Entwicklerhandbuch

AWS Data Pipeline

API-Version 2012-10-29

AWS Data Pipeline: Entwicklerhandbuch

Table of Contents

Was ist AWS Data Pipeline?

> **Note**
>
> AWS Data PipelineDer Service befindet sich im Wartungsmodus und es sind keine neuen Funktionen oder Regionserweiterungen geplant. Weitere Informationen und Informationen zur Migration Ihrer vorhandenen Workloads finden Sie unterMigrieren von Workloads von AWS Data Pipeline.

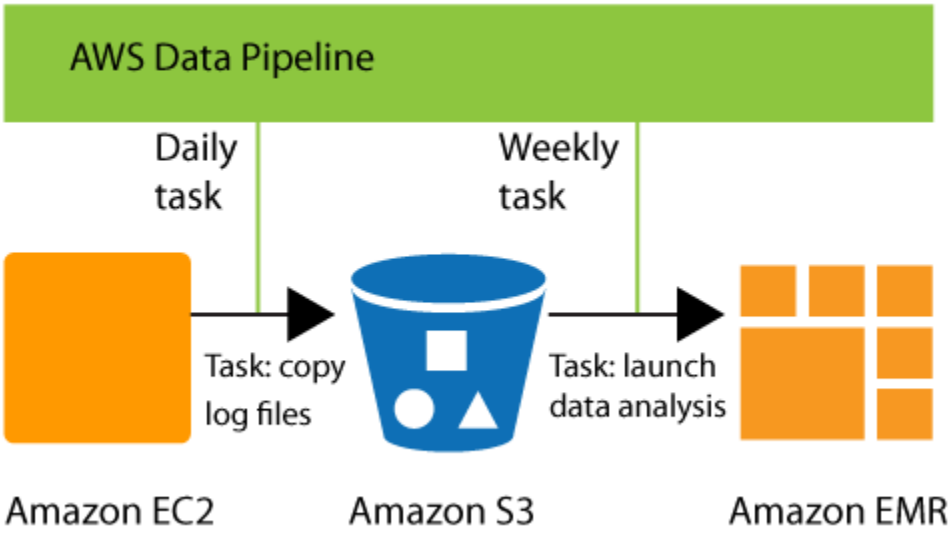
AWS Data Pipeline ist ein Web-Service, mit dem Sie das Verschieben und Transformieren von Daten automatisieren können. Sie können mit AWS Data Pipeline datengesteuerte Workflows erstellen und damit festlegen, dass bestimmte Aufgaben nur ausgeführt werden, wenn die vorherigen Aufgaben erfolgreich abgeschlossen wurden. Sie müssen nur die gewünschten Parameter für die Datentransformationen festlegen. AWS Data Pipeline setzt dann die konfigurierte Logik um.

Die folgenden Komponenten von AWS Data Pipeline sind an der Verwaltung der Daten beteiligt:

- Eine Pipeline-Definition legt die geschäftliche Logik der Datenverwaltung fest. Weitere Informationen finden Sie unter Syntax der Pipeline-Definitionsdatei.
- Eine Pipeline plant und führt Aufgaben aus, indem sie Amazon EC2-Instances erstellt, um die definierten Arbeitsaktivitäten auszuführen. Sie müssen nur die Pipeline-Definition in die Pipeline hochladen und diese anschließend aktivieren. Sie können auch die Pipeline-Definition einer gerade ausgeführten Pipeline bearbeiten. Sie müssen die Pipeline dann nur erneut aktivieren, damit die Änderungen wirksam werden. Außerdem können Sie die Pipeline deaktivieren, eine Datenquelle ändern und dann die Pipeline erneut aktivieren. Wenn Sie die Pipeline nicht mehr benötigen, können Sie sie löschen.
- Task Runner fragt nach Aufgaben ab und führt diese Aufgaben dann aus. Task Runner könnte beispielsweise Protokolldateien nach Amazon S3 kopieren und Amazon EMR-Cluster starten. Task Runner ist installiert und wird automatisch auf Ressourcen ausgeführt, die durch Ihre Pipeline-Definitionen erstellt wurden. Sie können eine benutzerdefinierte Task-Runner-Anwendung schreiben, oder Sie können die Task Runner-Anwendung verwenden, die von bereitgestellt wirdAWS Data Pipeline. Weitere Informationen finden Sie unter Task Runner.

