armazenamento de objetos seguro, durável e altamente escalável. Para obter mais informações, consulte o Guia do usuário do Amazon Simple Storage Service.

O AWS Data Pipeline trabalha com os serviços de computação a seguir para transformar dados.

- Amazon EC2 — Fornece capacidade computacional redimensionável — literalmente, servidores nos data centers da Amazon — que você usa para criar e hospedar seus sistemas de software. Para obter mais informações, consulte o Guia do usuário do Amazon EC2 para instâncias Linux.
- Amazon EMR — torna fácil, rápido e econômico distribuir e processar grandes quantidades de dados nos servidores Amazon EC2, usando uma estrutura como o Apache Hadoop ou o Apache Spark. Para obter mais informações, consulte o Guia do desenvolvedor do Amazon EMR.

Como acessar o 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 acessarAWS Data Pipeline.
- AWS Command Line Interface(AWS CLI) — Fornece comandos para um amplo conjunto de serviços da AWSAWS Data Pipeline, incluindo e é compatível com Windows, macOS e Linux. Para obter mais informações sobre como instalar a AWS CLI, consulte AWS Command Line Interface. Para obter uma lista de comandos do 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 AWS SDKs.
- 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.

Preços

Com o Amazon Web Services, você paga somente pelo que usar. No AWS Data Pipeline, você paga pelo pipeline com base na frequência com que suas atividades e precondições estão programas para execução e no local onde elas serão 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 precondições 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 preços 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 tipos de instância do Amazon Linux AMI.

A tabela a seguir descreve os tipos de instância compatíveis com o AWS Data Pipeline. 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 com suporte para o AWS Data Pipeline, consulte Regiões e endpoints da AWS.

Índice

Instâncias padrão do Amazon EC2 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 AWS Data Pipeline são usadas por padrão nas regiões em que AWS Data Pipeline há suporte.

Nome da região	Região	Tipo de instância
Leste dos EUA (N. 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 AWS Data Pipeline são 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
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

Nome da região	Região	Tipo de instância
UE (Londres)	eu-west-2	t2.small
UE (Paris)	eu-west-3	t2.small
América do Sul (São Paulo)	sa-east-1	m1.small

Instâncias adicionais suportadas do Amazon EC2

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 criar, 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
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 suportadas para clusters do Amazon EMR

Esta tabela lista as instâncias do Amazon EC2 que oferecem AWS Data Pipeline suporte e podem criar 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

Classe de instância	Instance Types (Tipos de instâncias)
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 | m4.16large | m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge | 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 | i2.4large | i2.8xlarge | d2.xlarge | d2.2xlarge| d2.4xlarge | d2.8xlarge
Computação acelerada	g2.2xlarge | cg1.4xlarge

Conceitos do AWS Data Pipeline

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

Índice

Definição de pipeline

Uma definição de pipeline é como você comunica sua lógica de negócios ao 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

Na definição do seu pipeline, o AWS Data Pipeline determina, programa e atribui as tarefas aos executores de tarefas. Se uma tarefa não for concluída com sucesso, o AWS Data Pipeline tentará executá-la novamente de acordo com as suas instruções e, se necessário, atribuirá novamente essa tarefa a outro executor de tarefas. Se a tarefa falhar repetidamente, você poderá configurar o pipeline para lhe notificar.

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

Você pode criar uma definição de pipeline das seguintes formas:

- Graficamente com o console do 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 (p. 14)

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

Atividades (p. 14)

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

Precondições (p. 15)

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

Recursos (p. 16)

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

Ações (p. 18)

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

Para obter mais informações, consulte Sintaxe do arquivo de definição de pipeline (p. 54).

Componentes, instâncias e tentativas de pipeline

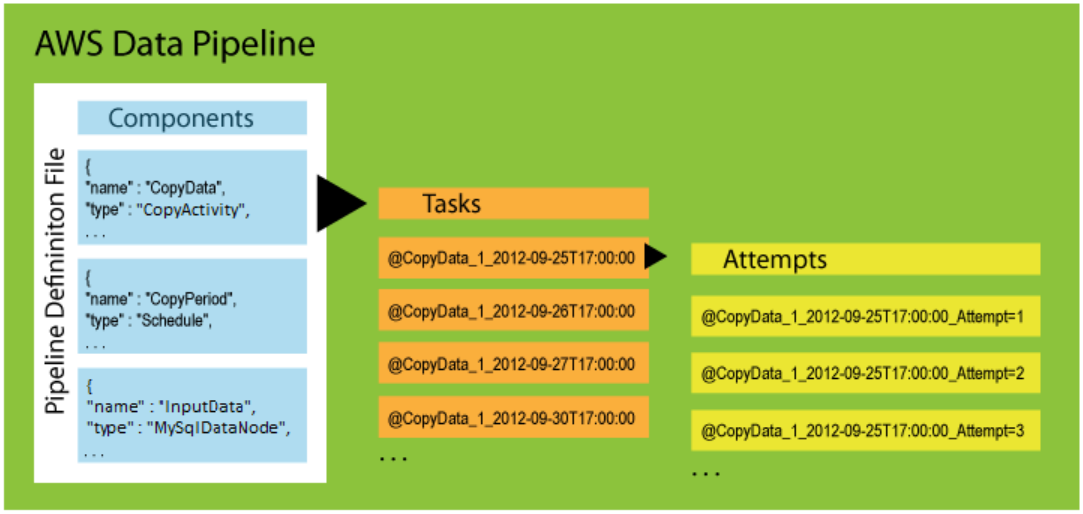
Existem três tipos de itens associados a um pipeline programado:

- Componentes do pipeline — Os componentes do pipeline representam a lógica de negócios do pipeline e são representados pelas diferentes seções da definição de um 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.
- Instances – Quando o 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. O AWS Data Pipeline entrega as instâncias aos executores de tarefas para processamento.
- 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 Pipelineexecuta novas tentativas usando os mesmos recursos das tentativas anteriores, como clusters do Amazon EMR e instâncias do 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 irrecuperável, 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. Por isso, considere cuidadosamente quando é apropriado

exceder as configurações padrão do AWS Data Pipelineusadas para controlar novas tentativas e configurações relacionadas.

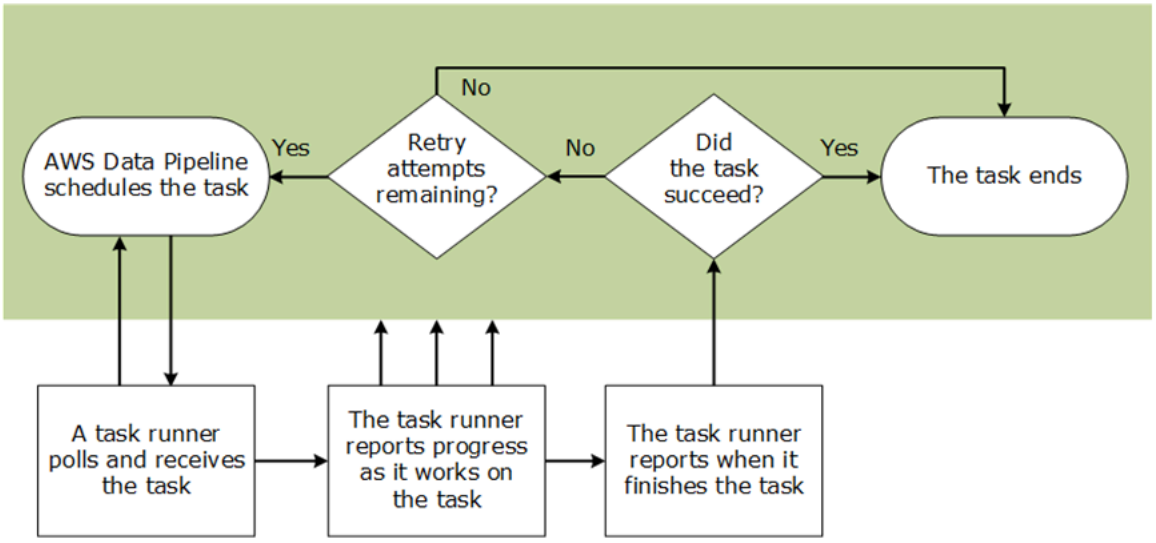


Executores de tarefas

Um executor de tarefas é um aplicativo que pesquisa tarefas no AWS Data Pipeline e, em seguida, executa essas tarefas.

O Executor de Tarefas é uma implementação padrão de um executor de tarefas que é 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 Executor de Tarefas, ele executa essa tarefa e reporta seu status paraAWS Data Pipeline.

O diagrama a seguir ilustra como o AWS Data Pipeline e um executor de tarefas interagem para processar uma tarefa programada. Uma tarefa é uma unidade de trabalho distinta que o serviço do AWS Data Pipeline 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.



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

- AWS Data Pipelineinstala o Task Runner para você em recursos que são lançados e gerenciados pelo serviço AWS Data Pipeline web.
- Você instala o Task Runner em um recurso computacional que você gerencia, como uma instância EC2 de longa execução ou um servidor local.

Para obter mais informações sobre como trabalhar com o Task Runner, consulte Trabalhando com o Task Runner (p. 277).

Nós de dados

No AWS Data Pipeline, um nó de dados define o local e o tipo de dados que uma atividade de pipeline usa como entrada ou saída. O AWS Data Pipeline oferece suporte aos seguintes tipos de nós de dados:

DynamoDBDataNode (p. 115)

Uma tabela do DynamoDB que contém dados para HiveActivity (p. 159) ou EmrActivity (p. 146) para usar.

SqlDataNode (p. 135)

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 (p. 125)

Uma tabela do Amazon Redshift que contém dados RedshiftCopyActivity (p. 181) para uso.

S3DataNode (p. 130)

Um local do Amazon S3 que contém um ou mais arquivos para uso em uma atividade de pipeline.

Bancos de dados

O AWS Data Pipeline oferece suporte aos seguintes tipos de bancos de dados:

JdbcDatabase (p. 249)

Um banco de dados JDBC.

RdsDatabase (p. 250)

Um banco de dados do Amazon RDS.

RedshiftDatabase (p. 252)

Um banco de dados do Amazon Redshift.

Atividades

No AWS Data Pipeline, uma atividade é um componente de pipeline que define o trabalho a ser realizado. O AWS Data Pipeline fornece várias atividades pré-empacotadas que acomodam cenários comuns, como o movimento de dados de um local para outro, a execução de 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.

O AWS Data Pipeline oferece suporte aos seguintes tipos de atividades:

CopyActivity (p. 140)

Copia dados de um local para outro.

EmrActivity (p. 146)

Executa um cluster do Amazon EMR.

HiveActivity (p. 159)

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

HiveCopyActivity (p. 165)

Executa uma consulta do Hive em um cluster do Amazon EMR com suporte para filtragem avançada de dados e suporte para e. S3DataNode (p. 130) DynamoDBDataNode (p. 115)

PigActivity (p. 171)

Executa um script Pig em um cluster do Amazon EMR.

RedshiftCopyActivity (p. 181)

Copia dados de e para tabelas do Amazon Redshift.

ShellCommandActivity (p. 190)

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

SqlActivity (p. 196)

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 obter mais informações, consulte Preparar dados e tabelas com atividades de pipeline (p. 47).

Precondições

No AWS Data Pipeline, uma precondição é um componente de pipeline que contém instruções condicionais que precisam 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 Pipelinefornece várias condições prévias pré-empacotadas 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 precondições: as gerenciadas pelo sistema e as gerenciadas pelo usuário. As precondições gerenciadas pelo sistema são gerenciadas pelo serviço web do AWS Data Pipeline em seu nome e não exigem um recurso computacional. As precondições 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 precondição.

Precondições gerenciadas pelo sistema

DynamoDBDataExists (p. 231)

Verifica se existem dados em uma tabela específica do DynamoDB.

DynamoDBTableExists (p. 233)

Verifica se existe uma tabela do DynamoDB.

S3KeyExists (p. 239)

Verifica se existe uma chave do Amazon S3.

S3PrefixNotEmpty (p. 242)

Verifica se um prefixo do Amazon S3 está vazio.

Precondições gerenciadas pelo usuário

Existe (p. 236)

Verifica se um nó de dados existe.

ShellCommandPrecondition (p. 245)

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

Recursos

No AWS Data Pipeline, um recurso é o recurso computacional que executa o trabalho que uma atividade de pipeline especifica. O AWS Data Pipeline oferece suporte aos seguintes tipos de recursos:

Ec2Resource (p. 201)

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

EmrCluster (p. 208)

Um cluster do Amazon EMR que executa o trabalho definido por uma atividade de pipeline, como. EmrActivity (p. 146)

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 obter mais informações, consulte Usar um pipeline com recursos em várias regiões (p. 51).

Limites de recurso

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 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 instâncias do EC2 definido como 20, você poderá esgotar inadvertidamente 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 para instâncias Linux.

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 AWS Data Pipeline

Os pipelines podem usar instâncias spot do Amazon EC2 para os nós de tarefas em 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 da sua instância spot para o Amazon EMR quando seu cluster é lançado. Ele alocará automaticamente o trabalho do cluster para o número de nós de tarefa da instância spot que você definiu usando o campo `taskInstanceCount`. O AWS Data Pipeline limita instâncias spot para nós de tarefas a fim de garantir que os nós core 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 as instâncias spot com o AWS Data Pipeline, as seguintes considerações se aplicam:

- Suas instâncias spot podem ser encerradas quando o preço da instância spot ultrapassar o preço máximo da instância ou devido a motivos de capacidade do Amazon EC2. No entanto, você não perderá seus dados, pois o AWS Data Pipeline emprega clusters com nós core que são sempre instâncias sob demanda e não estão sujeitos a encerramento.
- 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

As ações do AWS Data Pipeline são etapas que um componente de pipeline percorre quando ocorrem determinados eventos, como atividades realizadas com sucesso, falha ou atraso. O campo de evento de uma atividade refere-se a uma ação, como uma referência a `snsalarm` no campo `onLateAction` de `EmrActivity`.

AWS Data Pipelinedepende 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 console e a CLI do AWS Data Pipeline para obter informações de status do pipeline.

O AWS Data Pipeline oferece suporte às seguintes ações:

SnsAlarm (p. 263)

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

Encerrar (p. 264)

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 inicia na hora de início programada. AWS Data Pipelinefacilita a configuração de notificações fornecendo campos de eventos nos componentes do pipeline que você pode associar às notificações do Amazon SNS, como `onSuccessOnFail`, e. `onLateAction`

Configuração do AWS Data Pipeline

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

Tarefas

- Cadastre-se no AWS (p. 19)
- Crie funções do IAMAWS Data Pipeline e recursos de pipeline (p. 20)
- Permita que os diretores do IAM (usuários e grupos) realizem as ações necessárias (p. 20)
- Conceder acesso programático (p. 21)

Depois de concluir essas tarefas, você pode começar a usar o AWS Data Pipeline. Para ver um tutorial básico, consulte Conceitos básicos do AWS Data Pipeline (p. 23).

Cadastre-se no AWS

Quando você se cadastra na Amazon Web Services (AWS), a conta da AWS é cadastrada automaticamente em todos os serviços da AWS, incluindo o AWS Data Pipeline. Você será cobrado apenas pelos serviços que usar. Para obter mais informações sobre as taxas de uso do AWS Data Pipeline, consulte AWS Data Pipeline.

Cadastrar-se em uma Conta da AWS

Se você ainda não tem Conta da AWS, siga as etapas a seguir para criar um.

Para se cadastrar em uma Conta da AWS

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

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

   Quando você se cadastra em uma 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 acesso administrativo a um usuário administrativo e use somente o usuário raiz para realizar as tarefas que exigem acesso do usuário raiz.

AWSA envia um e-mail de confirmação depois que o processo de cadastramento é concluído. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando My Account (Minha conta).

Criar um usuário administrador

Depois de se inscrever em uma Conta da AWS, crie um usuário administrativo para não usar o usuário raiz em tarefas cotidianas.

Proteger seu Usuário raiz da conta da AWS

1. Faça login no AWS Management Console como o proprietário da conta ao escolher a opção Root user (Usuário raiz) e inserir o endereço de e-mail da Conta da AWS. Na próxima página, insira 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 MFA virtual para o usuário raiz de sua conta da Conta da AWS (console) no Guia do usuário do IAM.

Criar um usuário administrador

- Para suas tarefas administrativas diárias, conceda acesso administrativo a um usuário administrativo no AWS IAM Identity Center (successor to AWS Single Sign-On).

  Para obter instruções, consulte Getting started (Introdução) no Manual do usuário do AWS IAM Identity Center (successor to AWS Single Sign-On).

Fazer login como usuário administrador

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

  Para obter ajuda com o login utilizando um usuário do Centro de Identidade do IAM, consulte Fazer login no portal de acesso da AWS, no Guia do usuário do Início de Sessão da AWS.

Crie funções do IAMAWS Data Pipeline e recursos de pipeline

AWS Data Pipelineexige funções do IAM que determinem as permissões para realizar ações e acessarAWS recursos. A função de pipeline determina as permissões queAWS Data Pipeline tem, e uma função de recurso determina as permissões que os aplicativos executados nos recursos do pipeline, como instâncias do EC2, têm. É necessário especificar essas funções ao criar um pipeline. Mesmo que você não especifique uma função personalizada e use as funções`DataPipelineDefaultRole` padrão`DataPipelineDefaultResourceRole`, você deve primeiro criar as funções e anexar políticas de permissões. Para obter mais informações, consulte Funções do IAM para o AWS Data Pipeline (p. 67).

Permita que os diretores do IAM (usuários e grupos) realizem as ações necessárias

Para trabalhar com um pipeline, um diretor do IAM (um usuário ou grupo) em sua conta deve ter permissão para realizar AWS Data Pipelineas ações e ações necessárias 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 `aiam:PassRole` ação nas funções padrão usadasAWS 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 de 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 aos diretores do IAM. Para obter mais informações e políticas de permissões de exemplo, consulteExemplo de políticas para o AWS Data Pipeline (p. 65)

Uma declaração de política semelhante ao exemplo a seguir deve ser incluída em uma política anexada a qualquer diretor do IAM que use o pipeline. Essa declaração permite que o diretor do IAM execute a`PassRole` ação nas funções que um pipeline usa. Se você não usar funções padrão,*`MyPipelineRole`* substitua-as*`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 AWSDataPipeline_FullAccessgerenciada ao grupo e, em seguida, adicionar usuários ao grupo. Você pode usar esse procedimento para qualquer política embutida

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

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Grupos, Criar novo grupo.
3. Insira um nome de grupo, por exemplo**`DataPipelineDevelopers`**, e escolha Próxima etapa.
4. Digite**`AWSDataPipeline_FullAccess`** Filtro e, em seguida, selecione-o na lista.
5. Selecione Next Step (Próxima etapa) e, em seguida, Create Group (Criar grupo).
6. Para adicionar 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 na lista e escolha Adicionar usuários ao grupo.

Conceder acesso programático

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

Para conceder acesso programático aos usuários, escolha 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 a AWS CLI,	Siga as instruções da interface que deseja utilizar.

Qual usuário precisa de acesso programático?	Para	Por
	os SDKs da AWS ou as APIs da AWS.	• Para a AWS CLI, consulte Configuração da AWS CLI para usar o AWS IAM Identity Center (successor to AWS Single Sign-On) no Guia do usuário da AWS Command Line Interface. • Para os SDKs da AWS, ferramentas e APIs da AWS, consulte Autenticação do Centro de Identidade do IAM no Guia de referência de ferramentas e SDKs da AWS.
IAM	Use credenciais temporárias para assinar solicitações programáticas para a AWS CLI, os SDKs da AWS ou as APIs da AWS.	Siga as instruções em Como usar credenciais temporárias com recursos da AWS no Guia do usuário do IAM.
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitações programáticas para a AWS CLI, os SDKs da AWS ou as APIs da AWS.	Siga as instruções da interface que deseja utilizar. • Para a AWS CLI, consulte Autenticação usando as credenciais de usuário do IAM no Guia do usuário da AWS Command Line Interface. • Para as ferramentas e SDKs da AWS, consulte Autenticação usando as credenciais de longo prazo no Guia de referência de ferramentas e SDKs da AWS. • Para as APIs da AWS, consulte Gerenciamento de chaves de acesso de 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. Esse serviço facilita o design de atividades extract-transform-load (ETL) usando dados estruturados e não estruturados, tanto no local quanto na nuvem, com base em 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 (p. 14) que definem o trabalho a ser executado e nós de dados (p. 14) que definem a localização e o tipo dos 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. Esse pipeline é executado a cada 15 minutos por uma hora e grava a saída no Amazon S3 em cada iteração.

Pré-requisitos

Antes de começar, conclua as tarefas em Configuração do AWS Data Pipeline (p. 19).

Objetos de pipeline

O pipeline usa os seguintes objetos:

ShellCommandActivity (p. 190)

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

S3DataNode (p. 130) (entrada)

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

S3DataNode (p. 130) (saída)

O bucket do S3 para saída.

Ec2Resource (p. 201)

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 (p. 265)

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

Tarefas

- Criar o pipeline (p. 24)
- Monitorar o pipeline em execução (p. 24)
- Visualizar a saída (p. 25)
- Excluir o pipeline (p. 25)

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 AWS Data Pipeline console 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 depende de você ter criado um funil na região atual.

   a. Se você não criou um pipeline nessa região, o console exibirá uma tela introdutória. Selecione Get started now.

   b. Se você já criou um funil 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 funil.
5. (Opcional) Em Descrição, insira uma descrição para seu funil.
6. Em Fonte, selecione Criar usando um modelo e, em seguida, selecione o seguinte modelo: Introdução ao uso 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 ativado. Escolha o ícone de pasta na localização do S3 para registros, selecione um de seus buckets ou pastas e escolha Selecionar.

   Se preferir, em vez disso, você pode desativar o registro.
10. Em Segurança/Acesso, deixe as funções do IAM definidas como Padrão.
11. Clique em Activate.

    Se preferir, você pode escolher Editar no Architect para modificar esse pipeline. Por exemplo, você pode adicionar condições prévias.

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 (p. 289).

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 funil. A exclusão do funil exclui a definição do pipeline e todos os objetos associados.

Para excluir seu funil

1. Na página Listar pipelines, selecione seu funil.
2. Clique em Ações e escolha Excluir.
3. Quando a confirmação for solicitada, escolha Delete (Excluir).

Se você tiver concluído a saída deste 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. AWS As seções a seguir apresentam os conceitos fundamentais do AWS Data Pipeline e mostram como trabalhar com pipelines.

> Important
>
> Antes de começar, consulte Configuração do AWS Data Pipeline (p. 19).

Índice

Criação de um pipeline

O AWS Data Pipeline oferece várias maneiras de criar pipelines:

- Use o AWS Command Line Interface (CLI) com um modelo fornecido para sua conveniência. Para obter mais informações, consulte Crie um pipeline a partir de modelos do Data Pipeline usando a CLI (p. 26).
- 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 (p. 56).

Crie um pipeline a partir de modelos do Data Pipeline usando a CLI

O Data Pipeline fornece várias definições de pipeline pré-configuradas, 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 alcançar 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 JSON e criar um pipeline usando a CLI

1. Faça o download do modelo usando a `aws s3 cp` CLI 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 alterações no modelo baixado conforme necessário. Por exemplo, para usar a versão mais recente da versão do EMR, altere o `releaseLabel` campo no `EmrClusterForBackup` objeto, altere os tipos de instância principal e principal e altere os valores padrão dos parâmetros no modelo.
3. Crie um pipeline usando a `create-pipeline` CLI. Por exemplo:

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

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

Para obter mais informações sobre os modelos, consulte Escolher um modelo (p. 27).

Escolher um modelo

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

Modelos

Conceitos básicos do uso do ShellCommandActivity

O ShellCommandActivity modelo Getting Started using 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 data e hora em cada execução programada do pipeline.

O modelo usa os seguintes objetos de pipeline:

- ShellCommandActivity
- S3 InputNode
- S3 OutputNode
- Ec2Resource

Execute o comando AWS CLI

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

Exportar tabela do DynamoDB para o S3

O modelo Exportar tabela do DynamoDB para o S3 programa um cluster do Amazon EMR para 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 da taxa de transferência 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 umHiveActivity, mas agora usa nativoMapReduce.

O modelo usa os seguintes objetos de pipeline:

- EmrActivity (p. 146)
- EmrCluster (p. 208)
- DynamoDBDataNode (p. 115)
- S3DataNode (p. 130)

Importar dados de backup do DynamoDB do S3

O modelo Importar dados de backup do DynamoDB do S3 programa um cluster do Amazon EMR para carregar um backup do DynamoDB criado anteriormente no Amazon S3 em uma tabela do DynamoDB. Os itens existentes na tabela do DynamoDB são atualizados com os dos dados de backup e novos itens são adicionados à tabela. Esse modelo usa um cluster do Amazon EMR, que é dimensionado proporcionalmente ao valor da taxa de transferência 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 umHiveActivity, mas agora usa nativoMapReduce.

O modelo usa os seguintes objetos de pipeline:

- EmrActivity (p. 146)
- EmrCluster (p. 208)
- DynamoDBDataNode (p. 115)
- S3DataNode (p. 130)
- S3PrefixNotEmpty (p. 242)

Execute o 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 (p. 146)

- EmrCluster (p. 208)

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

A cópia completa da tabela MySQL do RDS para o modelo S3 copia uma tabela MySQL inteira 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 data e hora na localização especificada do Amazon S3.

O modelo usa os seguintes objetos de pipeline:

- CopyActivity (p. 140)
- Ec2Resource (p. 201)
- SqlDataNode (p. 135)
- S3DataNode (p. 130)

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

A cópia incremental da tabela MySQL do RDS para o modelo S3 faz uma cópia incremental dos dados de uma tabela MySQL do Amazon RDS e armazena a saída em um 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. O tipo de agendamento é uma série temporal; portanto, se uma cópia foi agendada para uma determinada hora, AWS Data Pipeline copia as linhas da tabela que têm um carimbo de data/hora da última modificação que esteja dentro da hora. As exclusões físicas feitas na tabela não são copiadas. A saída é gravada em uma subpasta com data e hora abaixo do local do Amazon S3 em cada execução programada.

O modelo usa os seguintes objetos de pipeline:

- CopyActivity (p. 140)
- Ec2Resource (p. 201)
- SqlDataNode (p. 135)
- S3DataNode (p. 130)

Carregue dados do S3 na tabela MySQL do Amazon RDS

O modelo Carregar dados do S3 na tabela MySQL do RDS programa uma instância do Amazon EC2 para copiar o arquivo CSV do caminho de 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 as 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 (p. 140)
- Ec2Resource (p. 201)
- SqlDataNode (p. 135)
- S3DataNode (p. 130)

Modelos do Amazon RDS para o Amazon Redshift

Os dois modelos a seguir copiam tabelas do Amazon RDS MySQL para o Amazon Redshift usando um script de tradução, 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 pular uma coluna que está presente em uma tabela MySQL do Amazon RDS ao fazer uma cópia para o Amazon Redshift.
- (Opcional) Você pode fornecer um mapeamento do tipo de dados de coluna do Amazon RDS MySQL para o 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 `0verwrite_Existing` Amazon Redshift estiver sendo usado:

- Se uma chave de distribuição não for fornecida, uma chave primária na tabela MySQL do Amazon RDS será usada.
- 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 chave primária na tabela do Amazon Redshift.
- Se uma chave de distribuição não for fornecida e não houver uma chave primária na tabela MySQL do Amazon RDS, a operação de cópia falhará.

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

- Amazon Redshift cluster (Cluster do Amazon Redshift)
- CÓPIA DO Amazon Redshift
- Estilos de distribuição e exemplos DISTKEY
- Chaves de classificação

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

Traduções de tipos de dados entre o MySQL e o 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. Amazon Redshift: INT2. Número inteiro de dois bytes assinado
TINYINT UNSIGNED, TINYINT (size) UNSIGNED	SMALLINT	MySQL: de 0 a 255 UNSIGNED. O número máximo de dígitos pode ser especificado entre parênteses. Amazon Redshift: INT2. Número inteiro de dois bytes assinado

Tipo de dados MySQL	Tipo de dados do Amazon Redshift	Observações
SMALLINT, SMALLINT(size)	SMALLINT	MySQL: de -32768 a 32767 normal. O número máximo de dígitos pode ser especificado entre parênteses. Amazon Redshift: INT2. Número inteiro de dois bytes assinado
SMALLINT UNSIGNED, SMALLINT(size) UNSIGNED,	INTEGER	MySQL: de 0 a 65535 UNSIGNED*. O número máximo de dígitos pode ser especificado entre parênteses Amazon Redshift: INT4. Número inteiro de quatro bytes assinado
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
BIGINT UNSIGNED BIGINT(size) UNSIGNED	VARCHAR(20*4)	MySQL: de 0 a 18446744073709551615 Amazon Redshift: Sem equivalente nativo, portanto, use uma matriz de caracteres.

Tipo de dados MySQL	Tipo de dados do Amazon Redshift	Observações
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
DECIMAL(size,d)	DECIMAL(size,d)	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. Amazon Redshift: Sem equivalente nativo.
CHAR(size)	VARCHAR(size*4)	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. Lado direito preenchido com espaços. Amazon Redshift: o tipo de dados CHAR não suporta caracteres de vários bytes, então VARCHAR é usado. O número máximo de bytes por caractere é 4 de acordo com RFC3629, o que limita a tabela de caracteres U+10FFFF.

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 diferente 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
TINYBLOB	VARCHAR (255)	0 a 255 bytes
BLOB	VARCHAR(máximo)	0 a 65,535 bytes
MEDIUMBLOB	VARCHAR(máximo)	0 a 16,777,215 bytes
LONGBLOB	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"

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

A cópia completa da tabela MySQL do Amazon RDS para o modelo do Amazon Redshift copia toda a tabela MySQL do Amazon RDS em uma tabela do Amazon Redshift por meio da preparação de dados em uma pasta do Amazon S3. A pasta de teste do Amazon S3 deve estar na mesma região do cluster do Amazon Redshift. Uma tabela do Amazon Redshift é criada com o mesmo esquema da tabela MySQL do Amazon RDS de origem, caso ela ainda não exista. Forneça qualquer substituição de tipo de dados de coluna do Amazon RDS MySQL para o Amazon Redshift que você gostaria de aplicar durante a criação da tabela do Amazon Redshift.

O modelo usa os seguintes objetos de pipeline:

- CopyActivity (p. 140)
- RedshiftCopyActivity (p. 181)
- S3DataNode (p. 130)
- SqlDataNode (p. 135)
- RedshiftDataNode (p. 125)
- RedshiftDatabase (p. 252)

Cópia incremental de uma tabela MySQL do Amazon RDS para o Amazon Redshift

A cópia incremental da tabela MySQL do Amazon RDS para o modelo do Amazon Redshift copia dados de uma tabela MySQL do Amazon RDS para uma tabela do Amazon Redshift ao armazenar dados em uma pasta do Amazon S3.

A pasta de teste do Amazon S3 deve estar na mesma região do cluster do Amazon Redshift.

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

Esse modelo copia as alterações feitas na tabela MySQL do Amazon RDS entre intervalos programados, começando na hora de início programada. As exclusões físicas 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.

Quando você usa o modelo padrão para criar pipelines para cópia incremental do Amazon RDS, uma atividade com o nome `RDSToS3CopyActivity` padrão é criada. Você pode renomeá-la.

O modelo usa os seguintes objetos de pipeline:

- CopyActivity (p. 140)
- RedshiftCopyActivity (p. 181)
- S3DataNode (p. 130)
- SqlDataNode (p. 135)
- RedshiftDataNode (p. 125)
- RedshiftDatabase (p. 252)

Carregue dados do Amazon S3 para o Amazon Redshift

O modelo Carregar dados do S3 para o Redshift copia dados de uma pasta do Amazon S3 em 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 do Amazon Redshift`COPY`. A tabela do Amazon Redshift deve ter o mesmo esquema dos dados no Amazon S3. Para obter `COPY` opções, consulte COPY no Guia do desenvolvedor de banco de dados do Amazon Redshift.

O modelo usa os seguintes objetos de pipeline:

- CopyActivity (p. 140)
- RedshiftCopyActivity (p. 181)
- S3DataNode (p. 130)
- RedshiftDataNode (p. 125)
- RedshiftDatabase (p. 252)
- Ec2Resource (p. 201)

Criação de 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

Adicione MyVariables à definição do pipeline

Ao criar o arquivo de definição de pipeline, especifique variáveis usando a seguinte sintaxe: #{my*Variable*}. É necessário que a variável seja prefixada por `my`. *Por exemplo, o arquivo de definição de pipeline a seguirpipeline-definition.json, inclui as seguintes variáveis: myShellCmd, MyS3 e MyS3 InputLoc. OutputLoc*

> **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 parâmetros

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 parâmetros para as *myShellCmd*OutputLocvariáveis MyS3 e *MyS3 InputLoc da definição de 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 parâmetros

Atributo	Type	Descrição
`id`	String	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	String	Uma descrição do parâmetro.
type	Cadeia de caracteres, número inteiro, duplo 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	Booliano	Indica se o parâmetro é opcional ou obrigatório. O padrão é `false`.
allowedValues	Lista de strings	Enumera todos os valores permitidos para o parâmetro.
padrão	String	O valor padrão do parâmetro. Se você especificar um valor para esse parâmetro usando valores de parâmetro, ele substituirá o valor padrão.
isArray	Booliano	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 OutputLoc variável *MyS3* da definição de pipeline de exemplo acima.

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

Enviando a definição do pipeline

Ao enviar a definição de pipeline, você pode especificar parâmetros, objetos de parâmetro e valores de parâmetro. Por exemplo, você pode usar o put-pipeline-definitionAWS CLIcomando da seguinte forma:

```
$ 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 seus pipelines usando 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 que pudesse 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 resultado de uma falha em cascata de uma de suas dependências, mas o componente provavelmente não foi a fonte original da falha.

`DEACTIVATING`

O gasoduto está sendo desativado.

`FAILED`

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

`FINISHED`

O componente concluiu o trabalho atribuído.

`INACTIVE`

O oleoduto foi desativado.

`PAUSED`

O componente foi pausado e não está funcionando no momento.

`PENDING`

O pipeline está pronto para ser ativado pela primeira vez.

`RUNNING`

O recurso está em execução e pronto para receber trabalho.

`SCHEDULED`

O recurso está programado para ser executado.

`SHUTTING_DOWN`

O recurso está sendo encerrado após concluir seu trabalho com êxito.

`SKIPPED`

O componente ignorou os intervalos de execução depois que o pipeline foi ativado usando um carimbo de data/hora posterior ao cronograma atual.

`TIMEDOUT`

O recurso ultrapassou o `terminateAfter` limite e foi interrompido. AWS Data Pipeline Depois que o recurso atingir esse status, AWS Data Pipeline ignora os `retryTimeout` valores `actionOnResourceFailureretryDelay`, e desse recurso. Esse status se aplica somente aos recursos.

`VALIDATING`

A definição do pipeline está sendo validada por. AWS Data Pipeline

`WAITING_FOR_RUNNER`

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

`WAITING_ON_DEPENDENCIES`

O componente está verificando se suas pré-condições padrão e configuradas 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 precondições 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 precondições 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 de linha de comando (CLI) para visualizar sua definição de pipeline. A CLI imprime um arquivo de definição de pipeline, no formato JSON. Para obter informações sobre a sintaxe e o uso de arquivos de definição de pipeline, consulte Sintaxe do arquivo de definição de pipeline (p. 54).

Ao usar a CLI, é uma boa ideia recuperar a definição do pipeline antes de enviar as modificações, pois é possível que outro usuário ou processo tenha alterado a definição do pipeline depois da última vez que você trabalhou com ele. 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.

Ao usar a CLI, você pode obter duas versões diferentes do seu 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 completa do pipeline, use o get-pipeline-definitioncomando. 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-00627471SOVYZEXAMPLE
```

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-00627471SOVYZEXAMPLE
```

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 (p. 287). Para obter mais informações sobre a solução de problemas de execuções de instâncias com falha ou incompletas do pipeline, consulte Resolver problemas comuns (p. 289).

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

O registro em nível de pipeline é suportado na criação do pipeline, especificando uma localização do Amazon S3 no console ou com um `pipelineLogUri` no objeto padrão no 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 (incluindo TaskRunner registros desses recursos) seguem a estrutura de registro do pipeline acima.

Para visualizar os registros de um determinado pipeline, execute:

1. Recupere o `ObjectId` chamando `query-objects` para obter o ID exato 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 a determinada. `pipeline-id` Você pode usar o token para realizar todas as tentativas até encontrar o objeto esperado. Por exemplo, um retornado ObjectId seria semelhante a:`@TableBackupActivity_2023-05-020T18:05:18_Attempt=1`.
2. Usando oObjectId, recupere o local do registro 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 que falhou

Para receber a mensagem de erro, primeiro obtenha o ObjectId usoquery-objects.

Depois de recuperar a falhaObjectId, use a `describe-objects` CLI 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 ou executar novamente ou marcar como concluído um objeto

Use a `set-status` CLI 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 `query-objects` CLI. Por exemplo:

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

Use a `set-status` CLI 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

Enquanto o pipeline estiver no `PENDING` estado e não estiver ativado, você não poderá fazer nenhuma alteração 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, baixe uma cópia da definição atual do pipeline usando o get-pipeline-definitioncomando. 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-00627471SOVYZEXAMPLE
```

Salve a definição do pipeline em um arquivo e a edite conforme necessário. Atualize sua definição de pipeline usando o put-pipeline-definitioncomando. O exemplo a seguir faz upload do arquivo de definição de pipeline atualizado.

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --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-00627471SOVYZEXAMPLE
```

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-00627471SOVYZEXAMPLE --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 AWS CLI:

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

A aplicação de uma tag em um pipeline também propaga as tags para seus recursos subjacentes (por exemplo, clusters do Amazon EMR e 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-00627471SOVYZEXAMPLE --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-00627471SOVYZEXAMPLE --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-00627471SOVYZEXAMPLE
```

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-00627471SOVYZEXAMPLE --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-00627471SOVYZEXAMPLE
```

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-00627471SOVYZEXAMPLE --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-00627471SOVYZEXAMPLE
```

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 escalonados no recurso Amazon EMR ou Amazon EC2 estão disponíveis usando variáveis especiais nos comandos do 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 (p. 190).

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 precondição 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_HHmmss')}/
items"
  }
},
{
  "id": "MyOutputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://my_bucket/destination/#{format(@scheduledStartTime,'YYYY-MM-
dd_HHmmss')}"
  }
},
...
```

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. Não há armazenamento de dados disponível para`DynamoDBDataNode`, portanto, primeiro você deve criar manualmente a tabela no script do hive, usando o nome da variável `#{input.tableName}` para se referir à tabela do DynamoDB. Uma nomenclatura semelhante se aplica se a tabela do DynamoDB for a saída, exceto que você usa 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 (p. 106).

```
{
  "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 exemplo de arquivo JSON do pipeline a seguir mostra como executar um `EmrCluster` recurso na região da Europa (Irlanda), supondo que exista 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 nas quais é AWS Data Pipeline possível orquestrar fluxos de trabalho e lançar recursos do Amazon EMR ou do Amazon EC2. AWS Data Pipelinepode não ser suportado nessas regiões. 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 (Norte 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
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 condições prévias

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`.

Executando novamente objetos com 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 de 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 (p. 103).

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 (p. 114).

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 (p. 103). Para obter mais informações sobre funções que você pode usar para avaliar valores de campo, consulte Avaliação de expressões (p. 106).

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 (p. 114).

```
{
  "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 nomeado `myCustomField` e `my_customFieldReference` adicionado a um DataNode objeto do S3:

```
{
  "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, consulteConfiguração do AWS Data Pipeline (p. 19)

A maneira mais fácil de escrever aplicativos que interagem com oAWS Data PipelineOu 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 (p. 56).

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-00627471SOVYZEXAMPLE",
 "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": "12345ABCDEFG"}
```

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-00627471SOVYZEXAMPLE"}
```

Segurança em AWS Data Pipeline

A segurança da 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 da 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

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 obter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para obter 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 (successor to AWS Single Sign-On) 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 obter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.
- AWS Data Pipelinesuporta 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 serviços 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, recomendemos 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 doAWS Data Pipeline e do IAMAWS Identity and Access Management (IAM) para permitir queAWS Data Pipeline outros usuários do acessem seusAWS Data Pipeline recursos do 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:

- Controle 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

AWS Data Pipelineestá integrado aoAWS Identity and Access Management (IAM), que oferece uma ampla variedade de recursos:

- Criar usuários e grupos na contaConta da AWS da
- Compartilhe facilmente seusAWS recursos entre os usuários do seuConta da AWS.
- Atribua credenciais de segurança exclusivas a cada usuário.
- Controle o acesso de cada usuário a serviços e recursos.
- Obtenha uma única fatura para todos os usuários do seuConta da AWS.

Ao usar o IAM com oAWS 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 políticas do IAM com base em tags de pipeline e grupos de trabalho para compartilhar seus pipelines com outros usuários e controlar o nível de acesso que eles têm.

Índice

Políticas do IAM para o AWS Data Pipeline

Por padrão, as entidades do IAM não têm permissão para criar ou modificar recursos da AWS. Para permitir que as entidades do IAM criem ou modifiquem recursos e realizem tarefas, você deve criar 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 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

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 padrão, 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 uma lista de ações paraAWS Data Pipeline, consulte Ações na referênciaAWS Data Pipeline da API.

- 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 concedam acesso ao usuário que criou o pipeline. Para obter um exemplo, consulte Conceder acesso total ao proprietário do pipeline (p. 66).
  - `datapipeline:Tag`— Para conceder acesso com base na marcação do pipeline. Para obter mais informações, consulte Controlar acesso aos pipelines usando tags (p. 63).
  - `datapipeline:workerGroup`— Conceder acesso com base no nome do grupo de trabalhadores. Para obter mais informações, consulte Controlar acesso aos pipelines usando grupos de operadores (p. 64).

Controlar acesso aos pipelines usando tags

Você pode criar políticas do IAM que fazem referência às tags do IAM. 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 pipelines no ambiente de produção, o gerente concede acesso de leitura/gravação aos usuários do grupo IAM de produção, mas concede acesso somente de leitura aos usuários do grupo IAM do desenvolvedor. Para pipelines no ambiente de desenvolvimento, o gerente concede acesso de leitura/gravação aos grupos de produção e desenvolvedores do IAM.

Para alcançar esse cenário, o gerente marca os pipelines de produção com a tag “environment=production” e anexa a seguinte política ao grupo de desenvolvedores do IAM. 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 atribui a seguinte política ao grupo 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 (p. 65) e Conceder acesso total aos usuários com base em uma tag (p. 66).

Controlar acesso aos pipelines usando grupos de operadores

Você pode criar políticas do IAM que criem nomes de grupos de trabalhadores de referência.

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 quer garantir que os usuários do grupo IAM de produção possam criar pipelines que enviem tarefas para recursos de produção e que os usuários do grupo IAM de desenvolvimento possam criar pipelines que enviem tarefas para recursos de pré-produção e de desenvolvedores.

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 seguinte política ao grupo de desenvolvedores do IAM para bloquear o acesso aos recursos do “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 atribui a seguinte política ao grupo de produção do IAM para conceder acesso aos recursos do “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

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 ListPipelines da API não oferece suporte à autorização baseada em 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 daAWS Data Pipeline API ListPipelines, com exceção de, mas somente com pipelines que tenham 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`PassRole` permissão baseada em identidade (IAM), consulte a postagem do blog Concedendo permissão para iniciar instâncias do EC2 com funções do IAM (PassRolepermissão).

```
{
 "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 PipelineusaAWS Identity and Access Management funções. As políticas de permissões associadas às funções do IAM determinam quais açõesAWS Data Pipeline e seus aplicativos podem executar e quaisAWS recursos eles podem acessar. Para obter mais informações, consulte Funções do IAM no Guia do usuário do IAM.

AWS Data Pipelinerequer duas funções do IAM:

- A função do pipeline controla oAWS Data Pipeline acesso aos recursos da AWS. Nas definições de objetos do pipeline, `orole` campo especifica essa função.
- A função 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 aosAWS recursos. Nas definições de objetos do pipeline, `oresourceRole` campo especifica essa função.

  Important

  Se você criou um pipeline antes de 3 de outubro de 2022 usando oAWS Data Pipeline console com funções padrão,AWS Data Pipeline criou o`DataPipelineDefaultRole` para você e anexou a política`AWSDataPipelineRole` gerenciada à função. A partir de 3 de outubro de 2022, a política`AWSDataPipelineRole` gerenciada 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 o`DataPipelineDefaultRole` está associado ao pipeline e se ele`AWSDataPipelineRole` está vinculado a essa função. Em caso afirmativo, 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 políticas anexadas a essa função conforme necessário. Como alternativa, você pode atualizar um pipeline para usar uma função criada com políticas de permissões diferentes.

Exemplo de políticas de permissões paraAWS Data Pipeline funções

Cada função tem uma ou mais políticas de permissões anexadas que determinam osAWS recursos 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 queAWS Data Pipeline exigem a execução de um pipeline com recursos do Amazon EC2 e do Amazon EMR. Ele também fornece permissões para acessar outrosAWS recursos, como o Amazon Simple Storage Service e o Amazon Simple Notification Service, que muitos pipelines exijam. Se os objetos definidos em um pipeline não precisarem dos recursos de umAWS serviço, é altamente recomendável que você remova as permissões para acessar esse serviço. Por exemplo, se seu funil não definirDynamoDBDataNode (p. 115) ou usar aSnsAlarm (p. 263) ação, recomendamos que você remova as instruções de permissão dessas ações.

- Substitua *`111122223333`* pelo ID de sua conta da AWS.
- *`NameOfDataPipelineRole`*Substitua pelo nome da função do pipeline (a função à qual essa política está anexada).
- *`NameOfDataPipelineResourceRole`*Substitua pelo nome da função da instância do EC2.
- *`us-west-1`*Substitua 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 do`AmazonEC2RoleforDataPipelineRole` é exibido abaixo. Essa é a política gerenciada associada à função de recurso padrão paraAWS Data Pipeline,`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 açõesAWS de serviço que não são necessárias.

É mostrada a versão 3 da política, que é a versão mais recente no momento em que este artigo foi escrito. 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": ["*"]
    }]
}
```

CriaçãoAWS Data Pipeline e edição de funções do IAM para permissões de função

Use os procedimentos a seguir para criar funções paraAWS Data Pipeline usar o console do IAM. O processo consiste em duas etapas. Primeiro, você cria uma política de permissões para anexar à função. Em seguida, você cria a função e anexa política. Depois de criar uma função, você pode alterar as permissões da função anexando e desanexando políticas de permissões.

> **Note**
>
> Quando você cria funções paraAWS Data Pipeline usar o console conforme descrito abaixo, o IAM cria e anexa as políticas de confiança apropriadas que a função exige.

Para criar uma política de permissões para usar com uma função paraAWS 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 políticaExemplo de política de permissões da função de pipeline (p. 68), editando-o conforme apropriado para seus requisitos de segurança. Como alternativa, se você estiver criando uma função de instância EC2 personalizada, faça o mesmo com o exemplo emPolítica gerenciada padrão para a função de instância do EC2 (p. 71).
5. Escolha Review policy (Revisar política).
6. Insira um nome para a política, por exemplo`MyDataPipelineRolePolicy`, e uma Descrição opcional e escolha Criar política.
7. Anote o nome da política. Você precisa dela 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, escolha Create role (Funções) e Create role (Criar função).
3. Em Escolher um caso de uso, escolha Data Pipeline.
4. Em Select your use case (Selecionar caso de uso), execute uma das seguintes ações:
   - Escolha`Data Pipeline` criar uma função de pipeline.
   - Escolha`EC2 Role for Data Pipeline` criar uma função de recurso.
5. Escolha Next: Permissions (Próximo: permissões).
6. Se a política padrão paraAWS Data Pipeline estiver listada, siga as etapas a seguir para criar a função e edite-a de acordo com as instruções no próximo procedimento. Caso contrário, insira o nome da política que você criou no procedimento acima e selecione-a na lista.
7. Escolha Avançar: Tags, insira as tags a serem adicionadas à função e escolha Avançar: Revisão.
8. Insira um nome para a função, por exemplo`MyDataPipelineRole`, e uma Descrição opcional e escolha Criar função.

Para anexar ou desanexar uma política de permissões para uma função do IAM paraAWS Data Pipeline

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles
3. Na caixa de pesquisa, comece a digitar o nome da função que você deseja editar — por exemplo, DataPipelineDefaultRoleou MyDataPipelineRole— e escolha o nome da função na lista.
4. Na guia Permissões, faça o seguinte:
   - Para desanexar uma política de permissões, em Políticas de permissões, escolha o botão remover na extremidade direita da entrada da política. Escolha Desanexar quando solicitado a confirmar.
   - Para anexar uma política que você criou anteriormente, escolha Anexar políticas. Na caixa de pesquisa, comece a digitar o nome da política que você quer editar, selecione-a na lista e escolha Anexar política.

Alterando as funções de um pipelines existente

Se você quiser atribuir uma função de pipeline ou função de recurso diferente a um pipeline, você pode usar o editor de arquiteto noAWS Data Pipeline console.

Para editar as funções atribuídas a um pipeline usando o console

1. Abra oAWS Data Pipeline console em https://console.aws.amazon.com/datapipeline/.
2. Selecione o pipeline na lista e escolha Ações, Editar.

3. No painel direito do editor de arquitetura, escolha Outros.
4. Nas listas Função e Função do Recurso, escolha as funçõesAWS Data Pipeline que você deseja atribuir e escolha Salvar.

Registro em log e monitoramento no AWS Data Pipeline

AWS Data Pipelineé integrado aoAWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou umAWS serviço da noAWS Data Pipeline. CloudTrail captura todas as chamadas de API doAWS 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 CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para oAWS Data Pipeline. Se não configurar uma trilha, você ainda poderá visualizar os eventos mais recentes no CloudTrail console do em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação que foi feita aoAWS Data Pipeline, o endereço IP da solicitação, quem fez a solicitação, quando ela foi feita e outros detalhes.

Para saber mais sobre isso CloudTrail, consulte o GuiaAWS CloudTrail do usuário.

AWS Data PipelineInformações em CloudTrail

CloudTrail O está habilitado na suaAWS conta da ao criá-la. Quando ocorre uma atividade noAWS Data Pipeline, essa atividade é registrada em um CloudTrail evento junto com outros eventos deAWS serviços da em Event history (Histórico de eventos). Você pode visualizar, pesquisar e baixar eventos recentes em sua conta da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de CloudTrail eventos do.

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos do AWS Data Pipeline, crie uma trilha. Uma trilha permite CloudTrail a entrega de 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 produtosAWS da para analisar mais profundamente e agir sobre os dados de eventos coletados nos CloudTrail logs do. Para obter mais informações, consulte as informações a seguir.

- Visão geral da criação de uma trilha
- CloudTrail Serviços e integrações compatíveis
- Configuração de notificações do Amazon SNS para o CloudTrail
- Receber arquivos de CloudTrail log de várias regiões e Receber arquivos de CloudTrail log de várias contas

Todas asAWS Data Pipeline ações são registradas CloudTrail e documentadas no capítulo Ações de referência da API do AWS Data Pipeline. Por exemplo, chamadas da CreatePipelineação geram entradas nos arquivos de CloudTrail log do.