Sie können AWS Data Pipeline beispielsweise die Protokolle Ihres Webservers täglich im Amazon Simple Storage Service (Amazon S3) archivieren und dann einen wöchentlichen Amazon EMR

(Amazon EMR) -Cluster über diese Protokolle ausführen, um Verkehrsberichte zu erstellen. AWS Data Pipelineplant die täglichen Aufgaben zum Kopieren von Daten und die wöchentliche Aufgabe zum Starten des Amazon EMR-Clusters. AWS Data Pipelinestellt außerdem sicher, dass Amazon EMR wartet, bis die Daten des letzten Tages auf Amazon S3 hochgeladen werden, bevor es mit der Analyse beginnt, selbst wenn es zu einer unvorhergesehenen Verzögerung beim Hochladen der Protokolle kommt.



Inhalt

Migrieren von Workloads von AWS Data Pipeline

AWShat den AWS Data Pipeline Dienst 2012 gestartet. Zu dieser Zeit suchten Kunden nach einem Service, der ihnen hilft, Daten mithilfe einer Vielzahl von Rechenoptionen zuverlässig zwischen verschiedenen Datenquellen zu verschieben. Jetzt gibt es andere Dienste, die den Kunden ein besseres Erlebnis bieten. Sie können es beispielsweise verwenden, AWS Glue um Apache Spark-Anwendungen auszuführen und zu orchestrieren, AWS Step Functions zur Orchestrierung von AWS

Servicekomponenten oder Amazon Managed Workflows for Apache Airflow (Amazon MWAA), um die Workflow-Orchestrierung für Apache Airflow zu verwalten.

In diesem Thema wird erklärt, wie Sie von AWS Data Pipeline alternativen Optionen migrieren. Welche Option Sie wählen, hängt von Ihrem aktuellen Workload abAWS Data Pipeline. Sie können typische Anwendungsfälle entweder AWS Data Pipeline AWS Glue zu AWS Step Functions oder Amazon MWAA migrieren.

Migrieren von Workloads zu AWS Glue

AWS Glue ist ein Serverless-Datenintegrationsdienst, der es Analytics-Benutzern erleichtert, Daten aus mehreren Quellen zu erkennen, vorzubereiten, zu verschieben und zu integrieren. Es umfasst Tools für die Erstellung, Ausführung von Jobs und Orchestrierung von Workflows. Mit AWS Glue können Sie mehr als 70 verschiedene Datenquellen entdecken und sich mit ihnen verbinden sowie Ihre Daten in einem zentralen Datenkatalog verwalten. Sie können ETL-Pipelines (Extract, Transform, Load) visuell erstellen, ausführen und überwachen, um Daten in Ihre Date Lakes zu laden. Außerdem können Sie mithilfe von Amazon Athena, Amazon EMR und Amazon Redshift Spectrum sofort katalogisierte Daten durchsuchen und abfragen.

Wir empfehlen, Ihren AWS Data Pipeline Workload auf folgende Zeiten zu AWS Glue migrieren:

- Sie suchen nach einem serverlosen Datenintegrationsservice, der verschiedene Datenquellen, Authoring-Schnittstellen wie visuelle Editoren und Notebooks sowie erweiterte Datenverwaltungsfunktionen wie Datenqualität und Erkennung vertraulicher Daten unterstützt.
- Ihr Workload kann zu AWS Glue Workflows, Jobs (in Python oder Apache Spark) und Crawlern migriert werden (Ihre bestehende Pipeline basiert beispielsweise auf Apache Spark).
- Sie benötigen eine einzige Plattform, die alle Aspekte Ihrer Datenpipeline abwickeln kann, einschließlich Aufnahme, Verarbeitung, Übertragung, Integritätstests und Qualitätsprüfungen.
- Ihre bestehende Pipeline wurde anhand einer vordefinierten Vorlage auf der AWS Data Pipeline Konsole erstellt, z. B. durch den Export einer DynamoDB-Tabelle nach Amazon S3, und Sie suchen nach derselben Vorlage.
- Ihr Workload hängt nicht von einer bestimmten Hadoop-Ökosystemanwendung wie Apache Hive ab.
- Ihr Workload erfordert keine Orchestrierung lokaler Server.

AWSberechnet einen sekundengenauen Stundensatz für Crawler (Datenerfassung) und ETL-Jobs (Verarbeitung und Laden von Daten). AWS Glue Studio ist eine integrierte Orchestrierungs-Engine

für AWS Glue Ressourcen und wird ohne zusätzliche Kosten angeboten. Weitere Informationen zur Preisgestaltung finden Sie unter AWS GluePreisgestaltung.

Migration von Workloads zu AWS Step Functions

AWSStep Functions ist ein serverloser Orchestrierungsdienst, mit dem Sie Workflows für Ihre geschäftskritischen Anwendungen erstellen können. Mit Step Functions verwenden Sie einen visuellen Editor, um Workflows zu erstellen und direkt in über 11.000 Aktionen für über 250 AWS Services wie AWS Lambda, Amazon EMR, DynamoDB und mehr zu integrieren. Sie können Step Functions verwenden, um Datenverarbeitungspipelines zu orchestrieren, Fehler zu behandeln und mit den Drosselgrenzwerten der zugrunde liegenden Dienste zu arbeiten. AWS Sie können Workflows erstellen, die Machine-Learning-Modelle verarbeiten und veröffentlichen, Mikroservices orchestrieren und AWS Dienste steuern, z. B. AWS Glue um ETL-Workflows (Extrahieren, Transformieren und Laden) zu erstellen. Sie können auch lang andauernde, automatisierte Workflows für Anwendungen erstellen, die menschliche Interaktion erfordern.

Ähnlich AWS Data Pipeline wie ist AWS Step Functions ein vollständig verwalteter Dienst, der von bereitgestellt wirdAWS. Sie müssen nicht die Infrastruktur verwalten, Patch-Worker, Betriebssystemversionsupdates oder ähnliches verwalten.

Wir empfehlen, Ihren AWS Data Pipeline Workload zu AWS Step Functions zu migrieren, wenn:

- Sie suchen nach einem serverlosen, hochverfügbaren Workflow-Orchestrierungsdienst.
- Sie suchen nach einer kostengünstigen Lösung, die nach der Granularität der Ausführung einer einzelnen Aufgabe abrechnet.
- Ihre Workloads orchestrieren Aufgaben für mehrere andere AWS Services wie Amazon EMR, Lambda oder DynamoDB. AWS Glue
- Sie suchen nach einer Low-Code-Lösung, die über einen drag-and-drop visuellen Designer für die Workflow-Erstellung verfügt und für die kein Erlernen neuer Programmierkonzepte erforderlich ist.
- Sie suchen nach einem Service, der Integrationen mit über 250 anderen AWS Diensten bietet, die über 11.000 Aktionen abdeckenout-of-the-box, sowie Integrationen mit benutzerdefinierten AWS Nichtdiensten und Aktivitäten ermöglicht.

AWS Data PipelineSowohl als auch Step Functions verwenden das JSON-Format, um Workflows zu definieren. Dies ermöglicht es, Ihre Workflows in der Quellcodeverwaltung zu speichern, Versionen zu verwalten, den Zugriff zu kontrollieren und mit CI/CD zu automatisieren. Step Functions

verwenden eine Syntax namens Amazon State Language, die vollständig auf JSON basiert und einen nahtlosen Übergang zwischen der textuellen und visuellen Darstellung des Workflows ermöglicht.

Mit Step Functions können Sie dieselbe Version von Amazon EMR auswählen, in AWS Data Pipeline der Sie gerade verwenden.

Für die Migration von Aktivitäten auf AWS Data Pipeline verwalteten Ressourcen können Sie die AWSSDK-Dienstintegration auf Step Functions verwenden, um die Bereitstellung und Bereinigung von Ressourcen zu automatisieren.

Für die Migration von Aktivitäten auf lokalen Servern, benutzerverwalteten EC2-Instances oder einem benutzerverwalteten EMR-Cluster können Sie einen SSM-Agenten auf der Instance installieren. Sie können den Befehl über den AWSSystems Manager Run-Befehl von Step Functions aus starten. Sie können die State Machine auch anhand des in Amazon definierten Zeitplans initiierenEventBridge.

AWSStep Functions hat zwei Arten von Workflows: Standard-Workflows und Express-Workflows. Für Standard-Workflows werden Ihnen Gebühren auf der Grundlage der Anzahl der Statusübergänge berechnet, die für die Ausführung Ihrer Anwendung erforderlich sind. Für Express-Workflows werden Ihnen die Gebühren auf der Grundlage der Anzahl der Anfragen für Ihren Workflow und seiner Dauer berechnet. Weitere Informationen zur Preisgestaltung finden Sie unter AWSStep Functions Pricing.