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 função da raiz ou 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 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. CloudTrail arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. CloudTrail Os arquivos de log não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

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


```
    {
  "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 grande impacto 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. fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, altas taxas de transferência e redes altamente redundantes. Com as zonas de disponibilidade, você pode 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,AWS Data Pipelineé protegido peloAWSprocedimentos de segurança de rede global da descritos noAmazon Web Services: Visão geral da do processo de segurançawhitepaper.

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 orientam você no step-by-step processo de criação e uso de pipelines com. AWS Data Pipeline

Tutoriais

- Processe dados usando o Amazon EMR com o Hadoop Streaming (p. 76)
- Copie dados CSV entre buckets do Amazon S3 usando AWS Data Pipeline (p. 80)
- Exporte dados do MySQL para o Amazon S3 usando AWS Data Pipeline (p. 86)
- Copie dados para o Amazon Redshift usando AWS Data Pipeline (p. 94)

Processe dados usando o Amazon EMR com o Hadoop Streaming

Você pode usar AWS Data Pipeline para gerenciar seus clusters do Amazon EMR. 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), 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 um cluster simples do Amazon EMR executar um trabalho preexistente de streaming do Hadoop fornecido pelo Amazon EMR e enviar uma notificação do Amazon SNS após a conclusão bem-sucedida da tarefa. Você usa o recurso de cluster do Amazon EMR fornecido pela AWS Data Pipeline para essa tarefa. O aplicativo de amostra é chamado WordCount e também pode ser executado manualmente no console do Amazon EMR. Observe que os clusters gerados por AWS Data Pipeline em seu nome são exibidos no console do Amazon EMR e são cobrados em sua conta da AWS.

Objetos de pipeline

O pipeline usa os seguintes objetos:

EmrActivity (p. 146)

Define o trabalho a ser executado no pipeline (executar um trabalho preexistente do Hadoop Streaming fornecido pelo Amazon EMR).

EmrCluster (p. 208)

O recurso que o AWS Data Pipeline usa para executar essa atividade.

Um cluster é um conjunto de instâncias do Amazon EC2. AWS Data Pipelineinicia o cluster e o encerra após a conclusão da tarefa.

Schedule (p. 265)

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

SnsAlarm (p. 263)

Envia uma notificação do Amazon SNS para o tópico que você especificar após a conclusão bem-sucedida da tarefa.

Índice

Antes de começar

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

- Conclua as tarefas em Configuração do AWS Data Pipeline (p. 19).
- (Opcional) Configure uma VPC para o cluster e um security group para a VPC.
- 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.

Iniciar um cluster usando a linha de comando

Se você executa regularmente um cluster do Amazon EMR para analisar registros da web ou realizar análises de dados científicos, você pode usá-lo AWS Data Pipeline para gerenciar seus clusters do Amazon EMR. Com oAWS 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 explica o lançamento de um cluster que pode ser um modelo para um pipeline simples baseado no Amazon EMR ou como parte de um pipeline mais complexo.

Pré-requisitos

Antes de usar a CLI, é necessário executar as seguintes etapas:

1. Instale e configure uma interface de linha de comando (CLI). Para obter mais informações, consulte Como acessar o AWS Data Pipeline (p. 7).
2. Certifique-se de que as funções do IAM sejam nomeadas DataPipelineDefaultRolee DataPipelineDefaultResourceRoleexistam. O AWS Data Pipeline console cria essas funções para você automaticamente. Se você não usou o AWS Data Pipeline console pelo menos uma vez, deve criar essas funções manualmente. Para obter mais informações, consulte Funções do IAM para o AWS Data Pipeline (p. 67).

Tarefas

Criar o arquivo de definição de pipeline

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

Copie este código em um arquivo de texto e salve-o como `MyEmrPipelineDefinition.json`. Você deve substituir a localização 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 os clusters imediatamente, defina `startDateTime` para uma data de um dia atrás e `endDateTime` para uma data um dia depois.

Em seguida, o AWS Data Pipeline iniciará imediatamente os clusters “em atraso” em uma tentativa de processar o que ele considera um acúmulo de trabalho. Essa alocação significa que você não precisa esperar uma hora para ver o 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 instâncias do Amazon EC2 usadas para executar o cluster. Você pode especificar o tamanho e o número de instâncias do EC2 para serem executadas como o 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 na AMI fornecida pelo Amazon EMR.
- `MyEmrActivity`, que representa a computação a ser processada com o cluster. O Amazon EMR oferece suporte a vários tipos de clusters, incluindo streaming, Cascading e Scripted Hive. O `runsOn` campo se refere novamente aMyCluster, 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 funil 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 de definição do pipeline`*. `.json`

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 comandos da CLI.

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

Para carregar sua definição de pipeline, use o put-pipeline-definitioncomando a seguir.

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

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

Para ativar seu pipeline, use o seguinte comando activate-pipeline.

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

Você pode verificar se seu pipeline aparece na lista de pipelines usando o comando list-pipelines a seguir.

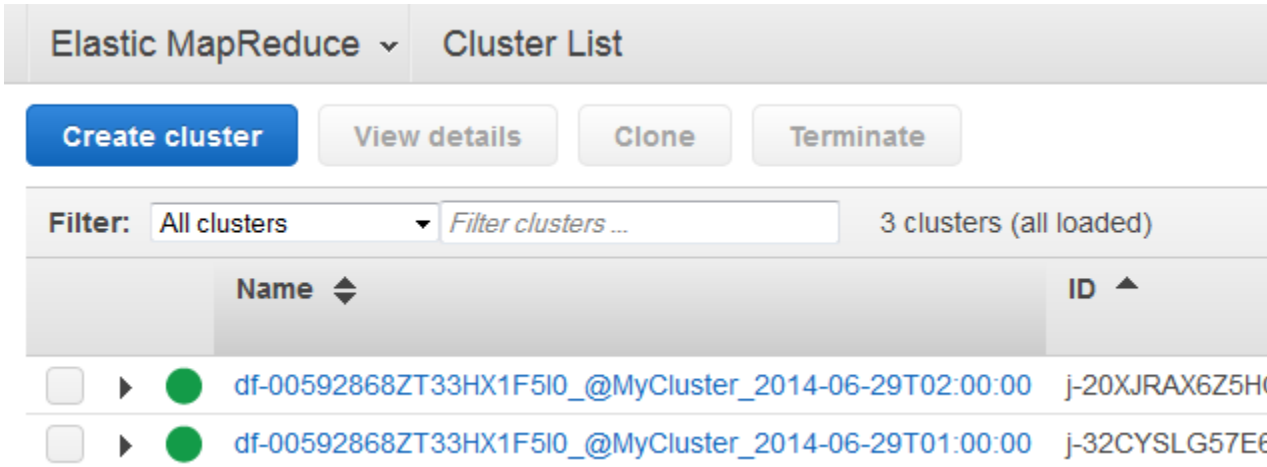
```
aws datapipeline list-pipelines
```

Monitorar as execuções do pipeline

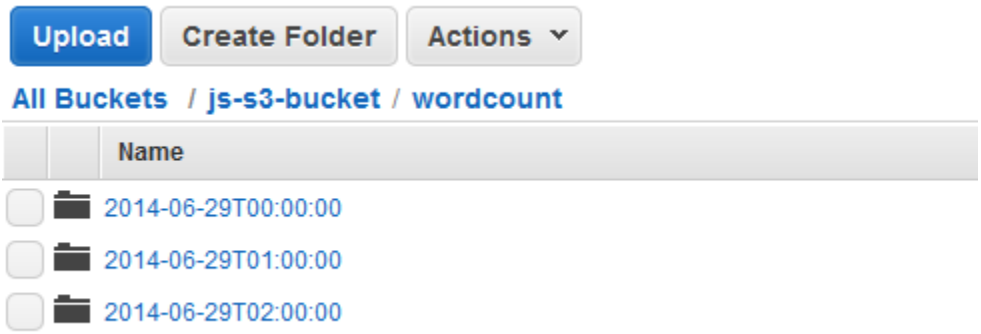
Você pode visualizar clusters lançados AWS Data Pipeline usando o console do Amazon EMR e visualizar a pasta de saída usando o console do Amazon S3.

Para verificar o andamento dos clusters iniciados pelo AWS Data Pipeline

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



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



Copie dados CSV entre buckets do Amazon S3 usando AWS Data Pipeline

Depois de ler O que é o AWS Data Pipeline? (p. 1) 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 mostra o 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 bem-sucedida 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 (p. 140)

A atividade que AWS Data Pipeline executa para esse pipeline (copiar 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 (p. 140).

Schedule (p. 265)

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

Ec2Resource (p. 201)

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

S3DataNode (p. 130)

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

SnsAlarm (p. 263)

A ação AWS Data Pipeline deve ser executada quando as condições especificadas forem atendidas (enviar notificações do Amazon SNS para um tópico após a conclusão bem-sucedida da tarefa).

Índice

Antes de começar

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

- Conclua as tarefas em Configuração do AWS Data Pipeline (p. 19).
- (Opcional) Configure uma VPC para a instância e um security group para a VPC.
- Crie um bucket do Amazon S3 como fonte de dados.

  Para obter mais informações, consulte Criar um bucket no Guia do usuário do Amazon Simple Storage Service.
- Carregue seus dados para o 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 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 função e relações de confiança do IAM, siga as instruções descritas emFunções do IAM para o AWS Data Pipeline (p. 67).

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 uma interface de linha de comando (CLI). Para obter mais informações, consulte Como acessar o AWS Data Pipeline (p. 7).
2. Certifique-se de que as funções do IAM sejam nomeadas DataPipelineDefaultRolee DataPipelineDefaultResourceRoleexistam. O AWS Data Pipeline console cria essas funções para você automaticamente. Se você não usou o AWS Data Pipeline console pelo menos uma vez, deve criar essas funções manualmente. Para obter mais informações, consulte Funções do IAM para o AWS Data Pipeline (p. 67).

Tarefas

Definir um pipeline no formato JSON

Esse exemplo de cenário mostra como usar as definições de pipeline JSON e a AWS Data Pipeline CLI 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"
},
```

Nódulos de dados do Amazon S3

Em seguida, o componente de DataNode pipeline de entrada do S3 define um local para os arquivos de entrada; nesse caso, um local do bucket do Amazon S3. O DataNode componente S3 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).

Type (Tipo)

O tipo de componente do pipeline, que é “S3DataNode” para corresponder ao local em que os dados residem, em um bucket do Amazon S3.

Schedule

Uma referência ao componente de agendamento que criamos nas linhas anteriores do arquivo JSON denominado “”. MySchedule

Caminho

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 de um caminho do Amazon S3 segue uma sintaxe diferente que é apropriada para uma tabela de banco de dados.

Em seguida, o DataNode componente S3 de saída define o local de destino de saída para os dados. Ele segue o mesmo formato do DataNode componente S3 de entrada, exceto o 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.

Type (Tipo)

O tipo de recurso computacional para executar o trabalho. Nesse caso, uma instância do EC2. Há outros tipos de recursos disponíveis, como um EmrCluster tipo.

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, consulte o tópico Tipos de instância do Amazon EC2 em http://aws.amazon.com/ec2/instance-types/.

Função

A função IAM da conta que acessa recursos, como acessar um bucket do Amazon S3 para recuperar dados.

resourceRole

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

Atividades

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.

Type (Tipo)

O tipo de atividade a ser executada, comoMyCopyActivity.

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 funil 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 de definição do pipeline`*`.json`

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 comandos da CLI.

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

Para carregar sua definição de pipeline, use o put-pipeline-definitioncomando a seguir.

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

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

Para ativar seu pipeline, use o seguinte comando activate-pipeline.

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

Você pode verificar se seu pipeline aparece na lista de pipelines usando o comando list-pipelines a seguir.

```
aws datapipeline list-pipelines
```

Exporte dados do MySQL para o Amazon S3 usando AWS Data Pipeline

Este tutorial mostra o 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 vírgula) 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 (p. 140)
- Ec2Resource (p. 201)
- MySqlDataNode (p. 120)
- S3DataNode (p. 130)
- SnsAlarm (p. 263)

Índice

Antes de começar

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

- Conclua as tarefas em Configuração do AWS Data Pipeline (p. 19).
- (Opcional) Configure uma VPC para a instância e um security group para a VPC.
- Crie um bucket do Amazon S3 como 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.

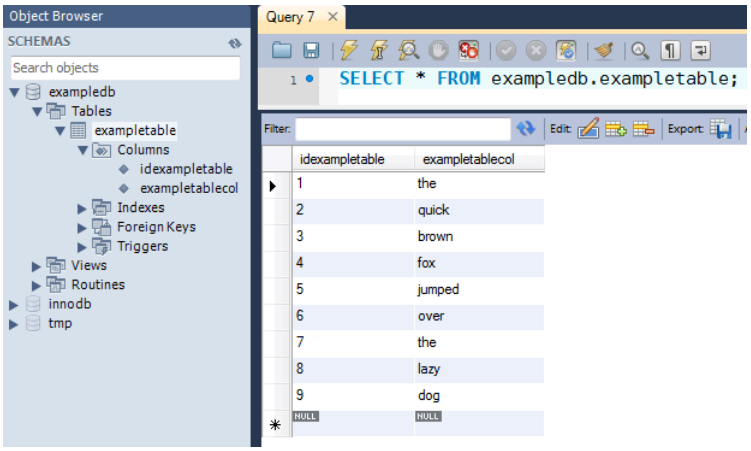
Para obter mais informações, consulte Iniciar uma instância de banco de dados no Guia de conceitos básicos do Amazon RDS. Depois de ter uma instância do Amazon RDS, consulte Criar uma tabela na documentação do MySQL.

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 sua política de função e relações de confiança do IAM, siga as instruções descritas emFunções do IAM para o AWS Data Pipeline (p. 67).

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 uma interface de linha de comando (CLI). Para obter mais informações, consulte Como acessar o AWS Data Pipeline (p. 7).
2. Certifique-se de que as funções do IAM sejam nomeadas DataPipelineDefaultRolee DataPipelineDefaultResourceRoleexistam. O AWS Data Pipeline console cria essas funções para você automaticamente. Se você não usou o AWS Data Pipeline console pelo menos uma vez, deve criar essas funções manualmente. Para obter mais informações, consulte Funções do IAM para o AWS Data Pipeline (p. 67).

3. Configure um bucket do Amazon S3 e uma instância do Amazon RDS. Para obter mais informações, consulte Antes de começar (p. 86).

Tarefas

- Definir um pipeline no formato JSON (p. 88)
- Fazer upload e ativar a definição do pipeline (p. 93)

Definir um pipeline no formato JSON

Esse exemplo de cenário mostra como usar as definições de pipeline JSON e a AWS Data Pipeline CLI para copiar dados (linhas) de uma tabela em um banco de dados MySQL para um arquivo CSV (valores separados por vírgula) em um bucket do Amazon S3 em um intervalo de tempo 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 do MySqlDataNode pipeline de entrada define um local para os dados de entrada; nesse caso, uma instância do Amazon RDS. O MySqlDataNode componente 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.

Name (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 para seu usuário. O campo de senha é precedido pelo caractere especial asterisco. Para obter mais informações, consulte Caracteres especiais (p. 113).

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 CopyActivity objeto se conectar 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.

Type (Tipo)

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

> **Note**
>
> O tipo MySqlDataNode está obsoleto. Embora você ainda possa usarMySqlDataNode, recomendamos o usoSqlDataNode.

Nó de dados do Amazon S3

Em seguida, o componente do pipeline S3Output define um local para o arquivo de saída; nesse caso, um arquivo CSV em um local de bucket do Amazon S3. O DataNode componente S3 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.

Name (Nome)

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

Type (Tipo)

O tipo de objeto do pipeline, que é S3 DataNode para corresponder ao 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.

Name (Nome)

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

Função

A função IAM da conta que acessa recursos, como acessar um bucket do Amazon S3 para recuperar dados.

Type (Tipo)

O tipo de recurso computacional para executar o trabalho. Nesse caso, uma instância do EC2. Há outros tipos de recursos disponíveis, como um EmrCluster tipo.

resourceRole

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

Atividades

A última seção no arquivo JSON é a definição da atividade que representa o trabalho a ser executado. Nesse caso, usamos um CopyActivity componente 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

Name (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 (p. 263) a ser enviado se a atividade for concluída com sucesso

onFail

SnsAlarm (p. 263) a ser enviado se a atividade falhar

Resultado

A localização do arquivo de saída CSV no Amazon S3

Type (Tipo)

O tipo da atividade a ser executada.

Fazer upload e ativar a definição do pipeline

Você deve fazer o upload da definição do funil 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 de definição do pipeline*.`.json`

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 comandos da CLI.

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

Para carregar sua definição de pipeline, use o put-pipeline-definitioncomando a seguir.

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

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

Para ativar seu pipeline, use o seguinte comando activate-pipeline.

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

Você pode verificar se seu pipeline aparece na lista de pipelines usando o comando list-pipelines a seguir.

```
aws datapipeline list-pipelines
```

Copie dados para o Amazon Redshift usando AWS Data Pipeline

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

O Amazon S3 é um serviço web que permite armazenar dados na nuvem. Para obter mais detalhes, consulte o Manual do usuário do Amazon Simple Storage Service.

O Amazon Redshift é um serviço de armazenamento de dados 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

Antes de copiar dados para o Amazon Redshift internoAWS Data Pipeline, certifique-se de:

- Carregue dados do Amazon S3.
- Configure a `COPY` atividade 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 obter `COPY` opções, consulte COPY no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Para ver as etapas para carregar dados do Amazon S3, consulte Carregar dados do Amazon S3 no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por exemplo, o comando SQL a seguir no Amazon Redshift cria uma nova tabela chamada `LISTING` e copia dados de amostra de um bucket disponível ao público no Amazon S3.

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

Para obter detalhes sobre esse exemplo, consulte Carregar dados de amostra 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://awssampledbuswest2/tickit/listings_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region 'us-west-2';
```

Configure o pipeline, crie um grupo de segurança e crie um cluster do Amazon Redshift

Para se preparar para o tutorial

1. Conclua as tarefas em Configuração do AWS Data Pipeline (p. 19).
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 do 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 tiver um cluster Amazon Redshift existente, anote o ID do cluster.

Para criar um novo cluster e carregar dados de amostra, siga as etapas em Introdução ao Amazon Redshift. Para obter mais informações sobre a criação de clusters, consulte Criação de 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.

Copie dados para o Amazon Redshift usando a linha de comando

Este tutorial demonstra como copiar dados do Amazon S3 para o Amazon Redshift. Você criará uma nova tabela no Amazon Redshift e usará AWS Data Pipeline para transferir dados para essa tabela de um bucket público do Amazon S3, que contém exemplos de dados de entrada no formato CSV. Os registros são salvos em um bucket do Amazon S3 que você possui.

O Amazon S3 é um serviço web que permite armazenar dados na nuvem. Para obter mais detalhes, consulte o Manual do usuário do Amazon Simple Storage Service. O Amazon Redshift é um serviço de armazenamento de dados 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 uma interface de linha de comando (CLI). Para obter mais informações, consulte Como acessar o AWS Data Pipeline (p. 7).
2. Certifique-se de que as funções do IAM sejam nomeadas DataPipelineDefaultRolee DataPipelineDefaultResourceRoleexistam. O AWS Data Pipeline console cria essas funções para você automaticamente. Se você não usou o AWS Data Pipeline console pelo menos uma vez, deve criar essas funções manualmente. Para obter mais informações, consulte Funções do IAM para o AWS Data Pipeline (p. 67).
3. Configure o `COPY` comando no Amazon Redshift, pois você precisará ter essas mesmas opções funcionando ao realizar a cópia interna. AWS Data Pipeline Para ter mais informações, consulte Antes de começar: configurar as opções COPY e carregar dados (p. 94).
4. Configure um banco de dados do Amazon Redshift. Para obter mais informações, consulte Configure o pipeline, crie um grupo de segurança e crie um cluster do Amazon Redshift (p. 95).

Tarefas

- Definir um pipeline no formato JSON (p. 96)
- Fazer upload e ativar a definição do pipeline (p. 102)

Definir um pipeline no formato JSON

Esse exemplo de cenário 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 (p. 98)
- Recurso (p. 100)
- Atividades (p. 101)

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 do `S3DataNode` pipeline de entrada define a localização dos dados de entrada no Amazon S3 e o formato dos dados de entrada. Para obter mais informações, consulte S3DataNode (p. 130).

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 `RedshiftDataNode` pipeline de saída define um local para os dados de saída; nesse caso, uma tabela em um banco de dados do Amazon Redshift. Para obter mais informações, consulte RedshiftDataNode (p. 125). 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 Amazon Redshift.

Banco de dados

O componente `RedshiftDatabase` é definido pelos seguintes campos. Para obter mais informações, consulte RedshiftDatabase (p. 252).

```
{
  "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 (p. 201).

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`

A função IAM da conta que acessa recursos, como acessar um bucket do Amazon S3 para recuperar dados.

`logUri`

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

`resourceRole`

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

Atividades

A última seção no arquivo JSON é a definição da atividade que representa o trabalho a ser executado. Nesse caso, usamos um `RedshiftCopyActivity` componente para copiar dados do Amazon S3 para o Amazon Redshift. Para obter mais informações, consulte RedshiftCopyActivity (p. 181).

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 ao 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 destinos do Amazon Redshift.

Fazer upload e ativar a definição do pipeline

Você deve fazer o upload da definição do funil 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 de definição do pipeline`*`.json`

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 comandos da CLI.

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

Para carregar sua definição de pipeline, use o put-pipeline-definitioncomando a seguir.

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

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

Para ativar seu pipeline, use o seguinte comando activate-pipeline.

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

Você pode verificar se seu pipeline aparece na lista de pipelines usando o comando list-pipelines a seguir.

```
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

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"
```

Numeric

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 obter mais informações sobre o `node`, consulte Referenciar campos e objetos (p. 104). Para obter mais informações sobre os tipos de objetos de pipeline, consulte Referência de objeto de pipeline (p. 114).

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"
```

String

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 (p. 104) e Avaliação de expressões (p. 106).

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 (p. 106).

```
#{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 precondição.