Migration von Workloads zu Amazon MWAA

Amazon MWAA (Managed Workflows for Apache Airflow) ist ein verwalteter Orchestrierungsservice für Apache Airflow, der es einfacher macht, durchgängige Datenpipelines in der Cloud in großem Maßstab einzurichten und zu betreiben. Apache Airflow ist ein Open-Source-Tool, mit dem Sequenzen von Prozessen und Aufgaben, die als „Workflows“ bezeichnet werden, programmgesteuert erstellt, geplant und überwacht werden. Mit Amazon MWAA können Sie die Programmiersprachen Airflow und Python verwenden, um Workflows zu erstellen, ohne die zugrunde liegende Infrastruktur aus Gründen der Skalierbarkeit, Verfügbarkeit und Sicherheit verwalten zu müssen. Amazon MWAA skaliert seine Workflow-Ausführungskapazität automatisch an Ihre Bedürfnisse und ist in AWS Sicherheitsservices integriert, um Ihnen einen schnellen und sicheren Zugriff auf Ihre Daten zu ermöglichen.

Ähnlich AWS Data Pipeline wie Amazon MWAA handelt es sich um vollständig verwaltete Dienste, die von bereitgestellt werden. AWS Sie müssen sich zwar mit einigen neuen Konzepten vertraut machen, müssen sich aber nicht mit der Verwaltung der Infrastruktur, Patchworkern, der Verwaltung von Betriebssystemversionsupdates oder ähnlichem befassen.

Wir empfehlen, Ihre AWS Data Pipeline Workloads zu Amazon MWAA zu migrieren, wenn:

- Sie suchen nach einem verwalteten, hochverfügbaren Dienst zur Orchestrierung von in Python geschriebenen Workflows.
- Sie möchten auf eine vollständig verwaltete, weit verbreitete Open-Source-Technologie, Apache Airflow, umsteigen, um maximale Portabilität zu erzielen.
- Sie benötigen eine einzige Plattform, die alle Aspekte Ihrer Datenpipeline abwickeln kann, einschließlich Aufnahme, Verarbeitung, Übertragung, Integritätstests und Qualitätsprüfungen.
- Sie suchen nach einem Service, der für die Orchestrierung von Datenleitungen entwickelt wurde und Funktionen wie eine umfangreiche Benutzeroberfläche für Observability, Neustarts für fehlgeschlagene Workflows, Backfills und Wiederholungsversuche für Aufgaben bietet.
- Sie suchen nach einem Service, der mehr als 800 vorgefertigte Operatoren und Sensoren umfasst und sowohl Dienstleistungen AWS als auch andere AWS Dienste abdeckt.

Amazon MWAA-Workflows werden mithilfe von Python als Directed Acyclic Graphs (DAGs) definiert, sodass Sie sie auch als Quellcode behandeln können. Mit dem erweiterbaren Python-Framework von Airflow können Sie Workflows erstellen, die mit praktisch jeder Technologie verbunden sind. Es verfügt über eine umfangreiche Benutzeroberfläche zur Anzeige und Überwachung von Workflows und kann problemlos in Versionskontrollsysteme integriert werden, um den CI/CD-Prozess zu automatisieren.

Mit Amazon MWAA können Sie dieselbe Version von Amazon EMR wählen, in der Sie derzeit verwenden. AWS Data Pipeline

AWSberechnet die Zeit, in der Ihre Airflow-Umgebung ausgeführt wird, zuzüglich zusätzlicher automatischer Skalierung, um mehr Worker- oder Webserverkapazität bereitzustellen. Weitere Informationen zur Preisgestaltung finden Sie unter Amazon Managed Workflows for Apache Airflow Pricing.

Abbildung der Konzepte

Die folgende Tabelle enthält eine Übersicht der wichtigsten Konzepte, die von den Diensten verwendet werden. Es wird Personen, die mit Data Pipeline vertraut sind, helfen, die Step-Funktionen und die MWAA-Terminologie zu verstehen.