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}, será avaliado para["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 (p. 56).

```
{
  "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}`Result:`0` Exemplo: `#{1.0 / 2}`Result:`.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
+	Concatenação. Os valores que não são de string são convertidos primeiro em valores de strings. Exemplo: `#{"hel" + "lo"}` Result: `"hello"`

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
`int day(DateTime myDateTime)`	Obtém o dia do valor DateTime como um número inteiro. Exemplo: `#{day(myDateTime)}` Result: `24`
`int dayOfYear(DateTime myDateTime)`	Obtém o dia do ano de DateTime como um número inteiro. Exemplo: `#{dayOfYear(myDateTime)}` Result: `144`

`DateTime firstOfMonth(DateTime myDateTime)`	Cria um objeto DateTime para o início do mês no DateTime especificado.
	
	Exemplo:
	`#{firstOfMonth(myDateTime)}`
	
	Result:
	`"2011-05-01T17:10:00z"`

`String format(DateTime myDateTime,String format)`	Cria um objeto String que é o resultado da conversão do DateTime especificado usando a string de formato especificada.
	
	Exemplo:
	`#{format(myDateTime,'YYYY-MM-dd HH:mm:ss z')}`
	
	Result:
	`"2011-05-24T17:10:00 UTC"`

`int hour(DateTime myDateTime)`	Obtém a hora do valor DateTime como um número inteiro. Exemplo: `#{hour(myDateTime)}` Result: `17`

Função	Descrição

`DateTime makeDate(int year,int month,int day)`	Cria um objeto DateTime, em UTC, com ano, mês e dia especificados, à meia-noite.
	
	Exemplo:
	`#{makeDate(2011,5,24)}`
	
	Result:
	`"2011-05-24T0:00:00z"`

`DateTime makeDateTime(int year,int month,int day,int hour,int minute)`	Cria um objeto DateTime, em UTC, com ano, mês, dia, hora e minuto especificados.
	
	Exemplo:
	`#{makeDateTime(2011,5,24,14,21)}`
	
	Result:
	`"2011-05-24T14:21:00z"`

`DateTime midnight(DateTime myDateTime)`	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:
	
	Exemplo:
	`#{midnight(myDateTime)}`
	
	Result:
	`"2011-05-25T0:00:00z"`

`DateTime minusDays(DateTime myDateTime,int daysToSub)`	Cria um objeto DateTime que é o resultado da subtração do número de dias especificado a partir do DateTime especificado.
	
	Exemplo:
	`#{minusDays(myDateTime,1)}`
	
	Result:
	`"2011-05-23T17:10:00z"`

`DateTime minusHours(DateTime myDateTime,int hoursToSub)`	Cria um objeto DateTime que é o resultado da subtração do número de horas especificado a partir do DateTime especificado.
	
	Exemplo:
	`#{minusHours(myDateTime,1)}`
	
	Result:
	`"2011-05-24T16:10:00z"`

Função	Descrição

`DateTime minusMinutes(DateTime myDateTime,int minutesToSub)`	Cria um objeto DateTime que é o resultado da subtração do número de minutos especificado a partir do DateTime especificado.
	
	Exemplo:
	`#{minusMinutes(myDateTime,1)}`
	
	Result:
	`"2011-05-24T17:09:00z"`

`DateTime minusMonths(DateTime myDateTime,int monthsToSub)`	Cria um objeto DateTime que é o resultado da subtração do número de meses especificado a partir do DateTime especificado.
	
	Exemplo:
	`#{minusMonths(myDateTime,1)}`
	
	Result:
	`"2011-04-24T17:10:00z"`

`DateTime minusWeeks(DateTime myDateTime,int weeksToSub)`	Cria um objeto DateTime que é o resultado da subtração do número de semanas especificado a partir do DateTime especificado.
	
	Exemplo:
	`#{minusWeeks(myDateTime,1)}`
	
	Result:
	`"2011-05-17T17:10:00z"`

`DateTime minusYears(DateTime myDateTime,int yearsToSub)`	Cria um objeto DateTime que é o resultado da subtração do número de anos especificado a partir do DateTime especificado.
	
	Exemplo:
	`#{minusYears(myDateTime,1)}`
	
	Result:
	`"2010-05-24T17:10:00z"`

`int minute(DateTime myDateTime)`	Obtém o minuto do valor DateTime como um número inteiro. Exemplo: `#{minute(myDateTime)}` Result: `10`

Função	Descrição
`int month(DateTime myDateTime)`	Obtém o mês do valor DateTime como um número inteiro. Exemplo: `#{month(myDateTime)}` Result: `5`

`DateTime plusDays(DateTime myDateTime,int daysToAdd)`	Cria um objeto DateTime que é o resultado da adição do número de dias especificado ao DateTime especificado.
	
	Exemplo:
	`#{plusDays(myDateTime,1)}`
	
	Result:
	`"2011-05-25T17:10:00z"`

`DateTime plusHours(DateTime myDateTime,int hoursToAdd)`	Cria um objeto DateTime que é o resultado da adição do número de horas especificado ao DateTime especificado.
	
	Exemplo:
	`#{plusHours(myDateTime,1)}`
	
	Result:
	`"2011-05-24T18:10:00z"`

`DateTime plusMinutes(DateTime myDateTime,int minutesToAdd)`	Cria um objeto DateTime que é o resultado da adição do número de minutos especificado ao DateTime especificado. Exemplo: `#{plusMinutes(myDateTime,1)}` Result: `"2011-05-24 17:11:00z"`

`DateTime plusMonths(DateTime myDateTime,int monthsToAdd)`	Cria um objeto DateTime que é o resultado da adição do número de meses especificado ao DateTime especificado.
	
	Exemplo:
	`#{plusMonths(myDateTime,1)}`
	
	Result:
	`"2011-06-24T17:10:00z"`

Função	Descrição

`DateTime plusWeeks(DateTime myDateTime,int weeksToAdd)`	Cria um objeto DateTime que é o resultado da adição do número de semanas especificado ao DateTime especificado.
	
	Exemplo:
	`#{plusWeeks(myDateTime,1)}`
	
	Result:
	`"2011-05-31T17:10:00z"`

`DateTime plusYears(DateTime myDateTime,int yearsToAdd)`	Cria um objeto DateTime que é o resultado da adição do número de anos especificado ao DateTime especificado.
	
	Exemplo:
	`#{plusYears(myDateTime,1)}`
	
	Result:
	`"2012-05-24T17:10:00z"`

`DateTime sunday(DateTime myDateTime)`	Cria um objeto DateTime para o domingo anterior, em relação ao DateTime especificado. Se o DateTime especificado for um domingo, o resultado será o DateTime especificado. Exemplo: `#{sunday(myDateTime)}` Result: `"2011-05-22 17:10:00 UTC"`
`int year(DateTime myDateTime)`	Obtém o ano do valor DateTime como um número inteiro. Exemplo: `#{year(myDateTime)}` Result: `2011`

`DateTime yesterday(DateTime myDateTime)`	Cria um objeto DateTime para o dia anterior, em relação ao DateTime especificado. O resultado é o mesmo que minusDays (1).
	
	Exemplo:
	`#{yesterday(myDateTime)}`
	
	Result:
	`"2011-05-23T17:10:00z"`

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.	@actualStartTime @failureReason @resourceStatus
#	Expressão. As expressões são delimitadas por: “#{“ e ”}”, e o conteúdo das chaves é avaliado pelo AWS Data Pipeline. Para obter mais informações, consulte Expressões (p. 104).	#{format(myDateTime,'YYYY-MM-dd hh:mm:ss')} s3://mybucket/#{id}.csv
*	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.	*password

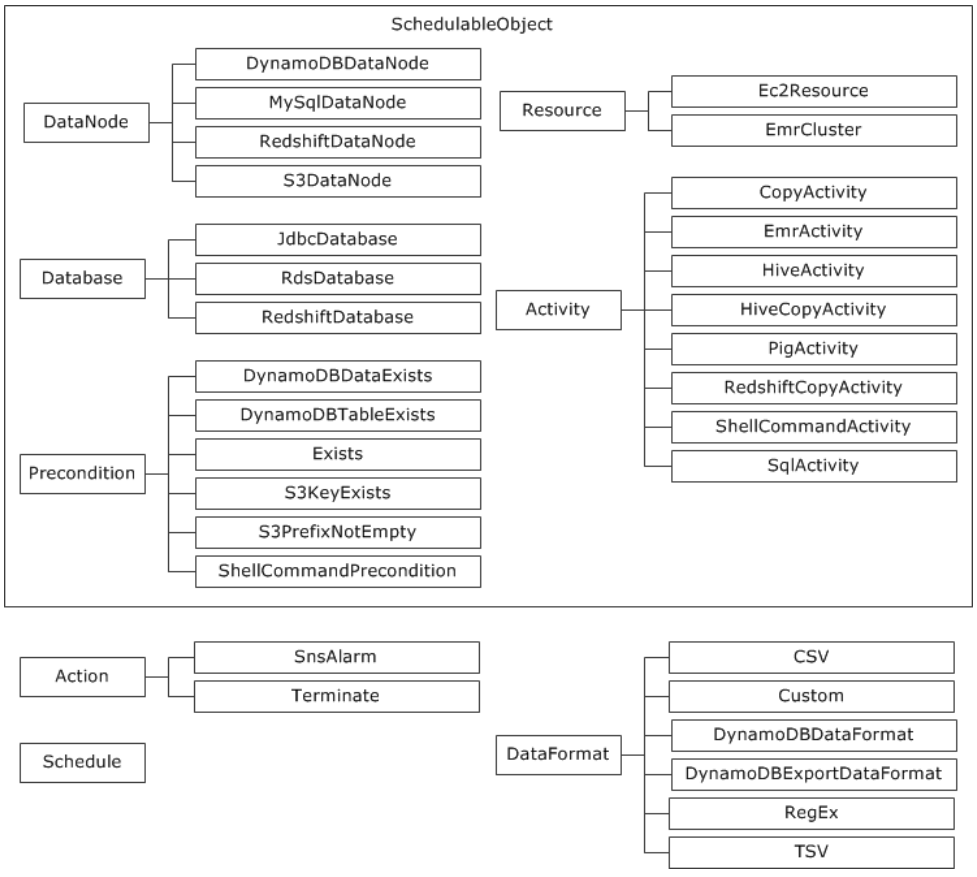
Referência de objeto de pipeline

Você pode usar os objetos e componentes de pipeline a seguir na sua definição de pipeline.

Índice

> **Note**
>
> Para um exemplo de aplicativo que usa oAWS Data PipelineJava SDK, consulteExemplo de exportação de Java do Data Pipeline DynamoDBemGitHub.

Veja a seguir a hierarquia de objetos do AWS Data Pipeline.



Nós de dados

Veja a seguir os objetos de nó de dados do AWS Data Pipeline:

Objetos

DynamoDBDataNode

Define um nó de dados usando o DynamoDB, que é especificado como uma entrada para um`HiveActivity`ou`EMRActivity`objeto.

> Note
>
> O objeto `DynamoDBDataNode` não oferece suporte à precondição `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 precondição.

```
{
  "id" : "MyDynamoDBTable",
  "type" : "DynamoDBDataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "tableName" : "adEvents",
  "precondition" : { "ref" : "Ready" }
}
```

Sintaxe

Campos obrigatórios	Descrição	Tipo de slot
tableName	A tabela do DynamoDB.	Segmento

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.	Objeto de referência, por exemplo, “cronograma”: {"ref":”myScheduleId“}

Campos de invocação de objetos	Descrição	Tipo de slot
	Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando “cronograma”: {"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.	

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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
dataFormat	DataFormatpara os dados descritos por esse nó de dados. Atualmente suportado porHiveActivityeHiveCopyActivity.	Objeto de referência, “dataFormat”: {"ref” :"myDynamoDBDataFormatId“}
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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

Campos opcionais	Descrição	Tipo de slot
		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 URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"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
region	O código da região na qual a tabela do DynamoDB está. Por exemplo, us-east-1. Isso é usado porHiveActivityquando ele executa a preparação de tabelas do DynamoDB no Hive.	Enumeração
reportProgressTimeout	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
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 instância do Amazon EC2 ou um cluster do Amazon EMR.	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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação 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 valor de runsOn e workerGroup existir, workerGroup será ignorado.	Segmento
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”:”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {"

Campos de tempo de execução	Descrição	Tipo de slot
		ref":"myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHorário	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorário	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

MySqlDataNode

Define um nó de dados usando o MySQL.

> Note
>
> O tipo `MySqlDataNode` está obsoleto. Em vez disso, recomendamos o uso de SqlDataNode (p. 135).

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 precondição.

```
{
  "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 banco de dados do MySQL.	Segmento

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	Objeto de referência, por exemplo,

Campos de invocação de objetos	Descrição	Tipo de slot
	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 “cronograma”: {"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.	“cronograma”: {"ref”:”myScheduleId“}

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 expressão de tabela de criação do SQL que cria a tabela.	Segmento
banco de dados	O nome do banco de dados.	Objeto de referência, por exemplo, “banco de dados”: {"ref”:”myDatabaseId“}
dependsOn	Especifica uma 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
insertQuery	Uma instrução do SQL para inserir dados na tabela.	Segmento
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido como`ondemand`.	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	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 URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
reportProgressTimeout	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
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 instância do Amazon EC2 ou um cluster do Amazon EMR.	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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação para cada execução subsequente. Os valores são: cron, ondemand e timeseries.	Enumeração
schemaName	O nome do esquema que mantém a tabela	Segmento
selectQuery	Uma instrução do SQL para obter dados na tabela.	Segmento
workerGroup	O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, workerGroup será ignorado.	Segmento

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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHora	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHora	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- S3DataNode (p. 130)

RedshiftDataNode

Define um nó de dados usando o Amazon Redshift.`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 será criada se ainda não existir e você tiver fornecidocreateTableSql.	Segmento

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	Objeto de referência, por exemplo,

Campos de invocação de objetos	Descrição	Tipo de slot
	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 “cronograma”: {"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.	“cronograma”: {"ref”:”myScheduleId“}

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 expressão SQL 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 distkey,numberOfWinsinteiro (sortKey).AWS Data Pipelineexecuta o script nocreateTableSqlcampo se a tabela, especificada por tableName, não existir no esquema, especificado pelo campo SchemaName. Por exemplo, se você especificar SchemaName como MySchema, mas não incluir MySchema nocreateTableSqlcampo, a tabela é criada no esquema errado (por padrão, ela seria criada em PUBLIC). Isso ocorre porque o AWS Data Pipeline não analisa suas instruções CREATE TABLE.	Segmento
dependsOn	Especifique a dependência em outro objeto executável	Objeto de referência, por exemplo, “dependsOn”: {"ref”:”myActivityId“}

Campos opcionais	Descrição	Tipo de slot
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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“}
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
pipelineLogUri	O URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
primaryKeys	Se você não especificar primaryKeys para uma tabela de destino em RedShiftCopyActivity, poderá especificar uma lista de colunas usando primaryKeys, que agem como um mergeKey. No entanto, se você tiver uma PrimaryKey existente definida em uma tabela do Amazon Redshift, essa configuração substituirá a chave existente.	Segmento
reportProgressTimeout	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
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 instância do Amazon EC2 ou um cluster do Amazon EMR.	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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação 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 é PUBLIC, que é o esquema padrão no Amazon Redshift. Para obter mais informações, consulte o Guia do desenvolvedor do banco de dados do Amazon Redshift.	Segmento
workerGroup	O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, workerGroup será ignorado.	Segmento

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 "}
@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	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo"cascadeFailedOn": {" ref":"myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHora	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHora	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.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

S3DataNode

Define um nó de dados usando o Amazon S3. Por padrão, o S3DataNodeusa criptografia do lado do servidor. Se você quiser desativar isso, defina s3EncryptionTypepara NENHUM.

> Note
>
> Ao usar um `S3DataNode` como entrada para `CopyActivity`, haverá suporte apenas os formatos de dados CSV e TSV.

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.	Objeto de referência, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Campos de invocação de objetos	Descrição	Tipo de slot
	Os usuários podem satisfazer esse requisito definindo explicitamente uma programação no objeto, por exemplo, especificando “cronograma”: {"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.	

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 para os dados descritos pelo S3DataNode. “none” não é compressão e “gzip” é compactado com o algoritmo gzip. Esse campo só é compatível com o Amazon Redshift e quando você usa o S3DataNodecomCopyActivity.	Enumeração
dataFormat	DataFormatpara os dados descritos por este S3DataNode.	Objeto de referência, por exemplo, “dataFormat”: {"ref”:”myDataFormatID"}
dependsOn	Especifique a dependência em outro objeto executável	Objeto de referência, por exemplo, “dependsOn”: {"ref”:”myActivityId“}
directoryPath	Caminho do diretório do Amazon S3 como URI: s3://my-bucket/my-key-for-directory. Você precisa fornecer um valor filePath ou directoryPath.	Segmento
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 como um URI, por exemplo: s3://my-bucket/my-key-	Segmento

Campos opcionais	Descrição	Tipo de slot
	for-file. Você precisa fornecer um valor filePath ou directoryPath. Eles representam um nome de pasta e de arquivo. Use o valor directoryPath para acomodar vários arquivos em um diretório.	
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	Período
manifestFilePath	O caminho do Amazon S3 para um arquivo de manifesto no formato suportado pelo Amazon Redshift.AWS Data Pipelineusa o arquivo de manifesto para copiar os arquivos especificados do Amazon S3 na tabela. Esse campo é válido somente quando umRedShiftCopyActivityfaz referência ao S3DataNode.	Segmento
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“}
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
pipelineLogUri	O URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
reportProgressTimeout	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
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 instância do Amazon EC2 ou um cluster do Amazon EMR.	Objeto de referência, por exemplo, “runsOn”: {"ref”:”myResourceId“}
s3EncryptionType	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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação 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 valor de runsOn e workerGroup existir, workerGroup será ignorado.	Segmento

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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento

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":"myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHora	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHora	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

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”:”myRunnableObjectID "}

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- MySqlDataNode (p. 120)

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 precondição.

```
{
  "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 banco de dados do SQL.	Segmento

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 “cronograma”: {"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.	Objeto de referência, por exemplo, “cronograma”: {"ref":”myScheduleId“}

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 expressão de tabela de criação do SQL que cria a tabela.	Segmento
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, “dependsOn”: {"ref":”myActivityId“}
failureAndRerunModo	Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente.	Enumeração
insertQuery	Uma instrução do SQL para inserir dados na tabela.	Segmento

Campos opcionais	Descrição	Tipo de slot
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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“}
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
pipelineLogUri	O URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
reportProgressTimeout	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
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 instância do Amazon EC2 ou um cluster do Amazon EMR.	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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação para cada execução subsequente. Os valores são: cron, ondemand e timeseries.	Enumeração
schemaName	O nome do esquema que mantém a tabela	Segmento
selectQuery	Uma instrução do SQL para obter dados na tabela.	Segmento
workerGroup	O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, workerGroup será ignorado.	Segmento

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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHoraHora	a 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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorario	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- S3DataNode (p. 130)

Atividades

Veja a seguir os objetos de atividade do AWS Data Pipeline:

Objetos

- CopyActivity (p. 140)
- EmrActivity (p. 146)
- HadoopActivity (p. 152)
- HiveActivity (p. 159)
- HiveCopyActivity (p. 165)
- PigActivity (p. 171)
- RedshiftCopyActivity (p. 181)
- ShellCommandActivity (p. 190)
- SqlActivity (p. 196)

CopyActivity

Copia dados de um local para outro.`CopyActivity`aguentaS3DataNode (p. 130)eSqlDataNode (p. 135)como entrada e saída e a operação de cópia é normalmente executadarecord-by-record. No entanto,`CopyActivity`fornece uma cópia de alto desempenho do Amazon S3 para o Amazon S3 quando todas as seguintes condições são atendidas:

- A entrada e a saída são S3DataNodes
- 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`suporta a cópia de um diretório para outro diretório e a cópia de um arquivo para um diretório, masrecord-by-recorda cópia ocorre ao copiar um diretório em um arquivo. Finalmente,`CopyActivity`não suporta a cópia de arquivos do Amazon S3 com várias partes.

`CopyActivity` tem limitações específicas para suporte a CSV. Quando você usa um S3DataNodecomo entrada para`CopyActivity`, você só pode usar uma variante Unix/Linux do formato de arquivo de dados CSV 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 o valor ASCII 92 (barra invertida).
- O identificador de fim de registro é o valor ASCII 10 (ou "\n").

Os sistemas baseados em Windows normalmente usam um e diferentend-of-recordsequência de caracteres: um retorno de carro e alimentação de linha juntos (valor ASCII 13 e valor ASCII 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 fazer exportações a partir de um objeto PostgreSQL do RDS para um formato de dados TSV, o caractere NULL 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 “cronograma”: {"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	Objeto de referência, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Campos de invocação de objetos	Descrição	Tipo de slot
	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 instância do Amazon EC2 ou um cluster do Amazon EMR.	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, workerGroup será ignorado.	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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
input	A fonte de dados de entrada.	Objeto de referência, por exemplo, “input”: {"ref”:”myDataNodeID "}
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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

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“}
output	A fonte de dados de saída.	Objeto de referência, por exemplo, “saída”: {"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 URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
reportProgressTimeout	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
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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação 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”:”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHoraHora 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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHoraHora 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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@sphere	A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos	Segmento

Campos do sistema	Descrição	Tipo de slot
	objetos de instância que executam os objetos de tentativa.	

Consulte também

- ShellCommandActivity (p. 190)
- EmrActivity (p. 146)
- Exporte dados do MySQL para o Amazon S3 usando AWS Data Pipeline (p. 86)

EmrActivity

Executa um cluster do EMR.

AWS Data Pipelineusa um formato de etapas diferente do Amazon EMR; por exemplo,AWS Data Pipelineusa argumentos separados por vírgula após o nome JAR no`EmrActivity`campo de degraus. O exemplo a seguir mostra uma etapa formatada para o Amazon EMR, seguida por suaAWS Data Pipelineequivalente:

```
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 do Amazon EMR. Verifique a exatidão deste exemplo com a versão do cluster Amazon EMR 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 campo `step` pela string de cluster que quiser. Ela pode ser um script do Pig, um cluster de streaming Hadoop, seu próprio JAR personalizado (incluindo seus respectivos parâmetros) e assim por diante.

Hadoop 2.x (AMI 3.x)

```
{
  "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 argumento no JSON, poderia realizar o escape dele 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	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 `"schedule": {"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), 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, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
runsOn	O cluster do Amazon EMR no qual esse trabalho será executado.	Objeto de referência, por exemplo, “runsOn”: {"ref”:”myEmrClusterID "}

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
workerGroup	O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de `runs0n` e `workerGroup` existir, será ignorado.`workerGroup`	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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”: {"ref”:”myDataNodeID "}
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento 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“}
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	O local dos dados de saída.	Objeto de referência, por exemplo, “saída”: {"ref":”myDataNodeID "}
parent	O pai do objeto atual do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref":”myBaseObjectID "}
pipelineLogUri	O URI do Amazon S3, como 's3://BucketName/Prefix/ 'para fazer upload de registros para o pipeline.	Segmento
postStepCommand	Scripts de shell a serem executados depois que todas as etapas são concluídas. Para especificar vários scripts, até 255, adicione vários campos `postStepCommand`.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"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`.	Segmento
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
resizeClusterBeforeCorrente	Redimensione o cluster antes de realizar essa atividade para acomodar tabelas do DynamoDB especificadas como entradas ou saídas. **Note** Se sua `EmrActivity` usar um `DynamoDBDataNode` como um nó de entrada ou de saída, e se você definir o `resizeClusterBeforeRunning` como `TRUE`, o AWS Data Pipeline passa a usar tipos de instâncias `m3.xlarge`. Isso substitui suas escolhas de tipo de instância por `m3.xlarge`, o que pode aumentar seus custos mensais.	Booliano
resizeClusterMaxInstâncias	Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento.	Inteiro
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 final do intervalo. Os valores são: `cron`, `ondemand` e `timeseries`. A programação `timeseries` significa que as instâncias são programadas no final de cada intervalo. A programação `cron` significa que as instâncias são programadas no início de cada intervalo. Uma programação `ondemand` 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 `ondemand`, ela precisará ser especificada no objeto padrão, além de ser a única `scheduleType` especificada para objetos no pipeline. Para usar pipelines `ondemand`, chame a operação `ActivatePipeline` para cada execução subsequente.	Enumeração
etapa	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 nome JAR. Por exemplo: "`s3://example-bucket/MyWork.jar,arg1,arg2,arg3`".	Segmento

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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo,”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
emrStepLog	Registros de etapas do Amazon EMR disponíveis somente em tentativas de atividade do EMR	Segmento
errorId	A `errorId` se esse objeto apresentou falha.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
errorMessage	A `errorMessage` se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHorário	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorário	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@sphere	A esfera de um objeto denota seu lugar no ciclo de vida: os objetos componentes dão origem aos	Segmento

Campos do sistema	Descrição	Tipo de slot
	objetos de instância que executam os objetos de tentativa.	

Consulte também

- ShellCommandActivity (p. 190)
- CopyActivity (p. 140)
- EmrCluster (p. 208)

HadoopActivity

Executa umMapReducetrabalho em um cluster. O cluster pode ser um cluster EMR gerenciado porAWS Data Pipelineou outro recurso se você usarTaskRunner. UsarHadoopActivityquando você quiser executar o trabalho em paralelo. Isso permite que você use os recursos de agendamento da estrutura YARN ou doMapReducenegociador de recursos no Hadoop 1. Se você quiser executar o trabalho sequencialmente usando a ação Amazon EMR Step, você ainda pode usarEmrActivity (p. 146).

Exemplos

HadoopActivityusando um cluster EMR gerenciado porAWS Data Pipeline

O seguinteHadoopActivityobjeto usa umEmrClusterrecurso 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 oFairSchedulere filas no YARN para AMIs baseadas em Hadoop 2:

```
{
  "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”]
}
```

Este é oEmrClustervocê usa para configurarFairSchedulerno 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"
          }
```

O seguinteEmrClusterconfiguraCapacitySchedulerpara AMIs baseadas em Hadoop 2:

```
{
      "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"
    }
```

HadoopActivityusando um cluster EMR existente

Neste exemplo, você usa grupos de trabalho e umTaskRunnerpara executar um programa em um cluster EMR existente. A seguinte definição de pipeline usaHadoopActivitypara:

- Execute umMapReduceprograma somente em*myWorkerGroup*recursos. Para obter mais informações sobre grupos de operadores, consulte Executando o trabalho em recursos existentes usando o Task Runner (p. 279).
- Execute umpreActivityTaskConfiguração epostActivityTaskConfiguração

```
{
  "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 para execuçãoHadoopActivity.	Segmento

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 “cronograma”: {"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.	Objeto de referência, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
runsOn	Cluster do EMR no qual o trabalho será executado.	Objeto de referência, por exemplo, “runsOn”: {"ref”:”myEmrClusterID "}
workerGroup	O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um valor de runsOn e workerGroup existir, workerGroup será ignorado.	Segmento

Campos opcionais	Descrição	Tipo de slot
argument	Os argumentos a serem transmitidos ao JAR.	Segmento
attemptStatus	Status mais recente da atividade remota.	Segmento
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,

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
hadoopQueue	O nome da fila do programador do Hadoop em que a atividade será enviada.	Segmento
input	Local dos dados de entrada.	Objeto de referência, por exemplo, “input”: {"ref”:”myDataNodeID "}
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	Período
mainClass	A classe principal do JAR com a qual você está executandoHadoopActivity.	Segmento
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, “saída”: {"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 URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
postActivityTaskConfiguração	Script de configuração pós-atividade a ser executado. Consiste em um URI do script de shell no Amazon S3 e uma lista de argumentos.	Objeto de referência, por exemplo”postActivityTaskConfiguração {"ref”:”myShellScriptConfigId“}

Campos opcionais	Descrição	Tipo de slot
preActivityTaskConfiguração	Script de configuração pré-atividade a ser executado. Consiste em um URI do script de shell no Amazon S3 e uma lista de argumentos.	Objeto de referência, por exemplo”preActivityTaskConfiguração”: {"ref”:”myShellScriptConfigId“}
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
reportProgressTimeout	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
retryDelay	A duração do tempo limite entre duas novas tentativas.	Período
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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação 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”:”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime

Campos de tempo de execução	Descrição	Tipo de slot
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo"cascadeFailedOn“: {" ref":"myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHorário	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorário	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

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”:”myRunnableObjectID "}

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandActivity (p. 190)
- CopyActivity (p. 140)
- EmrCluster (p. 208)

HiveActivity

Executa uma consulta do Hive em um cluster do EMR.`HiveActivity`facilita a configuração de uma atividade do Amazon EMR e cria automaticamente tabelas do Hive com base nos dados de entrada provenientes do Amazon S3 ou do Amazon RDS. Tudo o que você precisa especificar é HiveQL para executar nos dados de origem. O AWS Data Pipeline cria automaticamente tabelas do Hive com `${input1}`, `${input2}` e assim por diante, com base nos campos de entrada no objeto `HiveActivity`.

Para entradas do Amazon S3, o`dataFormato` campo é usado para criar os nomes das colunas do Hive.

Para entradas do MySQL (Amazon RDS), os nomes das colunas da consulta SQL são usados para criar os nomes das colunas do Hive.

> Note
>
> Essa atividade usa o CSV Serde do Hive.

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
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. Você pode satisfazer esse requisito definindo explicitamente um cronograma no objeto, por exemplo, especificando “cronograma”: {"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), 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, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
hiveScript	O script Hive a ser executado.	Segmento
scriptUri	O local do script Hive a ser executado (por exemplo, s3://scriptLocation).	Segmento

Grupo obrigatório	Descrição	Tipo de slot
runsOn	O cluster do EMR em que `HiveActivity` está sendo executada.	Objeto de referência, por exemplo, “runsOn”: {"ref”:”myEmrClusterID "}
workerGroup	O grupo de operadores. Isso é usado para tarefas de roteamento. Se você fornecer um	Segmento

Grupo obrigatório	Descrição	Tipo de slot
	valor de `runs0n` e `workerGroup` existir, será ignorado.`workerGroup`	
input	A fonte de dados de entrada.	Objeto de referência, como “input”: {"ref”:”myDataNodeID "}
output	A fonte de dados de saída.	Objeto de referência, como “saída”: {"ref”:”myDataNodeID "}

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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.	Segmento
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento 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, 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“}

Campos opcionais	Descrição	Tipo de slot
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, como “parent”: {"ref”:”myBaseObjectID "}
pipelineLogUri	O URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
postActivityTaskConfiguração	Script de configuração pós-atividade a ser executado. Consiste em um URI do script de shell no Amazon S3 e uma lista de argumentos.	Objeto de referência, como”postActivityTaskConfiguração”: {"ref”:”myShellScriptConfigId“}
preActivityTaskConfiguração	Script de configuração pré-atividade a ser executado. Consiste em um URI do script de shell no Amazon S3 e uma lista de argumentos.	Objeto de referência, como”preActivityTaskConfiguração”: {"ref”:”myShellScriptConfigId“}
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, como “condição prévia”: {"ref”:”myPreconditionId“}
reportProgressTimeout	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
resizeClusterBeforeCorrendo	Redimensione o cluster antes de realizar essa atividade para acomodar os nós de dados do DynamoDB especificados como entradas ou saídas. **Note** Se a sua atividade usar um `DynamoDBDataNode` como um nó de dados de entrada ou de saída, e se você definir o `resizeClusterBeforeRunning` como `TRUE`, o AWS Data Pipeline começará a usar tipos de instâncias `m3.xlarge`. Isso substitui suas escolhas de tipo de instância por `m3.xlarge`, o que pode aumentar seus custos mensais.	Booliano
resizeClusterMaxInstâncias	Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento.	Inteiro
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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação para cada execução subsequente. Os valores são: cron, ondemand e timeseries.	Enumeração
scriptVariable	Especifica variáveis de script para que o Amazon EMR passe para o Hive durante a execução de um script. Por exemplo, as seguintes variáveis do script de exemplo enviariam variáveis SAMPLE e FILTER_DATE para o Hive: `SAMPLE=s3://elasticmapreduce/samples/hive-ads` e `FILTER_DATE=#{format(@scheduledStartTime,'YYYY-MM-dd')}%`. Este campo aceita vários valores e funciona com os campos `script` e `scriptUri`. Além disso, o `scriptVariable` funciona independentemente do estágio estar definido como `true` ou `false`. Este campo é especialmente útil para enviar valores dinâmicos para o Hive usando expressões e funções do AWS Data Pipeline.	Segmento
stage	Determina se a migração de dados está habilitada antes ou depois de executar o script. Não é permitido com o Hive 11, para uso em uma AMI do Amazon EMR versão 3.2.0 ou superior.	Booliano

Campos de tempo de execução	Descrição	Tipo de slot
@activeInstances	Lista dos objetos da instância ativa agendados no momento.	Objeto de referência, como “ActiveInstances”: {"ref":”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime

Campos de tempo de execução	Descrição	Tipo de slot
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, como"cascadeFailedOn": {" ref":"myRunnableObjectID "}
emrStepLog	Os registros de etapas do Amazon EMR estão disponíveis somente em tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHora	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHora	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@waitingOn	Descrição da lista de dependências em que este objeto está aguardando.	Objeto de referência, como “waitingOn”: {"ref":”myRunnableObjectID "}

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandActivity (p. 190)
- EmrActivity (p. 146)

HiveCopyActivity

Executa uma consulta do Hive em um cluster do EMR.`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	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 “cronograma”: {"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	Objeto de referência, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Campos de invocação de objetos	Descrição	Tipo de slot
	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	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`	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	O status mais recente da atividade remota.	Segmento
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 instrução SQL do Hive que filtra um subconjunto de dados do DynamoDB ou do Amazon S3 para copiar. O filtro deve conter apenas predicados e não começar com uma cláusula `WHERE`, pois o AWS Data Pipeline a adiciona automaticamente.	Segmento
input	A fonte de dados de entrada. Deve ser `S3DataNode` ou `DynamoDBDataNode`. Se você usar `DynamoDBNode`, especifique um `DynamoDBExportDataFormat`.	Objeto de referência, por exemplo, “input”: {"ref":”myDataNodeID "}

Campos opcionais	Descrição	Tipo de slot
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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“}
output	A fonte de dados de saída. Se a entrada for S3DataNode, a saída precisará ser DynamoDBDataNode. Caso contrário, ela poderá ser S3DataNode ou DynamoDBDataNode. Se você usar DynamoDBNode, especifique um DynamoDBExportDataFormat.	Objeto de referência, por exemplo, “saída”: {"ref”:”myDataNodeID "}
parent	O pai do objeto atual do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
pipelineLogUri	O URI do Amazon S3, como 's3://BucketName/Key/', para fazer upload de registros para o pipeline.	Segmento
postActivityTaskConfiguração	O script de configuração pós-atividade a ser executado. Consiste em um URI do script de shell no Amazon S3 e uma lista de argumentos.	Objeto de referência, por exemplo”postActivityTaskConfiguração {"ref”:”myShellScriptConfigId“}
preActivityTaskConfiguração	O script de configuração pré-atividade a ser executado. Consiste em um URI do script de shell no Amazon S3 e uma lista de argumentos.	Objeto de referência, por exemplo”preActivityTaskConfiguração”: {"ref”:”myShellScriptConfigId“}
precondition	Opcionalmente define uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}

Campos opcionais	Descrição	Tipo de slot
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
resizeClusterBeforeCorrente	Redimensione o cluster antes de realizar essa atividade para acomodar os nós de dados do DynamoDB especificados como entradas ou saídas. Note Se a sua atividade usar um `DynamoDBDataNode` como um nó de dados de entrada ou de saída, e se você definir o `resizeClusterBeforeRunning` como `TRUE`, o AWS Data Pipeline começará a usar tipos de instâncias `m3.xlarge`. Isso substitui suas escolhas de tipo de instância por `m3.xlarge`, o que pode aumentar seus custos mensais.	Booliano
resizeClusterMaxInstâncias	Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento	Inteiro
retryDelay	A duração do tempo limite entre duas novas tentativas.	Período
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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação 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":”myRunnableObjectID"}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID"}
emrStepLog	Os registros de etapas do Amazon EMR estão disponíveis somente em tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHora	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHora	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 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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandActivity (p. 190)
- EmrActivity (p. 146)

PigActivity

PigActivityfornece suporte nativo para scripts Pig emAWS Data Pipelinesem a necessidade de usar`ShellCommandActivity`ou`EmrActivity`. Além disso,PigActivitysuporta o 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 faz o upload para o Amazon S3 como uma segunda saída.
- MyPigActivity3 carrega o segundo dado de saída, insere duas linhas de dados e somente a coluna chamada “quinta” no Amazon RDS.
- MyPigActivity4 carrega dados do Amazon RDS, 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/pigTestScipt.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
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 “cronograma”: {"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.	Objeto de referência, por exemplo, “cronograma”: {"ref":”mySchedule Id“}

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
script	O script do Pig a ser executado.	Segmento
scriptUri	O local do script do Pig a ser executado (por exemplo, s3://scriptLocation).	Segmento

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
runsOn	Cluster do EMR em que PigActivity está sendo executada.	Objeto de referência, por exemplo, “runsOn”: {"ref":”myEmrClusterID "}
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`	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	O status mais recente da atividade remota.	Segmento
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”: {"ref”:”myDataNodeID "}
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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“}
output	A fonte de dados de saída.	Objeto de referência, por exemplo, “saída”: {"ref”:”myDataNodeID "}
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
pipelineLogUri	O URI do Amazon S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
postActivityTaskConfiguração	Script de configuração pós-atividade a ser executado. Isso consiste em um URI do script do shell no Amazon S33 e em uma lista de argumentos.	Objeto de referência, por exemplo,"postActivityTaskConfiguraçã {"ref":"myShellScriptConfigId"}
preActivityTaskConfiguração	Script de configuração pré-atividade a ser executado. Consiste em um URI do script de shell no Amazon S3 e uma lista de argumentos.	Objeto de referência, por exemplo,"preActivityTaskConfiguração" {"ref":"myShellScriptConfigId"}
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, "condição prévia": {"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
resizeClusterBeforeCorrendo	Redimensione o cluster antes de realizar essa atividade para acomodar os nós de dados do DynamoDB especificados como entradas ou saídas. **Note** Se a sua atividade usar um `DynamoDBDataNode` como um nó de dados de entrada ou de saída, e se você definir o `resizeClusterBeforeRunning` como `TRUE`, o AWS Data Pipeline começará a usar tipos de instâncias `m3.xlarge`. Isso substitui suas escolhas de tipo de instância por `m3.xlarge`, o que pode aumentar seus custos mensais.	Booliano
resizeClusterMaxInstâncias	Um limite no número máximo de instâncias que pode ser solicitado pelo algoritmo de redimensionamento.	Inteiro
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 precisará ser especificada no objeto padrão, além de ser a única scheduleType especificada para objetos no pipeline. Para usar pipelines sob demanda, basta ligar para oActivatePipelineoperação para cada execução subsequente. Os valores são: cron, ondemand e timeseries.	Enumeração
scriptVariable	Os argumentos a serem transmitidos para o script do Pig. Você pode usar scriptVariable com script ou scriptUri.	Segmento
stage	Determina se a preparação está ou não habilitada e permite que o script do Pig acesse as tabelas de dados preparados, como ${INPUT1} e ${OUTPUT1}	Booliano

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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo,”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
emrStepLog	Os registros de etapas do Amazon EMR estão disponíveis somente em tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHora	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHora	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandActivity (p. 190)
- EmrActivity (p. 146)

RedshiftCopyActivity

Copia dados do DynamoDB ou do 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 usandoAWS Data Pipelinepara organizar seus dados no Amazon S3.
2. Usar`RedshiftCopyActivity`para mover os dados do Amazon RDS e do Amazon EMR para o Amazon Redshift.

   Isso permite que você carregue seus dados no Amazon Redshift, onde você pode analisá-los.
3. UsarSqlActivity (p. 196)para realizar consultas SQL 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 S3DataNode (p. 130).

Exemplo

Veja a seguir um exemplo deste tipo de objeto.

Para garantir a conversão de formatos, este exemplo usa os parâmetros de conversão especiais EMPTYASNULL e IGNOREBLANKLINES em `commandOptions`. Para obter informações, consulteParâmetros de conversão de dadosnaGuia do desenvolvedor do banco de dados 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",
      "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": "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 Copie dados para o Amazon Redshift usando AWS Data Pipeline (p. 94).

Sintaxe

Campos obrigatórios	Descrição	Tipo de slot
insertMode	Determina o que o AWS Data Pipeline faz com os dados preexistentes na tabela de destino que	Enumeração

Campos obrigatórios	Descrição	Tipo de slot
	se sobrepõem com linhas aos dados a serem carregados. Os valores válidos são: `KEEP_EXISTING`, `OVERWRITE_EXISTING`, `TRUNCATE` e `APPEND`. `KEEP_EXISTING` adiciona novas linhas à tabela deixando quaisquer linhas existentes sem modificações. `KEEP_EXISTING` e `OVERWRITE_EXISTING` usam as chaves primária, de classificação e de distribuição para identificar quais linhas de entrada correspondem a linhas existentes. VejaAtualizando e inserindo novos dadosno Amazon RedshiftGuia do desenvolvedor de banco de dados. `TRUNCATE` exclui todos os dados na tabela de destino antes de gravar os novos dados. `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.	

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. 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"}`. Se a programação principal do seu pipeline contiver programações aninhadas, crie um objeto pai que tenha uma referência de programação. Para obter mais informações sobre configurações opcionais de programação de exemplo, consulte Programação.	Objeto de referência, como: `"schedule": {"ref":"myScheduleId"}`

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 instância do Amazon EC2 ou um cluster do Amazon EMR.	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, workerGroup será ignorado.	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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	Usa parâmetros para passar para o nó de dados do Amazon Redshift durante o`COPY`operação. Para obter informações sobre parâmetros, consulteCOPIARno Amazon RedshiftGuia do desenvolvedor de banco de dados. 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, consulteParâmetros de conversão de dadosno Amazon RedshiftGuia do desenvolvedor de banco de dados. Se um formato de dados é associado ao nó de dados de entrada ou saída, os parâmetros fornecidos são ignorados. 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`.	Segmento

Campos opcionais	Descrição	Tipo de slot
	Além disso, em alguns casos, quando é necessário descarregar dados do cluster do Amazon Redshift e criar arquivos no Amazon S3, o`RedshiftCopyActivity`depende do`UNLOAD`operação do Amazon Redshift. Para melhorar o desempenho ao copiar e descarregar, especifique o parâmetro `PARALLEL OFF` do comando `UNLOAD`. Para obter informações sobre parâmetros, consulte`DESCARREGAR`no Amazon RedshiftGuia do desenvolvedor de banco de dados.	
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento 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"}`
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"}`

Campos opcionais	Descrição	Tipo de slot
pipelineLogUri	O URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência: `"precondition": {"ref":"myPreconditionId"}`
fila	Corresponde a`query_group` configuração no Amazon Redshift, que permite atribuir e priorizar atividades simultâneas com base em sua colocação nas filas. O Amazon Redshift limita o número de conexões simultâneas a 15. Para obter mais informações, consulteAtribuindo consultas às filasno Amazon RDSGuia do desenvolvedor de banco de dados.	Segmento
reportProgressTimeout	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
retryDelay	A duração do tempo limite entre duas novas tentativas.	Período
scheduleType	Permite que você especifique a programação para objetos no pipeline. Os valores são: `cron`, `ondemand` e `timeseries`. A programação `timeseries` significa que as instâncias são programadas no final de cada intervalo. A programação `Cron` significa que as instâncias são programadas no início de cada intervalo. Uma programação `ondemand` 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. Para usar pipelines `ondemand`, chame a operação `ActivatePipeline` para cada execução subsequente. Se você usar uma programação `ondemand`, deverá especificá-la no objeto padrão, e este deverá ser o único `scheduleType` especificado para objetos no pipeline.	Enumeração

Campos opcionais	Descrição	Tipo de slot
transformSql	A expressão `SQL SELECT` usada para transformar os dados de entrada. Execute a expressão `transformSql` na tabela chamada `staging`. Quando você copia dados do DynamoDB ou do Amazon S3,AWS Data Pipelinecria uma tabela chamada “preparação” e inicialmente carrega os dados nela. Os dados dessa tabela são usados para atualizar a tabela de destino. O esquema de saída de `transformSql` deve corresponder ao esquema da tabela de destinos finais. Se você especificar a opção `transformSql`, uma segunda tabela de preparação será criada a partir da instrução SQL especificada. Os dados na segunda tabela de preparação são, então, atualizados na tabela de destino final.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@activeInstances	Lista dos objetos da instância ativa agendados no momento.	Objeto de referência: `"activeInstances": {"ref":"myRunnableObjectId"}`
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência: `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime

Campos de tempo de execução	Descrição	Tipo de slot
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHoraInstância	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHoraInício	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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

ShellCommandActivity

Executa um comando ou script. Você pode usar `ShellCommandActivity` para executar séries temporais ou tarefas programadas parecidas com Cron.

Quando `ostageo` campo é definido como verdadeiro e usado com um`S3DataNode`,`ShellCommandActivity`suporta o conceito de armazenamento de dados, o que significa que você pode mover dados do Amazon S3 para um local de estágio, como o Amazon EC2 ou seu ambiente local, realizar trabalhos nos dados usando scripts e o`ShellCommandActivity`e mova-o de volta para 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 shell pode ser armazenada em um diretório de saída para ser enviada automaticamente para o Amazon S3, referido 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	Esse objeto é invocado durante a execução de um intervalo `schedule`. Para definir a ordem de execução de dependência desse objeto, especifique uma referência `schedule` a outro objeto. Para atender a esse requisito, defina explicitamente um `schedule` no objeto, por exemplo, especificando `"schedule": {"ref": "DefaultSchedule"}`. Na maioria dos casos, é melhor colocar a referência `schedule` no objeto de pipeline padrão para que todos os objetos herdem essa programação. Se o pipeline consiste em uma	Objeto de referência, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Campos de invocação de objetos	Descrição	Tipo de slot
	árvore de programações (programações aninhadas na programação principal), crie um objeto pai que tenha uma referência de programação. Para distribuir a carga, o AWS Data Pipeline cria objetos físicos um pouco antes da programação, mas os executa de acordo com a 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
command	O comando a ser executado. Use $ para fazer referência aos parâmetros posicionais e `scriptArgument` 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.	Segmento
scriptUri	Um caminho de URI do Amazon S3 para um arquivo do qual você fará download e executará como um comando shell. Especifique somente um campo `scriptUri` ou `command`. `scriptUri` não pode usar parâmetros, portanto, em vez disso, use `command`.	Segmento

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 instância do Amazon EC2 ou um cluster do Amazon EMR.	Objeto de referência, por exemplo, “runsOn”: {"ref”:”myResourceId“}
workerGroup	Usado para tarefas de roteamento. Se você fornecer um valor de `runs0n` e `workerGroup` existir, será ignorado.`workerGroup`	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	O status mais recente da atividade remota.	Segmento

Campos opcionais	Descrição	Tipo de slot
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“}
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”: {"ref”:”myDataNodeID "}
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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 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, “saída”: {"ref”:”myDataNodeID "}
parent	O pai do objeto atual do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
pipelineLogUri	O URI do Amazon S3, como`'s3://BucketName/Key/'`para fazer upload de registros para o pipeline.	Segmento

Campos opcionais	Descrição	Tipo de slot
precondition	Opcionalmente define uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
reportProgressTimeout	O tempo limite para chamadas sucessivas para `reportProgress` por atividades remotas. 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 novas tentativas.	Período
scheduleType	Permite que você especifique se os objetos na definição do pipeline devem ser programados no início ou no final do intervalo. Os valores possíveis são: `cron`, `ondemand` e `timeseries`. Se definido como `timeseries`, as instâncias são programadas no final de cada intervalo. Se definido como `Cron`, as instâncias são programadas no início de cada intervalo. Se definido como `ondemand`, 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 `ondemand`, deverá especificá-la no objeto padrão como o único `scheduleType` para objetos no pipeline. Para usar pipelines `ondemand`, chame a operação `ActivatePipeline` para cada execução subsequente.	Enumeração
scriptArgument	Um conjunto de strings em formato JSON para ser passado ao comando especificado pelo comando. Por exemplo, se o comando for `echo $1 $2`, especifique `scriptArgument` como `"param1", "param2"`. Para vários argumentos e parâmetros, passe o `scriptArgument` da seguinte forma: `"scriptArgument":"arg1","scriptArgument":"param1","scriptArgument":"a` O `scriptArgument` só pode ser usado com `command`. Usá-lo com `scriptUri` causa um erro.	Segmento
stage	Determina se a preparação está ou não ativada e permite que os comandos shell tenham acesso às variáveis de dados preparados, como `${INPUT1_STAGING_DIR}` e `${OUTPUT1_STAGING_DIR}`.	Booliano

Campos opcionais	Descrição	Tipo de slot
stderr	O caminho do que recebe mensagens de erro do sistema redirecionadas do comando. Se você usar `oruns0n`campo, esse deve ser um caminho do Amazon S3 devido à natureza transitória do recurso que executa sua atividade. No entanto, se você especificar o campo `workerGroup`, poderá usar um caminho de arquivo local.	Segmento
stdout	O caminho do Amazon S3 que recebe a saída redirecionada do comando. Se você usar `oruns0n`campo, esse deve ser um caminho do Amazon S3 devido à natureza transitória do recurso que executa sua atividade. No entanto, se você especificar o campo `workerGroup`, poderá usar um caminho de arquivo local.	Segmento

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”:”myRunnableObjectID "}
@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 `cancellationReason` se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	A descrição da cadeia de dependências que causou a falha no objeto.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
emrStepLog	Os registros de etapas do Amazon EMR estão disponíveis somente nas tentativas de atividade do Amazon EMR.	Segmento
errorId	A `errorId` se esse objeto apresentou falha.	Segmento
errorMessage	A `errorMessage` se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	O horário em que a execução do objeto foi concluída.	DateTime

Campos de tempo de execução	Descrição	Tipo de slot
hadoopJobLog	Registros de tarefas do Hadoop disponíveis em tentativas de atividades baseadas no Amazon EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	O ID do último objeto de instância que entrou em um estado concluído.	Segmento
@healthStatusUpdatedHorário	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorário	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.	Segmento
@versão	A versão do AWS Data Pipeline usada para criar o objeto.	Segmento
@waitingOn	A descrição da lista de dependências pelas quais esse objeto está aguardando.	Objeto de referência, por exemplo, “waitingOn”: {"ref”:”myRunnableObjectID "}

Campos do sistema	Descrição	Tipo de slot
@error	O erro ao descrever o objeto malformado.	Segmento
@pipelineId	O ID do pipeline ao qual esse objeto pertence.	Segmento
@sphere	O local de um objeto no ciclo de vida. Objetos de componentes dão origem a objetos de instância, que executam objetos de tentativa.	Segmento

Consulte também

- CopyActivity (p. 140)
- EmrActivity (p. 146)

SqlActivity

Executa uma consulta SQL (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 em que o script SQL fornecido será executado.	Objeto de referência, por exemplo, “banco de dados”: {"ref”:”myDatabaseId“}

Campos de invocação de objetos	Descrição	Tipo de slot
schedule	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 `"schedule": {"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. 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	Objeto de referência, por exemplo, “cronograma”: {"ref”:”myScheduleId“}

Campos de invocação de objetos	Descrição	Tipo de slot
	https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
script	O script SQL a ser executado. Você deve especificar script ou scriptUri. Quando o script é armazenado no Amazon S3, o script não é avaliado como uma expressão. Especificar vários valores para scriptArgument é útil quando o script é armazenado no Amazon S3.	Segmento
scriptUri	Um URI especificando o local de um script SQL a ser executado nesta atividade.	Segmento

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 instância do Amazon EC2 ou um cluster do Amazon EMR.	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 `runs0n` e `workerGroup` existir, será ignorado.`workerGroup`	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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

Campos opcionais	Descrição	Tipo de slot
input	Local dos dados de entrada.	Objeto de referência, por exemplo, “input”: {"ref”:”myDataNodeID "}
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 agendado ou ainda não tiver sido concluído no período desde o início programado do pipeline, conforme especificado por 'lateAfterTimeout'.	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. Isso só é útil para fazer referência a partir de um script (por exemplo`#{output.tablename}`) e para criar a tabela de saída definindo 'createTableSql'no nó de dados de saída. O resultado da consulta SQL não é gravado no nó de dados de saída.	Objeto de referência, por exemplo, “saída”: {"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 URI do S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
precondition	Se desejar, você pode definir uma precondição. Um nó de dados não fica marcado como "READY" até que todas as precondições tenham sido atendidas.	Objeto de referência, por exemplo, “condição prévia”: {"ref”:”myPreconditionId“}
fila	[Apenas para o Amazon Redshift] Corresponde à configuração query_group 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.	Segmento

Campos opcionais	Descrição	Tipo de slot
reportProgressTimeout	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
retryDelay	A duração do tempo limite entre duas novas tentativas.	Período
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. Os valores são: `cron`, `ondemand` e `timeseries`. A programação `timeseries` significa que as instâncias são programadas no final de cada intervalo. A programação `cron` significa que as instâncias são programadas no início de cada intervalo. Uma programação `ondemand` 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 `ondemand`, ela precisará ser especificada no objeto padrão, além de ser a única `scheduleType` especificada para objetos no pipeline. Para usar pipelines `ondemand`, chame a operação `ActivatePipeline` para cada execução subsequente.	Enumeração
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: # {format (@scheduledStartTime, “YY-MM-DD HH:MM:SS"}\ n# {format (PlusPeriod (@scheduledStartTime, “1 dia”), “YY-MM-DD HH:MM:SS"}	Segmento

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 "}
@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	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHorário	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorário	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.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Recursos

Veja a seguir os objetos de recursos do AWS Data Pipeline:

Objetos

- Ec2Resource (p. 201)
- EmrCluster (p. 208)
- HttpProxy (p. 229)

Ec2Resource

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

AWS Data Pipelineagora oferece suporte ao IMDSv2 para a instância do Amazon EC2, que usa um método orientado à sessão para lidar melhor com a autenticação ao recuperar informações de metadados das instâncias. Uma sessão inicia e termina uma série de solicitações que o software executado em uma instância do Amazon EC2 usa para acessar os metadados e as credenciais da instância do Amazon EC2 armazenados localmente. O software inicia uma sessão com uma simples solicitação HTTP PUT para IMDSv2. O IMDSv2 retorna um token secreto para o software em execução na instância do Amazon EC2, que usará o token como senha para fazer solicitações ao IMDSv2 para obter metadados e credenciais.

> Note
>
> Para usar o IMDSv2 na sua instância do Amazon EC2, você precisará modificar as configurações, pois a AMI padrão não é compatível com o IMDSv2. Você pode especificar uma nova versão da AMI que pode ser recuperada por meio do seguinte parâmetro SSM:`/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs`.

Para obter informações sobre instâncias padrão do Amazon EC2 queAWS Data Pipelinecria se você não especificar uma instância, consulte Instâncias padrão do Amazon EC2 por região da AWS (p. 8).

Exemplos

EC2-Classic

> **Important**
>
> SomenteAWSas 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 objetos EC2Resource para um pipeline em uma rede EC2-Classic em vez de uma VPC. É altamente recomendável que você crie recursos para todos os seus pipelines em VPCs. Além disso, se você tiver recursos existentes no EC2-Classic, recomendamos que você os migre para uma VPC.

O objeto de exemplo a seguir inicia uma instância do EC2 no 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 exemplo a seguir inicia uma instância do EC2 em uma VPC não padrão, 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 função do IAM que controla os recursos que a instância do Amazon EC2 pode acessar.	Segmento
função	O papel do IAM queAWS Data Pipelineusa para criar a instância do EC2.	Segmento

Campos de invocação de objetos	Descrição	Tipo de slot
schedule	Esse objeto é invocado durante a execução de um intervalo de programação. 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: • Para garantir que todos os objetos no pipeline herdem a programação, defina uma programação no objeto explicitamente: `"schedule": {"ref": "DefaultSchedule"}`. Na maioria dos casos, é útil colocar a referência de programação no objeto de pipeline padrão para que todos os objetos herdem essa programação. • 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.	Objeto de referência. Por exemplo `"schedule": {"ref":"myScheduleId"}`

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"`.	Segmento
actionOnTaskFalha	A ação executada após uma falha de tarefa para este recurso. Os valores válidos são `"continue"` ou `"terminate"`.	Segmento
associatePublicIpEndereço	Indica se um endereço IP público deve ou não ser atribuído à instância. Se a instância estiver no Amazon EC2 ou no Amazon VPC, o valor padrão será `true`. Caso contrário, o valor padrão será `false`.	Booliano
attemptStatus	Status mais recente da atividade remota.	Segmento
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
availabilityZone	A zona de disponibilidade na qual iniciar a instância do Amazon EC2.	Segmento

Campos opcionais	Descrição	Tipo de slot
Desativar IMDSv1	O valor padrão é false e ativa o IMDSv1 e o IMDSv2. Se você definir como verdadeiro, ele desabilitará o IMDSv1 e fornecerá apenas IMDSv2s	Booliano
failureAndRerunModo	Descreve o comportamento do nó do consumidor quando as dependências apresentam falhas ou são executadas novamente.	Enumeração
httpProxy	O host do proxy que os clientes usam para a conexão com serviços da AWS.	Objeto de referência. Por exemplo: `"httpProxy": {"ref":"myHttpProxyId"}`
imageId	O ID da AMI a ser usado para a instância. Por padrão, o AWS Data Pipeline usa o tipo de virtualização da AMI do HVM. Os IDs específicos da AMI utilizados são baseados em uma região. Você pode substituir a AMI padrão especificando a AMI do HVM de sua escolha. Para obter mais informações sobre os tipos de AMI, consulteTipos de virtualização do Linux AMIeEncontrando uma AMI do LinuxnaGuia do usuário do Amazon EC2 para instâncias Linux.	Segmento
initTimeout	A quantidade de tempo de espera antes da inicialização do recurso.	Período
instanceCount	Suspenso.	Inteiro
instanceType	O tipo de instância do Amazon EC2 a ser iniciada.	Segmento
keyPair	O nome do par de chaves. Se você iniciar uma instância do Amazon EC2 sem especificar um par de chaves, não poderá fazer login nela.	Segmento
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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 `"onFail": {"ref":"myActionId"}`

Campos opcionais	Descrição	Tipo de slot
onLateAction	Ações que devem ser acionadas se um objeto ainda não foi programado ou ainda está em execução.	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	O pai do objeto atual a partir do qual os slots são herdados.	Objeto de referência. Por exemplo: `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	O URI do Amazon S3 (como `'s3://BucketName/Key/'`) para fazer upload de registros para o pipeline.	Segmento
region	O código da região na qual a instância do Amazon EC2 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 `reportProgress`. 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 deve executar oTaskRunner.	Segmento
runsOn	Esse campo não é permitido neste objeto.	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 do intervalo, no final do intervalo ou sob demanda. Os valores são: • `timeseries`. As instâncias são programadas no final de cada intervalo. • `cron`. As instâncias são programadas no início de cada intervalo. • `ondemand`. 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 `scheduleType` especificado para objetos no pipeline. Para usar pipelines sob demanda, chame a operação `ActivatePipeline` para cada execução subsequente.	Enumeração
securityGroupIds	Os IDs de um ou mais grupos de segurança do Amazon EC2 a serem usados para as instâncias no pool de recursos.	Segmento
securityGroups	Um ou mais grupos de segurança do Amazon EC2 para usar nas instâncias no pool de recursos.	Segmento
spotBidPrice	O valor máximo por hora para sua instância spot em dólares, que é um valor decimal entre 0 e 20,00, exclusivos.	Segmento
subnetId	O ID da sub-rede do Amazon EC2 em que a instância será iniciada.	Segmento
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.	Booliano
workerGroup	Esse campo não é permitido neste objeto.	Segmento

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	O `cancellationReason` se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependências na qual o objeto apresentou falha.	Objeto de referência. Por exemplo: `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	Os registros de etapas estão disponíveis somente nas tentativas de atividade do Amazon EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@failureReason	O motivo da falha de recurso.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registros de tarefas do Hadoop disponíveis em tentativas de atividades do Amazon EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHorário	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorário	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

Campos de tempo de execução	Descrição	Tipo de slot
@status	O status deste objeto.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@sphere	O local de um objeto no ciclo de vida. Objetos de componentes dão origem a objetos de instância, que executam objetos de tentativa.	Segmento

EmrCluster

Representa a configuração de um cluster do Amazon EMR. Este objeto é usado porEmrActivity (p. 146)eHadoopActivity (p. 152)para iniciar um cluster.

Índice

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 os recursos uniformemente por um período significativo de tempo.
- CapacityScheduler— Usa filas para permitir que os 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 de lançamento do Amazon EMR

Uma versão do Amazon EMR 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 instalar e configurar o Amazon EMR ao criar um cluster. Especifique a versão usando o rótulo da versão. Os rótulos de versão estão no formato `emr-x.x.x`. Por exemplo, `emr-5.30.0`. Clusters do Amazon EMR com base no rótulo de lançamento`emr-4.0.0`e depois use `releaseLabel`propriedade para especificar o rótulo de liberação de um`EmrCluster`objeto. Versões anteriores usam a propriedade `amiVersion`.

> Important
>
> Todos os clusters do Amazon EMR criados usando a versão de lançamento 5.22.0 ou posterior usamSignature Versão 4para autenticar solicitações ao 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 lançamento do Amazon EMR compatível com a Signature Version 4. Para versões anteriores, começando com o EMR 4.7.x, a versão mais recente da série foi atualizada para oferecer suporte ao Signature versão 4. Ao usar uma versão anterior do EMR, recomendamos que você use a versão mais recente da série. Além disso, evite versões anteriores ao EMR 4.7.0.

Condições e limitações

Use a versão mais recente do Task Runner

Se você estiver usando um autogerenciado`EmrCluster`objeto com um rótulo de lançamento, use o Task Runner mais recente. Para mais informações sobre o Task Runner, consulte Trabalhando com o Task Runner (p. 277). Você pode configurar valores de propriedade para todas as classificações de configuração do Amazon EMR. Para obter mais informações, consulteConfigurando aplicativosnaGuia de lançamento do Amazon EMR, othe section called “EmrConfiguration” (p. 271), ethe section called “Propriedade” (p. 275)referências de objetos.

Suporte para IMDSv2

Mais cedo,AWS Data Pipelinesuportado somente IMDSv1. Agora,AWS Data Pipelinesuporta IMDSv2 no Amazon EMR 5.23.1, 5.27.1 e 5.32 ou posterior, e Amazon EMR 6.2 ou posterior. O IMDSv2 usa 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 chamadas IMDSv2 criando recursos gerenciados pelo usuário usandoTaskRunner-2,0.

Amazon EMR 5.32 ou posterior e Amazon EMR 6.x

A série de versões Amazon EMR 5.32 ou posterior e 6.x usa o Hadoop versão 3.x, que introduziu mudanças importantes 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.

SeEmrActivity (p. 146)ouHadoopActivity (p. 152)executa um arquivo Jar que tem dependências de uma biblioteca que foi removida no Hadoop 3.x, a etapa falha 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 do Amazon EMR 5.x.

Para corrigir o problema, você deve copiar as dependências do arquivo Jar para o caminho de classe do Hadoop em um`EmrCluster`objeto antes de iniciar o`EmrActivity`ou o`HadoopActivity`. Nós fornecemos um script bash para fazer isso. O script bash está disponível no seguinte local, onde*MyRegion*é oAWSRegião onde seu`EmrCluster`objetos executados, por exemplo`us-west-2`.

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh
```