Data Pipeline	Glue	Step Functions	Amazon MWAA
Pipelines	Arbeitsabläufe	Arbeitsabläufe	Direkte Acrylgrafiken

Data Pipeline	Glue	Step Functions	Amazon MWAA
Pipelinedefinition JSON	Workflow-Definition oder Python-basierte Blueprints	Amazon State Language JSON	Python-basiert
Aktivitäten	Aufträge	Staaten und Aufgaben	Aufgaben (Operatoren und Sensoren)
Instances	Job läuft	Hinrichtungen	DAG läuft
Attempts	Versuche erneut	Fänger und Retrier	Wiederholungen
Zeitplan für die Pipeline	Trigger planen	EventBridgeScheduler-Aufgaben	Cron, Fahrpläne, datenbewusst
Pipeline-Ausdrücke und -Funktionen	Blueprint-Bibliothek	Step Functions, intrinsische Funktionen und Lambda AWS	Erweiterbares Python-Framework

Beispiele

In den folgenden Abschnitten werden öffentliche Beispiele aufgeführt, auf die Sie zurückgreifen können, um von einzelnen Diensten AWS Data Pipeline zu migrieren. Sie können sie als Beispiele verwenden und Ihre eigene Pipeline für die einzelnen Dienste erstellen, indem Sie sie auf der Grundlage Ihres Anwendungsfalls aktualisieren und testen.

AWS Glue-Beispiele

Die folgende Liste enthält Beispielimplementierungen für die häufigsten AWS Data Pipeline Anwendungsfälle mit. AWS Glue

- Spark-Jobs ausführen
- Daten von JDBC nach Amazon S3 kopieren (einschließlich Amazon Redshift)
- Daten von Amazon S3 nach JDBC kopieren (einschließlich Amazon Redshift)
- Daten von Amazon S3 nach DynamoDB kopieren
- Verschieben von Daten zu und von Amazon Redshift
- Kontoübergreifender regionsübergreifender Zugriff auf DynamoDB-Tabellen

AWSBeispiele für Step Functions

Die folgende Liste enthält Beispielimplementierungen für die häufigsten AWS Data Pipeline Anwendungsfälle mit AWS Step Functions.

- Einen Amazon EMR-Job verwalten
- Ausführen eines Datenverarbeitungsauftrags auf Amazon EMR Serverless
- Hive/Pig/Hadoop-Jobs ausführen
- Abfragen großer Datensätze (Amazon Athena, Amazon S3,) AWS Glue
- ETL-Workflows mit Amazon Redshift ausführen
- Orchestrierung von Crawlern AWS Glue

Sehen Sie sich zusätzliche Tutorials und Beispielprojekte zur Verwendung von AWS Step Functions an.

Amazon MWAA-Beispiele

Die folgende Liste enthält Beispielimplementierungen für die häufigsten AWS Data Pipeline Anwendungsfälle mit Amazon MWAA.

- Einen Amazon EMR-Job ausführen
- Ein benutzerdefiniertes Plugin für Apache Hive und Hadoop erstellen
- Daten von Amazon S3 nach Redshift kopieren
- Ausführen eines Shell-Skripts auf einer Remote-EC2-Instance
- Orchestrierung hybrider (lokaler) Workflows

Weitere Tutorials und Beispielprojekte zur Verwendung von Amazon MWAA finden Sie hier.

Zugehörige Services

AWS Data Pipeline arbeitet zum Speichern von Daten mit den folgenden Services zusammen.

- Amazon DynamoDB — Bietet eine vollständig verwaltete NoSQL-Datenbank mit schneller Leistung zu geringen Kosten. Weitere Informationen finden Sie im Amazon DynamoDB Developer Guide.

- Amazon RDS — Stellt eine vollständig verwaltete relationale Datenbank bereit, die auf große Datensätze skaliert werden kann. Weitere Informationen finden Sie im Amazon Relational Database Service Developer Guide.
- Amazon Redshift — Bietet ein schnelles, vollständig verwaltetes Data Warehouse im Petabyte-Bereich, mit dem sich riesige Datenmengen einfach und kostengünstig analysieren lassen. Weitere Informationen finden Sie im Amazon Redshift Database Developer Guide.
- Amazon S3 — Bietet sicheren, dauerhaften und hochgradig skalierbaren Objektspeicher. Weitere Informationen finden Sie im Amazon Simple Storage Service-Benutzerhandbuch.

AWS Data Pipeline arbeitet zum Transformieren von Daten mit den folgenden Datenverarbeitungs-Services zusammen.