A maneira de executar o script depende de`EmrActivity`ou`HadoopActivity`é executado em um recurso gerenciado porAWS Data Pipelineou é executado em um recurso autogerenciado.

Se você usa um recurso gerenciado porAWS Data Pipeline, adicione um`bootstrapAction`para o`EmrCluster`objeto. O`bootstrapAction`especifica o script e os arquivos Jar a serem copiados como argumentos. Você pode adicionar até 255`bootstrapAction`campos por`EmrCluster`objeto, e você pode adicionar um`bootstrapAction`campo para um`EmrCluster`objeto que já tem 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*MyJarFilen*é o caminho absoluto no Amazon S3 de um arquivo Jar a ser copiado para o classpath do Hadoop. Por padrão, não especifique arquivos Jar que estejam no caminho de classe do Hadoop.

```
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.jare oemr-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 deve salvar e ativar o pipeline para mudar para o novo`bootstrapAction`para entrar em vigor.

Se você usa um recurso autogerenciado, pode 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.`de um arquivo chamado`datapipeline-jars.sh`nesse diretório. Os arquivos jar fornecidos como argumentos de linha de comando são copiados para um diretório criado pelo script chamado`/home/hadoop/datapipeline_jars`. Se o 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 do uso do`bootstrapAction`mostrado no exemplo anterior. Use espaços em vez de vírgulas entre 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
```

Permissões do Amazon EMR

Ao criar uma função personalizada do IAM, considere cuidadosamente as permissões mínimas necessárias para que seu cluster realize seu trabalho. Certifique-se de conceder acesso aos recursos necessários, como arquivos no Amazon S3 ou dados no Amazon RDS, Amazon Redshift ou DynamoDB. Se você quiser definir `visibleToAllUsers` como "False", sua função precisará

das permissões adequadas. `DataPipelineDefaultRole` não tem essas permissões. Você deve fornecer uma união do`DefaultDataPipelineResourceRole`e`DataPipelineDefaultRole`funções como`EmrCluster`função de objeto ou crie 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 `"schedule": {"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), 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. Por exemplo: `"schedule": {"ref":"myScheduleId"}`

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`".	Segmento
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".	Segmento
additionalMasterSecurityGroupIds	O identificador de security groups mestres adicionais do cluster do EMR, que segue o formulário sg-01XXXX6a. Para obter mais informações, consulteGrupos de segurança adicionais do Amazon EMRnaGuia de gerenciamento do Amazon EMR.	Segmento
additionalSlaveSecurityGroupIds	O identificador de security groups subordinados adicionais do cluster do EMR, que segue o formulário `sg-01XXXX6a`.	Segmento
amiVersion	A versão Amazon Machine Image (AMI) que o Amazon EMR usa para instalar os nós do cluster.	Segmento

Campos opcionais	Descrição	Tipo de slot
	Para obter mais informações, consulte o Guia de gerenciamento do Amazon EMR.	
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 o Amazon EMR versão 4.0 e posterior.	Segmento
attemptStatus	O status mais recente da atividade remota.	Segmento
attemptTimeout	Tempo limite para conclusão do trabalho remoto. Se definida, uma atividade remota não concluída dentro do prazo definido poderá ser executada novamente.	Período
availabilityZone	A zona de disponibilidade na qual o cluster será executado.	Segmento
bootstrapAction	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 `bootstrapAction`. O comportamento padrão é iniciar o cluster sem quaisquer ações de bootstrap.	Segmento
configuração	Configuração para o cluster Amazon EMR. Esse parâmetro é aplicável somente para o Amazon EMR versão 4.0 e posterior.	Objeto de referência. Por exemplo: `"configuration": {"ref":"myEmrConfigurationId"`
coreInstanceBidPreço	O preço spot máximo que você está disposto a pagar pelas instâncias do Amazon EC2. Se uma sugestão de preço for especificada, o Amazon EMR usará instâncias spot para o grupo de instâncias. Especificado em dólares americanos (USD).	Segmento
coreInstanceCount	O número de nós core a serem usados no cluster.	Inteiro
coreInstanceType	O tipo de instância do Amazon EC2 a ser usada para os nós principais. Consulte Instâncias do Amazon EC2 suportadas para clusters do Amazon EMR (p. 9).	Segmento
coreGroupConfiguration	A configuração para o grupo de instâncias principais do cluster Amazon EMR. Esse parâmetro é aplicável somente para o Amazon EMR versão 4.0 e posterior.	Objeto de referência. Por exemplo `“configuration”: {“ref”: “myEmrConfigurationId”}`
coreEbsConfiguration	A configuração dos volumes do Amazon EBS que serão anexados a cada um dos nós principais do grupo principal no cluster do Amazon EMR. Para obter mais informações, consulteTipos de instância que oferecem suporte à otimização do EBSnaGuia do usuário do Amazon EC2 para instâncias Linux.	Objeto de referência. Por exemplo `“coreEbsConfiguration”: {“ref”: “myEbsConfiguration”}`

Campos opcionais	Descrição	Tipo de slot
customAmiId	Aplica-se somente à versão 5.7.0 e posterior do Amazon EMR. Especifica o ID da AMI de uma AMI personalizada a ser usada quando o Amazon EMR provisiona instâncias do Amazon EC2. Ele também pode ser usado em vez das ações de bootstrap para personalizar as configurações dos nós do cluster. Para obter mais informações, consulte o tópico a seguir no Guia de gerenciamento do Amazon EMR. Usando uma AMI personalizada	Segmento
EbsBlockDeviceConfig	A configuração de um dispositivo de bloco do Amazon EBS solicitado associado ao grupo de instâncias. Inclui um número especificado dos volumes que serão associados a cada instância no grupo de instâncias. Inclui `volumesPerInstance` e `volumeSpecification`, em que: • `volumesPerInstance` é o número de volumes do EBS com configuração de volume específica que será associada a cada instância no grupo de instâncias. • `volumeSpecification`são as especificações de volume do Amazon EBS, como tipo de volume, IOPS e tamanho em Gigibytes (GiB), que serão solicitadas para o volume do EBS conectado a uma instância do EC2 no cluster do Amazon EMR.	Objeto de referência. Por exemplo `“EbsBlockDeviceConfig”: {“ref”: “myEbsBlockDeviceConfig”}`
emrManagedMasterSecurityGroupId	O identificador do grupo mestre de segurança do cluster Amazon EMR, que segue a forma de `sg-01XXXX6a`. Para obter mais informações, consulte Configurar grupos de segurança na Guia de gerenciamento do Amazon EMR.	Segmento
emrManagedSlaveSecurityGroupId	O identificador do grupo de segurança escravo do cluster Amazon EMR, que segue o formulário `sg-01XXXX6a`.	Segmento
enableDebugging	Permite a depuração no cluster do Amazon EMR.	Segmento
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 do proxy que os clientes utilizarão na conexão com serviços da AWS.	Objeto de referência, por exemplo, “HttpProxy”: {"ref":”myHttpProxyID "}
initTimeout	A quantidade de tempo de espera antes da inicialização do recurso.	Período

Campos opcionais	Descrição	Tipo de slot
keyPair	O par de chaves do Amazon EC2 a ser usado para fazer login no nó principal do cluster do Amazon EMR.	Segmento
lateAfterTimeout	O tempo decorrido após o início do pipeline dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	Período
masterInstanceBidPreço	O preço spot máximo que você está disposto a pagar pelas instâncias do Amazon EC2. É um valor decimal entre 0 e 20,00, exclusivos. Especificado em dólares americanos (USD). A definição deste valor permite instâncias spot para o nó principal do cluster do Amazon EMR. Se uma sugestão de preço for especificada, o Amazon EMR usará instâncias spot para o grupo de instâncias.	Segmento
masterInstanceType	O tipo de instância do Amazon EC2 a ser usada para o nó principal. Consulte Instâncias do Amazon EC2 suportadas para clusters do Amazon EMR (p. 9).	Segmento
masterGroupConfiguration	A configuração para o grupo de instâncias mestres do cluster do Amazon EMR. Esse parâmetro é aplicável somente para o Amazon EMR versão 4.0 e posterior.	Objeto de referência. Por exemplo `“configuration”: {“ref”: “myEmrConfigurationId”}`
masterEbsConfiguration	A configuração dos volumes do Amazon EBS que serão anexados a cada um dos nós principais no grupo mestre no cluster do Amazon EMR. Para obter mais informações, consulteTipos de instância que oferecem suporte à otimização do EBSnaGuia do usuário do Amazon EC2 para instâncias Linux.	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"}`
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"}`

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":"myBaseObjectId"}`
pipelineLogUri	O URI do Amazon S3 (como 's3://BucketName/Key/ ') para fazer upload de registros para o pipeline.	Segmento
region	O código da região em que o cluster do Amazon EMR 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 liberação para o cluster do EMR.	Segmento
reportProgressTimeout	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
resourceRole	O papel do IAM queAWS Data Pipelineusa para criar o cluster do Amazon EMR. A função padrão é `DataPipelineDefaultRole`.	Segmento
retryDelay	A duração do tempo limite entre duas novas tentativas.	Período
função	A função do IAM passou para o Amazon EMR para criar nós do EC2.	Segmento
runsOn	Esse campo não é permitido neste objeto.	Objeto de referência. Por exemplo: `"runsOn": {"ref":"myResourceId"}`
Configuração de segurança	O identificador da configuração de segurança do EMR que será aplicada ao cluster. Esse parâmetro é aplicável somente para o Amazon EMR versão 4.8.0 e posterior.	Segmento
serviceAccessSecurityGroupId	O identificador do grupo de segurança de acesso ao serviço do cluster Amazon EMR.	String. Segue a forma `sg-01XXXX6a`. Por exemplo: `sg-1234abcd`.

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 final do intervalo. Os valores são: `cron`, `ondemand` e `timeseries`. A programação `timeseries` significa que as instâncias são programadas no final de cada intervalo. A programação `cron` significa que as instâncias são programadas no início de cada intervalo. Uma programação `ondemand` 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 `ondemand`, ela precisará ser especificada no objeto padrão, além de ser a única `scheduleType` especificada para objetos no pipeline. Para usar pipelines `ondemand`, chame a operação `ActivatePipeline` para cada execução subsequente.	Enumeração
subnetId	O identificador da sub-rede na qual iniciar o cluster do Amazon EMR.	Segmento
supportedProducts	Um parâmetro que instala software de terceiros em um cluster do Amazon EMR, por exemplo, uma distribuição terceirizada do Hadoop.	Segmento
taskInstanceBidPreço	O preço máximo de instância spot que você está disposto a pagar por instâncias do EC2. Um valor decimal entre 0 e 20,00, exclusive. Especificado em dólares americanos (USD). Se uma sugestão de preço for especificada, o Amazon EMR usará instâncias spot para o grupo de instâncias.	Segmento
taskInstanceCount	O número de nós de tarefas a serem usados para o cluster do Amazon EMR.	Inteiro
taskInstanceType	O tipo de instância do Amazon EC2 a ser usada para nós de tarefas.	Segmento
taskGroupConfiguration	A configuração do grupo de instâncias de tarefas do cluster Amazon EMR. Esse parâmetro é aplicável somente para o Amazon EMR versão 4.0 e posterior.	Objeto de referência. Por exemplo `“configuration”: {“ref”: “myEmrConfigurationId”}`
taskEbsConfiguration	A configuração dos volumes do Amazon EBS que serão anexados a cada um dos nós de tarefa no grupo de tarefas no cluster do Amazon EMR. Para obter mais informações, consulteTipos de instância que oferecem suporte à otimização do EBSnaGuia do usuário do Amazon EC2 para instâncias Linux.	Objeto de referência. Por exemplo `“taskEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
terminateAfter	Encerrar o recurso após tantas horas.	Inteiro

Campos opcionais	Descrição	Tipo de slot
VolumeSpecification	As especificações de volume do Amazon EBS, como tipo de volume, IOPS e tamanho em Gigibytes (GiB), que serão solicitadas para o volume do Amazon EBS conectado a uma instância do Amazon EC2 no cluster do Amazon EMR. O nó pode ser um nó core, principal ou de tarefa. `VolumeSpecification` inclui: • `iops()` Inteiro. O número de operações de I/O por segundo (IOPS) que o volume do Amazon EBS suporta, por exemplo, 1000. Para obter mais informações, consulteCaracterísticas de E/S do EBSnaGuia do usuário do Amazon EC2 para instâncias Linux. • `sizeinGB()`. Número inteiro. O tamanho do volume do Amazon EBS, em gibibytes (GiB), por exemplo 500. Para obter informações sobre combinações válidas de tipos de volume e tamanhos de disco rígido, consulteTipos de volume do EBSnaGuia do usuário do Amazon EC2 para instâncias Linux. • `volumetType`. Corda. O tipo de volume do Amazon EBS, por exemplo, gp2. Os tipos de volume suportados incluem gp2, io1, ST1, SC1 padrão e outros. Para obter mais informações, consulteTipos de volume do EBSnaGuia do usuário do Amazon EC2 para instâncias Linux.	Objeto de referência. Por exemplo `“VolumeSpecification”: {“ref”: “myVolumeSpecification”}`
useOnDemandOnLastAttempt	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.	Booliano
workerGroup	Campo não é permitido neste objeto.	Segmento

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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime

Campos de tempo de execução	Descrição	Tipo de slot
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependências na qual o objeto apresentou falha.	Objeto de referência, por exemplo,"cascadeFailedOn": {" ref":"myRunnableObjectID "}
emrStepLog	Registros de etapas disponíveis somente nas tentativas de atividade do Amazon EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
@failureReason	O motivo da falha de recurso.	Segmento
@finishedTime	A hora em que esse objeto terminou a execução.	DateTime
hadoopJobLog	Registros de tarefas do Hadoop disponíveis em tentativas de atividades do Amazon EMR.	Segmento
@healthStatus	O status de integridade do objeto que indica se houve sucesso ou falha na última instância concluída do objeto.	Segmento
@healthStatusFromInstanceId	ID do último objeto da instância concluído.	Segmento
@healthStatusUpdatedHorário	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.	Segmento
@lastDeactivatedTime	A hora em que esse objeto foi desativado pela última vez.	DateTime
@latestCompletedRunHorário	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.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@sphere	O local de um objeto no ciclo de vida. Objetos de componentes dão origem a objetos de instância, que executam objetos de tentativa.	Segmento

Exemplos

Veja a seguir exemplos desse tipo de objeto.

Índice

Inicie um cluster do Amazon EMR com HadoopVersion

Example

O exemplo a seguir lança um cluster do Amazon EMR 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 cluster do Amazon EMR com o rótulo de lançamento emr-4.x ou superior

Example

O exemplo a seguir lança um cluster do Amazon EMR usando o mais novo`releaseLabel`campo:

```
{
  "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 cluster Amazon EMR

Example

`EmrCluster`fornece o`supportedProducts`campo que instala software de terceiros em um cluster do Amazon EMR, 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 atividade `EmrCluster` com um Hadoop de versão 2.x criado por AWS Data Pipeline habilita 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"
   }
```

Configurar ACLs do Hadoop KMS e criar zonas de criptografia no HDFS

Example

Os seguintes objetos criam ACLs para o Hadoop KMS, além de zonas de criptografia e chaves de criptografia correspondentes no 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 funções personalizadas do IAM

Example

Por padrão,AWS Data PipelinepassesDataPipelineDefaultRolecomo função de serviço Amazon EMR eDataPipelineDefaultResourceRolecomo o perfil da instância do Amazon EC2 para criar recursos em seu nome. No entanto, você pode criar uma função de serviço personalizada do

Amazon EMR e um perfil de instância personalizado e usá-los em vez disso.AWS Data Pipelinedeve ter permissões suficientes para criar clusters usando a função personalizada e você deve adicionarAWS Data Pipelinecomo uma entidade confiável.

O objeto de exemplo a seguir especifica funções personalizadas para o cluster do Amazon EMR:

```
{
   "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"
}
```

UsarEmrClusterRecurso no AWS SDK para Java

Example

O exemplo a seguir mostra como usar um`EmrCluster`e`EmrActivity`para 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 cluster do Amazon EMR em uma sub-rede privada

Example

Este exemplo inclui uma configuração que executa o cluster em uma sub-rede privada dentro de uma VPC. Para obter mais informações, consulteLance clusters do Amazon EMR em uma VPCnaGuia de gerenciamento do Amazon EMR. Essa configuração é opcional. Você pode usá-la em qualquer pipeline que usa um objeto `EmrCluster`.

Para iniciar um cluster do Amazon EMR em uma sub-rede privada, especifique`SubnetId`,`emrManagedMasterSecurityGroupId`,`emrManagedSlaveSecurityGroupId`, e`serviceAccessSecurityGroupId`no seu`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.tableName
      "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 os volumes do EBS aos nós de cluster

Example

Você pode anexar volumes do EBS a qualquer tipo de nó no cluster do EMR no seu pipeline. Para anexar volumes do EBS aos nós, use `coreEbsConfiguration`, `masterEbsConfiguration` e `TaskEbsConfiguration` na sua configuração `EmrCluster`.

Este exemplo do cluster do Amazon EMR usa volumes do Amazon EBS para seus nós mestre, tarefa e núcleo. Para obter mais informações, consulteVolumes do Amazon EBS no Amazon EMRnaGuia de gerenciamento do Amazon EMR.

Essas configurações são opcionais. Você pode usá-las em qualquer pipeline que usa um objeto `EmrCluster`.

No pipeline, clique na configuração de objeto `EmrCluster`, escolha Master EBS Configuration, Core EBS Configuration ou Task EBS Configuration e insira os detalhes de 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.tableName
      "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 (p. 146)

HttpProxy

HttpProxypermite que você configure seu próprio proxy e faça com que o Task Runner acesse oAWS Data Pipelineserviço por meio dele. Você não precisa configurar um Task Runner em execução com essas informações.

Exemplo de umHttpProxyemTaskRunner

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 utilizarão na conexão com os Serviços da AWS.	Segmento
port	Porta do host do proxy que os clientes utilizarão na conexão com os Serviços da AWS.	Segmento

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 "}
*password	Senha para o proxy.	Segmento
s3NoProxy	Desative o proxy HTTP ao se conectar com o Amazon S3	Booliano
username	Nome do usuário para o proxy.	Segmento
windowsDomain	O nome de domínio do Windows para o proxy NTLM.	Segmento
windowsWorkgroup	O nome do grupo de trabalho do Windows para o proxy NTLM.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Precondições

Veja a seguir os objetos de precondição do AWS Data Pipeline:

Objetos

DynamoDBDataExists

Uma condição prévia para verificar se os dados existem em uma tabela do DynamoDB.

Sintaxe

Campos obrigatórios	Descrição	Tipo de slot
função	Especifica a função a ser usada para executar a precondição.	Segmento
tableName	Tabela do DynamoDB para verificação.	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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

Campos opcionais	Descrição	Tipo de slot
		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 precondição é marcada como "com falha" se ainda não tiver sido atendida.	Período
reportProgressTimeout	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
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”:”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
currentRetryCount	O número de vezes que a precondição foi testada nesta tentativa.	Segmento
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
hostname	O nome do host do cliente que capturou a tentativa da tarefa.	Segmento
lastRetryTime	Última vez em que a precondição foi testada nessa tentativa.	Segmento
nó	O nó para o qual esta precondição está sendo realizada.	Objeto de referência, por exemplo, “node”: {"ref":”myRunnableObjectID"}
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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

DynamoDBTableExists

Uma condição prévia para verificar se a tabela do DynamoDB existe.

Sintaxe

Campos obrigatórios	Descrição	Tipo de slot
função	Especifica a função a ser usada para executar a precondição.	Segmento
tableName	Tabela do DynamoDB para verificação.	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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 precondição é marcada como "com falha" se ainda não tiver sido atendida.	Período
reportProgressTimeout	Tempo limite para as chamadas sucessivas de trabalho remoto para reportProgress. Se definidas, as atividades remotas sem progresso para o	Período

Campos opcionais	Descrição	Tipo de slot
	período especificado podem ser consideradas como interrompidas e executadas novamente.	
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”:”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
currentRetryCount	O número de vezes que a precondição foi testada nesta tentativa.	Segmento
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
hostname	O nome do host do cliente que capturou a tentativa da tarefa.	Segmento
lastRetryTime	Última vez em que a precondição foi testada nessa tentativa.	Segmento
nó	O nó para o qual esta precondição está sendo realizada.	Objeto de referência, por exemplo, “node”:

Campos de tempo de execução	Descrição	Tipo de slot
		{"ref":"myRunnableObjectID"}
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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Existe

Verifica se existe um objeto de nó de dados.

> **Note**
>
> Recomendamos que você use as precondições gerenciadas pelo sistema. Para obter mais informações, consulte Precondições (p. 15).

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.	Segmento
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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 precondição é marcada como "com falha" se ainda não tiver sido atendida.	Período
reportProgressTimeout	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
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”:”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
hostname	O nome do host do cliente que capturou a tentativa da tarefa.	Segmento
nó	O nó para o qual esta precondição está sendo realizada.	Objeto de referência, por exemplo, “node”: {"ref”:”myRunnableObjectID "}
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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@waitingOn	Descrição da lista de dependências em que este objeto está aguardando.	Objeto de referência, por exemplo, “waitingOn”:

Campos de tempo de execução	Descrição	Tipo de slot
		{"ref":"myRunnableObjectID"}

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandPrecondition (p. 245)

S3KeyExists

Verifica se existe uma chave em um nó de dados do Amazon S3.

Exemplo

Veja a seguir um exemplo deste tipo de objeto. A precondição 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 precondição 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 precondição `S3KeyExists` no segundo pipeline.

Sintaxe

Campos obrigatórios	Descrição	Tipo de slot
função	Especifica a função a ser usada para executar a precondição.	Segmento

Campos obrigatórios	Descrição	Tipo de slot
s3Key	A chave do Amazon S3.	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento 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“}
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 precondição é marcada como "com falha" se ainda não tiver sido atendida.	Período
reportProgressTimeout	Tempo limite para as chamadas sucessivas de trabalho remoto para `reportProgress`. Se configurada, as atividades remotas sem progresso para o período especificado poderão ser consideradas como interrompidas e serão executadas novamente.	Período

Campos opcionais	Descrição	Tipo de slot
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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref”:”myRunnableObjectID "}
currentRetryCount	O número de vezes que a precondição foi testada nesta tentativa.	Segmento
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
hostname	O nome do host do cliente que capturou a tentativa da tarefa.	Segmento
lastRetryTime	Última vez em que a precondição foi testada nessa tentativa.	Segmento
nó	O nó para o qual esta precondição está sendo realizada.	Objeto de referência, por exemplo, “node”: {"ref”:”myRunnableObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandPrecondition (p. 245)

S3PrefixNotEmpty

Uma condição prévia para verificar se os objetos do Amazon S3 com o prefixo fornecido (representado como um URI) 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
função	Especifica a função a ser usada para executar a precondição.	Segmento
s3Prefix	O prefixo Amazon S3 para verificar a existência de objetos.	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento não está definido comoondemand.	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 precondição é marcada como "com falha" se ainda não tiver sido atendida.	Período
reportProgressTimeout	Tempo limite para as chamadas sucessivas de trabalho remoto para reportProgress. Se definidas, as atividades remotas sem progresso para o	Período

Campos opcionais	Descrição	Tipo de slot
	período especificado podem ser consideradas como interrompidas e executadas novamente.	
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":”myRunnableObjectID "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“: {" ref":”myRunnableObjectID "}
currentRetryCount	O número de vezes que a precondição foi testada nesta tentativa.	Segmento
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
hostname	O nome do host do cliente que capturou a tentativa da tarefa.	Segmento
lastRetryTime	Última vez em que a precondição foi testada nessa tentativa.	Segmento
nó	O nó para o qual esta precondição está sendo realizada.	Objeto de referência, por exemplo, “node”:

Campos de tempo de execução	Descrição	Tipo de slot
		{"ref":"myRunnableObjectID"}
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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandPrecondition (p. 245)

ShellCommandPrecondition

Um comando shell do Unix/Linux que pode ser executado como uma precondição.

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.	Segmento
scriptUri	Um caminho de URI do Amazon S3 para um arquivo do qual você fará download e executará como um comando shell. Apenas um campo de comando ou scriptUri deve estar presente. scriptUri não pode usar parâmetros, portanto, em vez disso, use o comando.	Segmento

Campos opcionais	Descrição	Tipo de slot
attemptStatus	Status mais recente da atividade remota.	Segmento
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 dentro do qual o objeto deve ser concluído. Ela é acionada somente quando o tipo de agendamento 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”:

Campos opcionais	Descrição	Tipo de slot
		{"ref":"myBaseObjectID "}
preconditionTimeout	O período inicial após o qual a precondição é marcada como "com falha" se ainda não tiver sido atendida.	Período
reportProgressTimeout	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
retryDelay	A duração do tempo limite entre duas novas tentativas.	Período
scriptArgument	Argumento a ser passado para o script de shell	Segmento
stderr	O caminho do Amazon S3 que recebe mensagens de erro redirecionadas do sistema a partir do comando. Se você usar `oruns0n`campo, esse deve ser um caminho do Amazon S3 devido à natureza transitória do recurso que executa sua atividade. No entanto, se você especificar o campo `workerGroup`, poderá usar um caminho de arquivo local.	Segmento
stdout	O caminho do Amazon S3 que recebe a saída redirecionada do comando. Se você usar `oruns0n`campo, esse deve ser um caminho do Amazon S3 devido à natureza transitória do recurso que executa sua atividade. No entanto, se você especificar o campo `workerGroup`, poderá usar um caminho de arquivo local.	Segmento

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 "}
@actualEndTime	Hora em que a execução deste objeto foi concluída.	DateTime
@actualStartTime	Hora em que a execução deste objeto foi iniciada.	DateTime
cancellationReason	O motivo do cancelamento, se esse objeto foi cancelado.	Segmento
@cascadeFailedOn	Descrição da cadeia de dependência na qual o objeto apresentou falha.	Objeto de referência, por exemplo”cascadeFailedOn“:

Campos de tempo de execução	Descrição	Tipo de slot
		{" ref":"myRunnableObjectID "}
emrStepLog	Registros da etapa do EMR disponíveis somente nas tentativas de atividade do EMR.	Segmento
errorId	O ID do erro se esse objeto apresentou falha.	Segmento
errorMessage	A mensagem de erro se esse objeto apresentou falha.	Segmento
errorStackTrace	O rastreamento de pilha com erro se esse objeto apresentou falha.	Segmento
hadoopJobLog	Registos de trabalho do Hadoop disponíveis nas tentativas de atividades baseadas em EMR.	Segmento
hostname	O nome do host do cliente que capturou a tentativa da tarefa.	Segmento
nó	O nó para o qual esta precondição está sendo realizada.	Objeto de referência, por exemplo, "node": {"ref":"myRunnableObjectID "}
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.	Segmento
@versão	A versão do pipeline com que o objeto foi criado.	Segmento
@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 malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- ShellCommandActivity (p. 190)
- Existe (p. 236)

Bancos de dados

Veja a seguir os objetos de banco de dados do AWS Data Pipeline:

Objetos

- JdbcDatabase (p. 249)
- RdsDatabase (p. 250)
- RedshiftDatabase (p. 252)

JdbcDatabase

Define um banco de dados JDBC.

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 string de conexão JDBC para acessar o banco de dados.	Segmento
jdbcDriverClass	A classe de driver a ser carregada antes de estabelecer a conexão JDBC.	Segmento
*password	A senha a ser informada.	Segmento
username	O nome de usuário a ser informado ao se conectar com o banco de dados.	Segmento

Campos opcionais	Descrição	Tipo de slot
databaseName	Nome do banco de dados lógico para se conectar	Segmento

Campos opcionais	Descrição	Tipo de slot
jdbcDriverJarUri	O local no Amazon S3 do arquivo JAR do driver JDBC usado para se conectar ao banco de dados. O AWS Data Pipeline precisa ter permissão para ler esse arquivo JAR.	Segmento
jdbcProperties	Pares da forma A=B que serão definidos como propriedades em conexões JDBC para este banco de dados.	Segmento
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

RdsDatabase

Define um banco de dados do Amazon RDS.

> Note
>
> RdsDatabasenão é compatível com Aurora. Usarthe section called “JdbcDatabase” (p. 249)para Aurora, em vez disso.

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 Server, o campo `jdbcDriverJarUri` é necessário, e você pode especificar o seguinte driver: `https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774`. Para os mecanismos do MySQL e PostgreSQL, o campo `jdbcDriverJarUri` é opcional.

Sintaxe

Campos obrigatórios	Descrição	Tipo de slot
*password	A senha a ser informada.	Segmento
rdsInstanceId	`ODBInstanceIdentifier`propriedade da instância de banco de dados.	Segmento
username	O nome de usuário a ser informado ao se conectar com o banco de dados.	Segmento

Campos opcionais	Descrição	Tipo de slot
databaseName	Nome do banco de dados lógico para se conectar	Segmento
jdbcDriverJarUri	O local no Amazon S3 do arquivo JAR do driver JDBC usado para se conectar ao banco de dados. O AWS Data Pipeline precisa ter permissão para ler esse arquivo JAR. Para os mecanismos MySQL e PostgreSQL, o driver padrão é usado se este campo não for especificado, mas você pode substituir o padrão usando este campo. Para mecanismos Oracle e SQL Server, este campo é obrigatório.	Segmento
jdbcProperties	Pares da forma A=B que serão definidos como propriedades em conexões JDBC para este banco de dados.	Segmento
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
region	O código da região na qual o banco de dados está. Por exemplo, us-east-1.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

RedshiftDatabase

Define um banco de dados do Amazon Redshift.`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 (começa com “jdbc:redshift:”) no`connectionString`campo em vez disso.

Sintaxe

Campos obrigatórios	Descrição	Tipo de slot
*password	A senha a ser informada.	Segmento
username	O nome de usuário a ser informado ao se conectar com o banco de dados.	Segmento

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 exemplo, se o endpoint do seu cluster do Amazon Redshift for mydb.example.us-east-1.redshift.amazonaws.com, o identificador	Segmento

Grupo obrigatório (um dos seguintes é obrigatório)	Descrição	Tipo de slot
	correto serámydb. No console do Amazon Redshift, você pode obter este valor no identificador ou no nome do cluster.	
connectionString	O endpoint JDBC 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`.	Segmento

Campos opcionais	Descrição	Tipo de slot
databaseName	Nome do banco de dados lógico para se conectar.	Segmento
jdbcProperties	Pares da forma A=B que serão definidos como propriedades em conexões JDBC para este banco de dados.	Segmento
parent	Pai do objeto atual a partir do qual os slots são herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
region	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
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Formatos de dados

Veja a seguir os objetos de formato de dados do AWS Data Pipeline:

Objetos

Formatos de dados CSV

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. Ex: nome de host STRING para vários valores. Use nomes de colunas e tipos de dados separados por um espaço.	Segmento
escapeChar	Um caractere, por exemplo"\", que instrui o analisador para ignorar o próximo caractere.	Segmento
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

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.	Segmento

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 de host STRING para vários valores. Use nomes de colunas e tipos de dados separados por um espaço.	Segmento
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
recordSeparator	Um caractere que indica o fim de uma linha em um arquivo de dados, por exemplo "\n". Há suporte apenas para caracteres únicos.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

DynamoDBDataFormat

Aplica um esquema a uma tabela do DynamoDB para torná-la acessível por uma consulta do Hive.`DynamoDBDataFormat`é usado com um`HiveActivity`objeto e um`DynamoDBDataNode`entrada e saída.`DynamoDBDataFormat`exige que você especifique todas as colunas em sua consulta do Hive. Para obter mais flexibilidade para especificar determinadas colunas em uma consulta do Hive ou no suporte do Amazon S3, consulteDynamoDBExportDataFormat (p. 258).

> 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.	Segmento

Campos opcionais	Descrição	Tipo de slot
parent	O pai do objeto atual do qual os slots serão herdados.	Objeto de referência, como “parent”: {"ref”:”myBaseObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline usada para criar o objeto.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	O erro ao descrever o objeto malformado.	Segmento
@pipelineId	O ID do pipeline ao qual esse objeto pertence.	Segmento
@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.	Segmento

DynamoDBExportDataFormat

Aplica um esquema a uma tabela do DynamoDB para torná-la acessível 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 ao DynamoDB e ao Amazon S3
- Permite que você filtre dados por determinadas colunas na sua consulta do Hive
- Exporta todos os atributos do DynamoDB mesmo se você tiver um esquema esparso

> **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 `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: hostname STRING	Segmento

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 "}

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

RegExFormato 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. Ex: nome de host STRING para vários valores. Use nomes de colunas e tipos de dados separados por um espaço.	Segmento
inputRegEx	A expressão regular para analisar um arquivo de entrada do S3. inputRegExfornece uma maneira de recuperar colunas de dados relativamente não estruturados em um arquivo.	Segmento
outputFormat	Os campos da coluna recuperados porinputRegEx, mas referenciado como %1$s %2$s usando a sintaxe do formatador Java.	Segmento
parent	Pai do objeto atual a partir do qual os slots serão herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Formatos de dados TSV

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 tipo de dados com vírgulas (como exibido no exemplo).	Segmento
columnSeparator	O caractere que separa os campos em uma coluna de campos na próxima coluna. Assume '\t' como padrão.	Segmento
escapeChar	Um caractere, por exemplo"\", que instrui o analisador para ignorar o próximo caractere.	Segmento
parent	Pai do objeto atual a partir do qual os slots são herdados.	Objeto de referência, por exemplo, “parent”: {"ref”:”myBaseObjectID "}
recordSeparator	O caractere que separa registros. Assume '\n' como padrão.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Ações

Veja a seguir os objetos de ação do AWS Data Pipeline:

Objetos

SnsAlarm

Envia uma mensagem de notificação do Amazon SNS quando uma atividade falha ou é concluída com êxito.

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
mensagem	O texto do corpo da notificação do Amazon SNS.	Segmento
função	A função do IAM a ser usada para criar o alarme do Amazon SNS.	Segmento
subject	A linha de assunto da mensagem de notificação do Amazon SNS.	Segmento
topicArn	O ARN do tópico do Amazon SNS de destino para a mensagem.	Segmento

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 "}

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":”myRunnableObjectID "}
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Encerrar

Uma ação para acionar o cancelamento de atividades, recursos ou nós de dados pendentes ou não concluídos. O AWS Data Pipeline tenta colocar a atividade, o recurso ou o nó de dados no estado CANCELLED se não forem iniciados pelo valor `lateAfterTimeout`.

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”:”myBaseObjectID "}

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”:”myRunnableObjectID "}
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Schedule

Define o tempo de um evento programado, como quando uma atividade é executada.

> Note
>
> Quando a hora de início de uma programação está em atraso, o AWS Data Pipeline aloca seu pipeline e começa a programar execuções imediatamente após a hora de início especificada. Para testes/desenvolvimento, use um intervalo relativamente curto. Caso contrário, o AWS Data Pipeline tentará criar filas e programar todas as execuções do seu pipeline nesse intervalo. O AWS Data Pipeline tenta evitar alocações acidentais se o componente do pipeline `scheduledStartTime` for mais recente que 1 dia atrás, bloqueando a ativação do pipeline.

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 é obsoleto em favor da criação de um pipeline sob demanda.	Enumeração
startDateTime	A data e a hora para iniciar as execuções programadas. Você deve usar qualquer umstartDateTimeou StartAt, mas não ambos.	DateTime

Campos opcionais	Descrição	Tipo de slot
endDateTime	A data e a hora para terminar as execuções programadas. Deve ser uma data e hora posteriores ao valor destartDateTimeou 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 comendDateTime.	Inteiro

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 "}

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@firstActivationTime	O tempo de criação do objeto.	DateTime
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Utilitários

Os seguintes objetos utilitários configuram outros objetos do pipeline:

Tópicos

ShellScriptConfig

Use com uma atividade para executar um script de shell parapreActivityTaskConfiguração epostActivityTaskConfiguração. Este objeto está disponível paraHadoopActivity (p. 152),HiveActivity (p. 159),HiveCopyActivity (p. 165), ePigActivity (p. 171). Você especifica um URI do S3 e uma lista de argumentos para o script.

Exemplo

UMAShellScriptConfigcom 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”:”myBaseObjectID "}
scriptArgument	Uma lista de argumentos para uso com script de shell.	Segmento
scriptUri	O URI de script no Amazon S3 que deve ser obtido por download e executado.	Segmento

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

EmrConfiguration

OEmrConfigurationobjeto é a configuração usada para clusters EMR com versões 4.0.0 ou superiores. Configurações (como uma lista) é um parâmetro para oRunJobFlowChamada de API. A API de configuração do Amazon EMR usa uma classificação e propriedades.AWS Data PipelineusaEmrConfigurationcom objetos de propriedade correspondentes para configurar umEmrCluster (p. 208)aplicativos como Hadoop, Hive, Spark ou Pig em clusters do EMR lançados em uma execução de pipeline. Como a configuração só pode ser alterada para novos clusters, você não pode fornecer umEmrConfigurationobjeto para 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 umEmrConfigurationobjeto 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 do 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.	Segmento

Campos opcionais	Descrição	Tipo de slot
configuração	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
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- EmrCluster (p. 208)
- Propriedade (p. 275)
- Guia de apresentação do Amazon EMR

Propriedade

Uma única propriedade de valor-chave para uso com umEmrConfigurationobjeto.

Exemplo

A definição de pipeline a seguir mostra umEmrConfigurationobjeto e objetos de propriedade correspondentes para iniciar umEmrCluster:

```
{
  "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
value	value	Segmento

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”:”myBaseObjectID "}

Campos de tempo de execução	Descrição	Tipo de slot
@versão	A versão do pipeline com que o objeto foi criado.	Segmento

Campos do sistema	Descrição	Tipo de slot
@error	Erro ao descrever o objeto malformado.	Segmento
@pipelineId	ID do pipeline ao qual este objeto pertence.	Segmento
@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.	Segmento

Consulte também

- EmrCluster (p. 208)
- EmrConfiguration (p. 271)
- Guia de apresentação do Amazon EMR

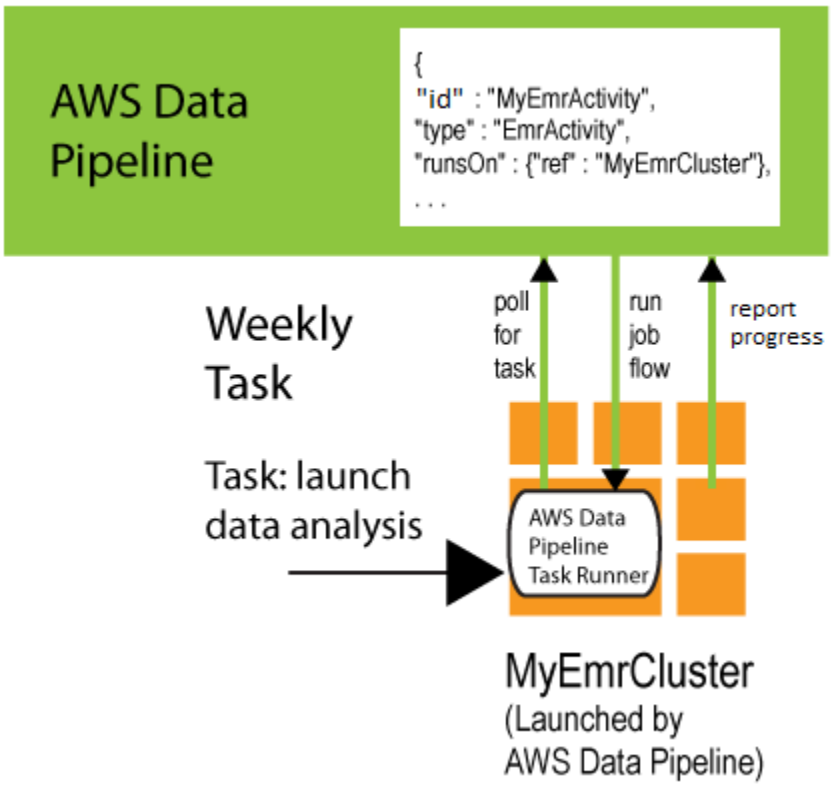
Trabalhando com o Task Runner

O Task Runner é um aplicativo de agente de tarefas que pesquisaAWS Data Pipeline tarefas agendadas e as executa em instâncias do Amazon EC2, clusters do Amazon EMR ou outros recursos computacionais, relatando o status da mesma. Dependendo do seu aplicativo, você pode optar pelo seguinte:

- PermitaAWS 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 campo de atividade runsOn é criado automaticamente. AWS Data Pipelinecuida da instalação do Task Runner em uma instância do EC2 ou no nó principal de um cluster do EMR. Nesse padrão, o AWS Data Pipeline pode fazer a maior parte do gerenciamento da instância ou do cluster para você.
- Executar todo o pipeline ou partes dele nos recursos que você gerencia. Os recursos potenciais incluem uma instância Amazon EC2 de longa duração, um cluster do Amazon EMR 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 seu próprio dispositivo) em praticamente qualquer lugar, desde que ele possa se comunicar com o serviçoAWS Data Pipeline web. Nesse padrão, você assume um controle quase total sobre quais recursos são usados e como eles são gerenciados, e você deve instalar e configurar manualmente o Task Runner. Para fazer isso, siga os procedimentos desta seção, conforme descrito em Executando o trabalho em recursos existentes usando o Task Runner (p. 279).

Executador de tarefasAWS Data Pipeline em recursos gerenciados

Quando um recurso é iniciado e gerenciado peloAWS Data Pipeline, o serviço web instala automaticamente o Task Runner nesse recurso para processar tarefas no pipeline. Você especifica um recurso computacional (uma instância do Amazon EC2 ou um cluster do Amazon EMR) para o`runs0n` campo de um objeto de atividade. AoAWS Data Pipeline iniciar esse recurso, ele instala o Task Runner nesse recurso e o configura para processar todos os objetos de atividade que tenham seus`runs0n` campos definidos para esse recurso. Quando o recurso éAWS Data Pipeline encerrado, os registros 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`. QuandoAWS Data Pipeline processa essa atividade, ele inicia um cluster do Amazon EMR e instala o Task Runner no nó principal. Em seguida, esse executor de tarefas processa as tarefas de atividades que têm seus`runsOn` campos definidos para esse`EmrCluster` objeto. 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 (p. 146).

Se você tiver vários recursosAWS Data Pipeline gerenciados em um pipeline, o Task Runner será instalado em cada um deles e todos eles pesquisarão as tarefasAWS Data Pipeline a serem processadas.

Executando o trabalho em recursos existentes usando o Task Runner

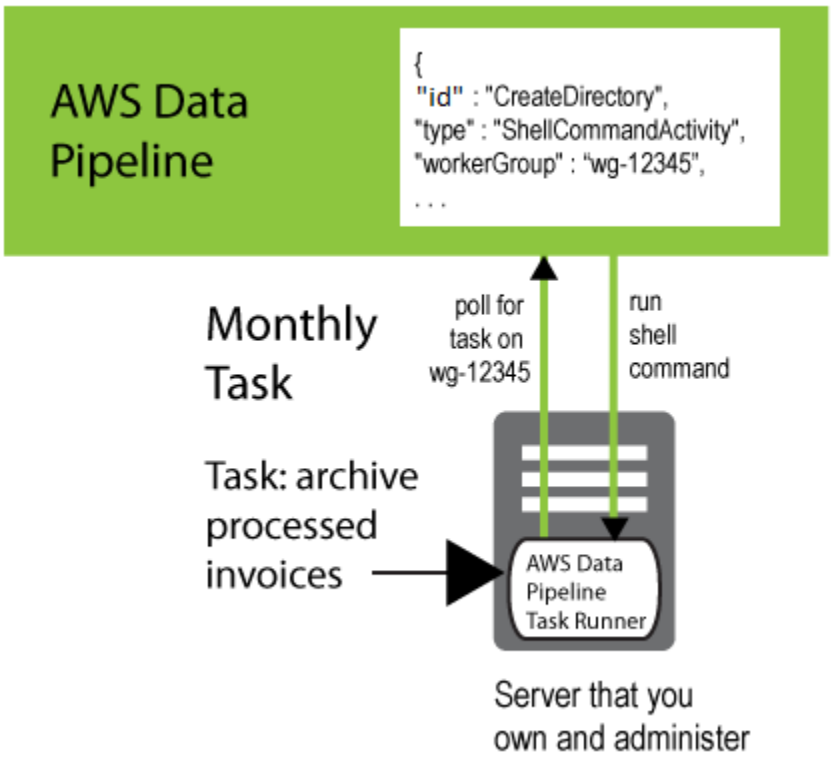
Você pode instalar o Task Runner em recursos computacionais que você gerencia, como uma instância do Amazon EC2 ou um servidor físico ou estação de trabalho. O Task Runner pode ser instalado em qualquer lugar, em qualquer hardware ou sistema operacional compatível, desde que possa se comunicar com o serviçoAWS Data Pipeline web.

Essa abordagem pode ser útil quando, por exemplo, você quiser usar o AWS Data Pipeline para processar dados armazenados no firewall da 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,AWS Data Pipeline pesquisar a próxima tarefa a ser executada. QuandoAWS Data Pipeline termina o processamento ou exclui o pipeline, a instância do Task Runner permanece em execução em seu recurso computacional até que você a desligue manualmente. Os registros do Task Runner persistem após a conclusão da execução do pipeline.

Para usar o Task Runner em um recurso que você gerencia, você deve primeiro baixar o Task Runner e depois instalá-lo em seu recurso computacional, usando os procedimentos desta seção.

> **Note**
>
> Você só pode instalar o Task Runner em Linux, UNIX ou macOS. O Task Runner não é compatível com o sistema operacional Windows.
> Para usar o Task Runner 2.0, a versão mínima do Java necessária é 1.7.

Para conectar um executor de tarefas que você instalou às atividades do pipeline que ele deve processar, adicione um`workerGroup` campo ao objeto e configure o Executador de Tarefas para pesquisar o valor desse grupo de trabalho. Você faz isso passando a string do grupo de trabalho como um parâmetro (por exemplo,`--workerGroup=wg-12345`) ao executar o arquivo JAR do Task Runner.




```
{
  "id" : "CreateDirectory",
  "type" : "ShellCommandActivity",
  "workerGroup" : "wg-12345",
  "command" : "mkdir new-directory"
}
```

Instalando o Runner de tarefas

Esta seção explica como instalar e configurar o Task Runner e seus pré-requisitos. A instalação é um processo manual simples.

Para instalar o Task Runner

1. O Task Runner requer as versões 1.6 ou 1.8 do Java. 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` em https://s3.amazonaws.com/datapipeline-us-eastTaskRunner -1/us-east-1/software/latest//TaskRunner-1.0.jar e copie-o em uma pasta no recurso

de computação de destino. Para clusters do Amazon EMR executando`EmrActivity` tarefas, instale o Task Runner no nó principal do cluster.

3. Ao usar o Task Runner para se conectar ao serviçoAWS 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 (p. 21).
4. O Task Runner se conecta ao serviçoAWS Data Pipeline web usando HTTPS. Se você estiver usando um recurso da AWS, verifique se o HTTPS esta habilitado na tabela de roteamento e na ACL de sub-rede apropriadas. Se você estiver usando um firewall ou proxy, verifique se a porta 443 está aberta.

(Opcional) Conceder acesso ao Task Runner ao Amazon RDS

O Amazon RDS permite controlar o acesso às suas instâncias de banco de dados usando grupos de segurança do banco de dados (grupos de segurança do 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 instâncias do Amazon RDS. O Task Runner obtém acesso ao Amazon RDS a partir da instância em que é executado, portanto, as contas e os grupos de segurança que você adiciona à sua instância do Amazon RDS dependem de onde você instala o Task Runner.

Para conceder acesso ao Runner de tarefas no EC2-Classic

1. Abra o console do Amazon RDS.
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, conforme descrito aqui:

   - `Ec2Resource`
     - Connection Type: `EC2 Security Group`

       Detalhes: *`my-security-group-name`*(o nome do grupo de segurança que você criou para a instância do EC2)
   - `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 seu computador estiver protegido por um firewall)
5. Clique em Add (Adicionar).

Para conceder acesso ao Task Runner no EC2-VPC

1. Abra o console do Amazon RDS.
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 console do Amazon EC2. 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 fonte depende de onde o Task Runner está sendo executado, conforme descrito aqui:

   - `Ec2Resource`
     - *`my-security-group-id`*(o ID do grupo de segurança que você criou para a instância do EC2)
   - `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 alcance do endereço IP da sua rede, se seu computador estiver protegido por um firewall)
5. Clique em Save (Salvar).

Iniciando o Runner de tarefas

Em uma nova janela do prompt de comando definida para o diretório em que você instalou o Task Runner, inicie o Task Runner com o seguinte comando.

```
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`--logUri` opção é usada para enviar seus registros compactados para um local no Amazon S3.

Quando o Task Runner está ativo, ele imprime o caminho para onde os arquivos de log são 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 (p. 283).

Verificando o registro do executor de tarefas

A maneira mais fácil de verificar se o Task Runner está funcionando é verificar se ele está gravando arquivos de log. O Task Runner grava arquivos de log de hora em hora no diretório`output/logs`,, sob o

diretório em que o Task Runner está instalado. O nome do arquivo é `Task Runner.log.YYYY-MM-DD-HH`, e HH vai de 00 a 23, em UDT. Para economizar espaço de armazenamento, todos os arquivos de log com mais de oito horas são compactados com o GZip.

Tópicos e condições prévias do Task Runner

O Task Runner usa um pool de threads para cada uma das tarefas, atividades e pré-condições. A configuração padrão de `--tasks` é 2. Isso significa que haverá dois threads alocados no grupo de tarefas e que cada thread pesquisará novas tarefas no serviço do AWS Data Pipeline. 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 repetição do pipeline para condições prévias acontece no Task Runner. Dois threads de pré-condição são alocados para pesquisar objetos de pré-condição no AWS Data Pipeline. O Task Runner respeita os campos do objeto de pré-condição retryDelay e preconditionTimeout que você define nas pré-condições.

Em muitos casos, diminuir o tempo limite da pesquisa de precondição e o número de novas tentativas ajuda a melhorar o desempenho do seu aplicativo. Da mesma forma, os aplicativos com precondições prolongadas podem precisar ter o tempo limite e os valores de novas tentativas aumentados. Para obter mais informações objetos de precondição, consulte Precondições (p. 15).

Opções de configuração do Task Runner

Essas são as opções de configuração disponíveis na linha de comando quando você inicia o Task Runner.

Parâmetro da linha de comando	Descrição
`--help`	Ajuda da linha de comando. Exemplo: `Java -jar TaskRunner-1.0.jar --help`
`--config`	O caminho e o nome do seu arquivo `credentials.json`.
`--accessId`	Seu ID de chave deAWS acesso para o Task Runner usar ao fazer solicitações. As`--secretKey` opções`--accessID` e fornecem uma alternativa ao uso de um arquivo credentials.json. Se um arquivo `credentials.json` também for fornecido, as opções `--accessID` e `--secretKey` terão prioridade.
`--secretKey`	Sua chaveAWS secreta para o Task Runner usar ao fazer solicitações. Para obter mais informações, consulte `--accessID`.
`--endpoint`	Um endpoint é uma URL que é o ponto de entrada para um serviço da Web. O endpoint de serviço do AWS Data Pipeline na região em que você está fazendo as solicitações. Opcional. De modo geral, especificar uma região é suficiente e não há necessidade de definir o endpoint. Para uma lista de regiões e endpoints do AWS Data Pipeline,

Parâmetro da linha de comando	Descrição
	consulte Regiões e endpoints do AWS Data Pipeline no Referência geral da AWS.
`--workerGroup`	O nome do grupo de operadores para o qual o executor de tarefas recupera o trabalho. Obrigatório. Quando o Task Runner pesquisa o serviço web, ele usa as credenciais que você forneceu e o valor de`workerGroup` para selecionar quais tarefas (se houver) recuperar. Você pode usar qualquer nome que seja significativo para você; o único requisito é que a sequência de caracteres corresponda entre o Task Runner e suas atividades de pipeline correspondentes. O nome do grupo de operadores está vinculado a uma região. Mesmo que haja nomes de grupos de trabalho idênticos em outras regiões, o Task Runner sempre obtém tarefas da região especificada em`--region`.
`--taskrunnerId`	O ID do executor de tarefas a ser usado para informar o andamento. Opcional.
`--output`	O diretório do Task Runner para arquivos de saída de log. Opcional. Os arquivos de log são armazenados em um diretório local até serem enviados para o Amazon S3. Essa opção substituirá o diretório padrão.
`--region`	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 tarefas da região de serviço padrão,`us-east-1`. Outras regiões com suporte são: `eu-west-1`, `ap-northeast-1`, `ap-southeast-2`, `us-west-2`.
`--logUri`	O caminho de destino do Amazon S3 para o Task Runner fazer backup dos arquivos de log a cada hora. Quando o Task Runner é encerrado, os registros ativos no diretório local são enviados para a pasta de destino do Amazon S3.
`--proxyHost`	O host do proxy usado pelos clientes do Task Runner para se conectar aos serviços da AWS.
`--proxyPort`	Porta do host proxy usada pelos clientes do Task Runner para se conectar aos serviços da AWS.
`--proxyUsername`	O nome do usuário para o proxy.
`--proxyPassword`	A senha para o proxy.
`--proxyDomain`	O nome de domínio do Windows para o proxy NTLM.

Parâmetro da linha de comando	Descrição
`--proxyWorkstation`	O nome da estação de trabalho do Windows para o proxy NTLM.

Usar o Task Runner com um proxy

Se você estiver usando um host de proxy, poderá especificar a configuração dele ao chamar o Task Runner ou configurar a variável de ambiente, HTTPS_PROXY. A variável de ambiente utilizada com o Task Runner aceitará a mesma configuração usada na Interface da Linha de Comando da AWS.

Executador de tarefas e AMIs personalizadas

Quando você especifica um`Ec2Resource` objeto para seu pipeline,AWS Data Pipeline cria uma instância do EC2 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 uma AMI personalizada com o Task Runner e, em seguida, especificar a ID dessa AMI usando o`imageId` campo do`Ec2Resource` objeto. Para obter mais informações, consulte Ec2Resource (p. 201).

Uma AMI personalizada deve atender aos seguintes requisitosAWS Data Pipeline para usá-la com êxito no Task Runner:

- Crie a AMI na mesma região em que as instâncias serão executadas. Para obter mais informações, consulte Criar sua própria AMI no Guia do usuário do Amazon EC2 para instâncias do Linux.
- Verifique se que o tipo de instância que você planeja usar oferece suporte ao tipo de virtualização da AMI. Por exemplo, os tipos de instância I2 e G2 requerem uma AMI HVM. Já os tipos de instância T1, C1, M1 e M2 requerem uma AMI PV. Para obter mais informações, consulte Tipos de virtualização do Linux AMI no Guia do usuário do Amazon EC2 para instâncias do Linux.
- 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

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 precondição 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 Instance summary (Resumo da instância), selecione o triângulo ao lado da instância para ver seus detalhes adicionais e selecione Details (Detalhes), More... (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`).

Identificando o cluster do Amazon EMR que atende seu pipeline

Se um `EMRCluster` ou `EMRActivity` falhar e as informações de erro fornecidas pelo AWS Data Pipeline console não estiverem 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 registros que o Amazon EMR fornece para obter mais detalhes sobre os erros que ocorrem.

Para ver 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, pesquise um cluster com o valor InstanceParent correspondente em seu nome e escolha Debug.

   Note

   Para que o botão Depurar funcione, sua definição de pipeline deve ter definido a EmrActivity `enableDebugging` opção como `true` e a `EmrLogUri` opção como um caminho válido.
5. Agora que você sabe qual cluster do Amazon EMR contém o erro que causa 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 que pudesse 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 resultado de uma falha em cascata de uma de suas dependências, mas o componente provavelmente não foi a fonte original da falha.

`DEACTIVATING`

O gasoduto está sendo desativado.

`FAILED`

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

`FINISHED`

O componente concluiu o trabalho atribuído.

`INACTIVE`

O oleoduto foi desativado.

`PAUSED`

O componente foi pausado e não está funcionando no momento.

`PENDING`

O pipeline está pronto para ser ativado pela primeira vez.

`RUNNING`

O recurso está em execução e pronto para receber trabalho.

`SCHEDULED`

O recurso está programado para ser executado.

`SHUTTING_DOWN`

O recurso está sendo encerrado após concluir seu trabalho com êxito.

`SKIPPED`

O componente ignorou os intervalos de execução depois que o pipeline foi ativado usando um carimbo de data/hora posterior ao cronograma atual.

`TIMEDOUT`

O recurso ultrapassou o `terminateAfter` limite e foi interrompido. AWS Data Pipeline Depois que o recurso atingir esse status, AWS Data Pipeline ignora os `retryTimeout` valores `actionOnResourceFailureretryDelay`, e desse recurso. Esse status se aplica somente aos recursos.

`VALIDATING`

A definição do pipeline está sendo validada por. AWS Data Pipeline

`WAITING_FOR_RUNNER`

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

`WAITING_ON_DEPENDENCIES`

O componente está verificando se suas pré-condições padrão e configuradas 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 pipelines para criar arquivos de log em um local persistente, como no exemplo a seguir, em que você usa o `pipelineLogUri` campo no `Default` objeto de um pipeline para fazer com que todos os componentes do pipeline usem um local de log do Amazon S3 por padrão (você pode substituir isso configurando um local de log em um componente específico do pipeline).

Note

Por padrão, o Task Runner armazena seus registros em um local diferente, que pode estar indisponível quando o pipeline termina e a instância que executa o Task Runner é encerrada. Para obter mais informações, consulte Verificando o registro do executor de tarefas (p. 282).

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 do pipeline, o Task Runner cria uma cópia dos registros em seu diretório, com a mesma formatação e os mesmos nomes de arquivo descritos na seção anterior sobre registros do Task Runner.

Registros de etapas do Hadoop Job e do Amazon EMR

Com qualquer atividade baseada no HadoopHadoopActivity (p. 152), comoHiveActivity (p. 159), ou PigActivity (p. 171) você pode visualizar os registros de tarefas do Hadoop no local retornado no slot de tempo de execução,. hadoopJobLog EmrActivity (p. 146)tem seus próprios recursos de registro e esses registros são armazenados usando o local escolhido pelo Amazon EMR e retornados pelo slot de tempo de execução,. emrStepLog Para obter mais informações, consulte Exibir 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

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 do WorkerGroup atribuído ao componente do pipeline tem exatamente o mesmo nome e maiúsculas e minúsculas do valor do WorkerGroup que você configurou para o Task Runner.

> Note
>
> Se você fornecer um valor de runsOn e workerGroup existir, workerGroup será ignorado.

Outra causa potencial desse problema é que o endpoint e a chave de acesso fornecidos ao Task Runner não são os mesmos do AWS Data Pipeline console ou do computador em que as ferramentas da AWS Data Pipeline CLI estão instaladas. Você pode ter criado novos pipelines sem erros visíveis, mas o Task Runner pesquisa o local errado devido à diferença de credenciais ou pesquisa o local correto sem permissões suficientes 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 precondições iniciais do seu pipeline foram atendidas. Se as precondições 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 `InputData` objeto tem uma pré-condição 'Pronto' especificando que os dados devem existir antes que o InputData objeto seja concluído. Se os dados não existirem, o InputData objeto permanecerá no `WAITING_ON_DEPENDENCIES` estado, aguardando que os dados especificados pelo campo do caminho se tornem disponíveis. Quaisquer objetos que dependam da InputData mesma forma permanecem em um `WAITING_ON_DEPENDENCIES` estado esperando que o InputData objeto alcance o `FINISHED` estado.

```
{
    "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 anterior, se as informações no campo de credenciais não tivessem permissão para acessar os dados especificados no campo de caminho, o InputData objeto ficaria preso em um `WAITING_ON_DEPENDENCIES` estado porque não pode acessar os dados especificados pelo campo de caminho, mesmo que esses dados existam.

Também é possível que um recurso que se comunica 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 corretamente as datas em seus objetos de agenda e se os endDateTime valores startDateTime e estão no formato UTC, como no exemplo a seguir:

```
{
    "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 precondições 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 precondições 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, políticas e relações de confiança do IAM conforme descrito emFunções do IAM para o AWS Data Pipeline (p. 67).

Permissões insuficientes para acessar recursos

As permissões que você define nas funções do IAM determinam se você AWS Data Pipeline pode acessar seus clusters do EMR e instâncias do EC2 para executar seus pipelines. Além disso, o IAM fornece o conceito de relações de confiança que vão além para permitir a criação de 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 AWS Data Pipeline não pode, verifique suas funções, políticas e relações de confiança do IAM, conforme descrito emFunções do IAM para o AWS Data Pipeline (p. 67).

Código de status: 400 Código de erro: PipelineNotFoundException

Esse erro significa que suas funções padrão do IAM podem não ter as permissões necessárias AWS Data Pipeline para funcionar corretamente. Para obter mais informações, consulte Funções do IAM para o AWS Data Pipeline (p. 67).

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

Esse erro significa que o Task Runner não pôde 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 registros 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: O usuário: arn:aws:sts: :xxxxxxxxxxxx:federated-user/i-xxxxxxxx não está autorizado a realizar: datapipeline:. PollForTask

  Note

  Nessa mensagem de erro, PollForTask pode ser substituído por nomes de outras AWS Data Pipeline permissões.

Essa mensagem de erro indica que a função do IAM que você especificou precisa de permissões adicionais necessárias para interagirAWS Data Pipeline. Certifique-se de que sua política de função do IAM contenha as seguintes linhas, substituídas pelo nome da permissão que você deseja adicionar (use * para conceder todas as permissões). PollForTask Para obter mais informações sobre como criar uma nova função do IAM e aplicar uma política a ela, consulte Gerenciando políticas do IAM no guia Como usar o IAM.

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

AMIs mais antigas do Amazon EMR podem criar dados falsos para arquivos CSV grandes

No Amazon EMR, as AMIs anteriores à 3.9 (3.8 e abaixo) AWS Data Pipeline usam um personalizado InputFormat para ler e gravar arquivos CSV para uso com trabalhos. MapReduce Isso é usado quando o serviço organiza tabelas de e para o Amazon S3. InputFormatFoi descoberto um problema em que a leitura de registros de arquivos CSV grandes pode resultar na produção de tabelas que não são copiadas corretamente. Esse problema foi corrigido em versões posteriores do Amazon EMR. Use o Amazon EMR AMI 3.9 ou uma versão 4.0.0 ou superior do Amazon EMR.

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 (p. 294). 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

Os seguintes limites aplicam-se a uma única conta da AWS. Se você precisar de capacidade adicional, você pode usar o formulário de solicitação do Centro de Support 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
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

Atributo	Limite	Ajustável
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 aosAWS Data Pipeline agentes que chamam a API do 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 normal 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 trabalhador, o Task Runner, repete automaticamente as chamadas de API que falham com uma exceção de limitação. O Task Runner tem um recuo para que as tentativas subsequentes de chamar a API ocorram em intervalos cada vez maiores. 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

API	Limite de taxa regular	Limite de intermitência
ReportTaskProgress	10 chamadas por segundo	100 chamadas
SetTaskStatus	10 chamadas por segundo	100 chamadas
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 configurarAWS Data Pipeline a criação automática de um cluster Amazon EMR de 20 nós para processar dados e suaAWS conta tiver um limite de instâncias 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.

Se você precisar de capacidade adicional, você pode usar o formulário de solicitação do Centro de Support 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 PipelineInformações sobre o produto: a página Web principal para obter informações sobreAWS Data Pipeline.
- AWS Data PipelinePerguntas frequentes técnicas — Abrange as 20 principais perguntas que os desenvolvedores fazem sobre esse produto.
- Notas de versão — Forneça 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 ao Amazon Web Services.

- Aulas e workshops: links para cursos de especialidades e baseados em função, além de laboratórios autoguiados para ajudar a aperfeiçoar suasAWS habilidades na e a obter experiência prática.
- AWSCentro do Desenvolvedor da: explore tutoriais, baixe ferramentas e informe-se sobre eventos paraAWS desenvolvedores da.
- AWSFerramentas do desenvolvedor da - Links para ferramentas de desenvolvedor, SDKs, toolkits de IDE e ferramentas da linha de comando para desenvolver e gerenciarAWS aplicações da.
- Centro de recursos de conceitos básicos: saiba como configurar aConta da AWS, participar daAWS comunidade da e iniciar sua primeira aplicação.
- Tutoriais práticos: siga os step-by-step tutoriais para iniciar sua primeira aplicação naAWS.
- AWSWhitepapers da: links para uma lista abrangente deAWS whitepapers técnicos da que abrangem tópicos, como arquitetura, segurança e economia, elaborados pelos arquitetos deAWS soluções da 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 principal página da Web para obter informações sobre oAWS Support one-on-one, um canal de suporte de resposta rápida para ajudar a construir e a executar aplicações 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
Documentação adicionada para executar determinados procedimentos usando a AWS CLI. Procedimentos relacionados ao AWS Data Pipeline console foram removidos.	Para obter mais informações, consulte Clonar o pipeline (p. 45), Visualizar logs de pipeline (p. 43) e Crie um pipeline a partir de modelos do Data Pipeline usando a CLI (p. 26).	26 de maio de 2023
Adicionou mais conteúdo e amostras para migrar AWS Data Pipeline para outros serviços alternativos.	Atualizou o tópico de migração AWS Data Pipeline para o AWS Step Functions ou para o Amazon MWAA com mais informações sobre cada alternativa, mapeamentos conceituais entre os serviços e amostras. AWS Glue Para obter mais informações, consulte Migração de cargas de trabalho do AWS Data Pipeline (p. 2).	31 de março de 2023
Foram adicionadas informações sobre o AWS Data Pipeline suporte do IMDSv2.	AWS Data Pipelinesuporta IMDSv2 para recursos do Amazon EMR e do Amazon EC2. Para obter mais informações, consulte Proteção de dados no AWS Data Pipeline (p. 60), EmrCluster (p. 208) e Ec2Resource (p. 201).	16 de dezembro de 2022
Foi adicionado um tópico para migrar AWS Data Pipeline para outros serviços alternativos.	Agora existem outros AWS serviços que oferecem aos clientes uma melhor experiência de integração de dados. Você pode migrar casos de uso típicos do AWS Data Pipeline AWS Step Functions ou do Amazon MWAA. AWS Glue Para obter mais informações, consulte Migração de cargas de trabalho do AWS Data Pipeline (p. 2).	16 de dezembro de 2022
Atualizou as listas de instâncias compatíveis do Amazon EC2 e do Amazon EMR. Atualização da lista de IDs de AMIs de HVM (Hardware Virtual Machine) usadas para as instâncias.	Atualizou as listas de instâncias compatíveis do Amazon EC2 e do Amazon EMR. Para obter mais informações, consulte Tipos de instância com suporte para as atividades de trabalho do pipeline (p. 7). Atualização da lista de IDs de AMIs de HVM (Hardware Virtual Machine) usadas para as instâncias. Para obter mais informações, consulte Sintaxe (p. 202) e pesquise `imageId`.	9 de novembro de 2018
Configuração adicionada para anexar volumes do Amazon EBS aos nós do cluster e para lançar um cluster do	Adição de opções de configuração a um objeto `EMRcluster`. Você pode usar essas opções em pipelines que usam clusters do Amazon EMR. Use os `TaskEbsConfiguration` campos `coreEbsConfigurationmasterEbsConfiguration`, e para configurar a anexação dos volumes do Amazon	19 de abril de 2018

Alteração	Descrição	Data de lançamento
Amazon EMR em uma sub-rede privada.	EBS aos nós principais, principais e de tarefas no cluster do Amazon EMR. Para obter mais informações, consulte Anexe os volumes do EBS aos nós de cluster (p. 226). Use os `ServiceAccessSecurityGroupId` campos `emrManagedMasterSecurityGroupIdemrManagedSlaveSecurityGroupId`, e para configurar um cluster do Amazon EMR em uma sub-rede privada. Para obter mais informações, consulte Configurar um cluster do Amazon EMR em uma sub-rede privada (p. 224). Para mais informações sobre sintaxe de `EMRcluster`, consulte EmrCluster (p. 208).	
Foi adicionada a lista de instâncias compatíveis do Amazon EC2 e do Amazon EMR.	Adição da lista de instâncias que o AWS Data Pipeline cria por padrão se você não especificar um tipo de instância na definição do pipeline. Foi adicionada uma lista de instâncias compatíveis do Amazon EC2 e do Amazon EMR. Para obter mais informações, consulte Tipos de instância com suporte para as atividades de trabalho do pipeline (p. 7).	22 de março de 2018
Adição do suporte aos pipelines sob demanda	• Mais suporte aos pipelines sob demanda, o que permite que você execute novamente um pipeline ao reativá-lo.	22 de fevereiro de 2016
Suporte adicional para bancos de dados do RDS	• `rdsInstanceId`, `region` e `jdbcDriverJarUri` adicionados a RdsDatabase (p. 250). • `database` atualizado em SqlActivity (p. 196) para oferecer suporte a `RdsDatabase` também.	17 de agosto de 2015
Suporte adicional a JDBC	• `database` atualizado em SqlActivity (p. 196) para oferecer suporte a `JdbcDatabase` também. • `jdbcDriverJarUri` adicionado a JdbcDatabase (p. 249) • `initTimeout` adicionado a Ec2Resource (p. 201) e EmrCluster (p. 208). • `runAsUser` adicionado a Ec2Resource (p. 201).	7 de julho de 2015
HadoopActivity, zona de disponibilidade e suporte pontual	• Suporte adicionado para enviar trabalhos paralelos aos clusters do Hadoop. Para obter mais informações, consulte HadoopActivity (p. 152). • Capacidade de solicitar instâncias spot com Ec2Resource (p. 201) e EmrCluster (p. 208). • Capacidade de iniciar recursos `EmrCluster` 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 (p. 46).	7 de abril de 2015

Alteração	Descrição	Data de lançamento
Modelos e console atualizados	Novos modelos adicionados. Atualizou o capítulo Introdução para usar o ShellCommandActivity modelo Introdução. Para obter mais informações, consulte Crie um pipeline a partir de modelos do Data Pipeline usando a CLI (p. 26).	25 de novembro de 2014
Suporte à VPC	Suporte adicionado para a iniciar recursos em uma nuvem privada virtual (VPC).	12 de março de 2014
Suporte de região	Suporte adicionado para várias regiões de serviços. Além da região `us-east-1`, o AWS Data Pipeline é compatível em `eu-west-1`, `ap-northeast-1`, `ap-southeast-2` e `us-west-2`.	20 de fevereiro de 2014
Suporte a Amazon Redshift	Foi adicionado suporte para o Amazon Redshift emAWS Data Pipeline, incluindo um novo modelo de console (Copiar para o Redshift) e um tutorial para demonstrar o modelo. Para obter mais informações, consulte Copie dados para o Amazon Redshift usando AWS Data Pipeline (p. 94), RedshiftDataNode (p. 125), RedshiftDatabase (p. 252) e RedshiftCopyActivity (p. 181).	6 de novembro de 2013
PigActivity	AdicionadoPigActivity, que fornece suporte nativo para o Pig. Para obter mais informações, consulte PigActivity (p. 171).	15 de outubro de 2013
Modelo, atividade e formato de dados novos do console	Foi adicionado o novo modelo de console do CrossRegion DynamoDB Copy, incluindo o novo HiveCopyActivity e o DynamoDB. ExportDataFormat	21 de agosto de 2013
Falhas e novas execuções em cascata	Informações adicionadas sobre o comportamento de falhas e novas execuções em cascata do AWS Data Pipeline. Para obter mais informações, consulte Falhas e novas execuções em cascata (p. 53).	8 de agosto de 2013
Vídeo sobre a solução de problemas	Vídeo adicionado sobre a solução de problemas básicos do AWS Data Pipeline. Para obter mais informações, consulte Solução de problemas (p. 286).	17 de julho de 2013
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 (p. 44).	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 (p. 51).	17 de junho de 2013
Status WAITING_ON_DEPENDENCIES	Status CHECKING_PRECONDITIONS alterado para WAITING_ON_DEPENDENCIES e adicionado o campo de tempo de execução @waitingOn para objetos do pipeline.	20 de maio de 2013
DynamoDB DataFormat	Modelo do DynamoDB DataFormat adicionado.	23 de abril de 2013

Alteração	Descrição	Data de lançamento
Vídeo Processar logs da web e suporte a instâncias spot	Apresentou o vídeo “Processe registros da Web com o AWS Data Pipeline, o Amazon EMR e o Hive” e o suporte para instâncias spot do Amazon EC2.	21 de fevereiro de 2013
	A versão inicial do Guia do desenvolvedor do AWS Data Pipeline.	20 de dezembro de 2012

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.