- Amazon EC2 — Bietet anpassbare Rechenkapazität — buchstäblich Server in den Rechenzentren von Amazon —, die Sie zum Aufbau und Hosten Ihrer Softwaresysteme verwenden. Weitere Informationen finden Sie im Amazon EC2-Benutzerhandbuch für Linux-Instances.
- Amazon EMR — Macht es Ihnen einfach, schnell und kostengünstig, riesige Datenmengen auf Amazon EC2-Servern zu verteilen und zu verarbeiten, indem Sie ein Framework wie Apache Hadoop oder Apache Spark verwenden. Weitere Informationen finden Sie im Amazon EMR Developer Guide.

Zugriff auf AWS Data Pipeline

Sie können Ihre Pipelines über die folgenden Schnittstellen erstellen und verwalten:

- AWS Management Console— Stellt eine Weboberfläche bereit, über die Sie zugreifen könnenAWS Data Pipeline.
- AWS Command Line Interface(AWS CLI) — Stellt Befehle für eine Vielzahl von AWS-Services bereitAWS Data Pipeline, einschließlich und wird unter Windows, macOS und Linux unterstützt. Weitere Informationen zum Installieren von AWS CLI finden Sie unter AWS Command Line Interface. Eine Liste der von AWS Data Pipeline unterstützten Befehle finden Sie unter datapipeline.
- AWS SDKs – Bietet sprachspezifische APIs und übernimmt viele der Verbindungsdetails, wie zum Beispiel die Berechnung der Signaturen, die Verarbeitung des erneuten Absendens von Anforderungen und die Fehlerbehandlung. Weitere Informationen finden Sie unter AWS SDKs.

- Abfrage-API — Stellt APIs auf niedriger Ebene bereit, die Sie mithilfe von HTTPS-Anfragen aufrufen. Die Verwendung der Abfrage-API ist die direkteste Möglichkeit für den Zugriff auf AWS Data Pipeline. Allerdings müssen dann viele technische Abläufe, wie beispielsweise das Erzeugen des Hashwerts zum Signieren der Anforderung und die Fehlerbehandlung, in der Anwendung durchgeführt werden. Weitere Informationen finden Sie in der AWS Data Pipeline-API-Referenz.

Preisgestaltung

Mit Amazon Web Services bezahlen Sie nur für das, was Sie tatsächlich nutzen. Die Pipeline-Kosten bei AWS Data Pipeline basieren darauf, für wie oft Ihre Aktivitäten und Vorbedingungen zur Ausführung geplant und wo sie ausgeführt werden. Weitere Informationen finden Sie unter AWS Data Pipeline-Preisgestaltung.

Wenn Ihr AWS-Konto jünger als 12 Monate ist, sind Sie zur Nutzung des kostenlosen Kontingents berechtigt. Das kostenlose Kontingent umfasst drei Vorbedingungen mit geringer Häufigkeit und fünf Aktivitäten mit geringer Häufigkeit pro Monat. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.

Unterstützte Instance-Typen für Pipeline-Aktivitäten

Wenn eine Pipeline AWS Data Pipeline ausgeführt wird, kompiliert sie die Pipeline-Komponenten, um eine Reihe von umsetzbaren Amazon EC2-Instances zu erstellen. Jede Instance enthält alle Informationen, die zum Ausführen einer bestimmten Aufgabe benötigt werden. Der komplette Satz an Instances stellt die To-do-Liste der Pipeline dar. AWS Data Pipeline übergibt die Instances zur Verarbeitung an Task Runner.

EC2 Instances haben verschiedene Konfigurationen, die als Instance-Typen bezeichnet werden. Jeder Instance-Typ verfügt über eine andere CPU, Eingabe/Ausgabe und Speicherkapazität. Zusätzlich zum Instance-Typ für eine Aktivität können Sie verschiedene Kaufoptionen auswählen. Nicht alle Instance-Typen stehen in allen AWS-Regionen zur Verfügung. Wenn ein Instance-Typ nicht verfügbar ist, kann Ihre Pipeline möglicherweise nicht bereitgestellt werden oder wird bei der Bereitstellung eingefroren. Informationen zur Verfügbarkeit von Instances finden Sie auf der Amazon EC2-Preisseite. Öffnen Sie den Link für Ihre Instance-Kaufoptionen und filtern Sie nach Region, um zu sehen, ob ein Instance-Typ in dieser Region verfügbar ist. Weitere Informationen zu diesen Instance-Typen, -Familien und Virtualisierungstypen finden Sie unter Amazon EC2-Instances und Amazon Linux AMI Instance Type Matrix.

Die folgenden Tabellen beschreiben die Instance-Typen, die von AWS Data Pipeline unterstützt werden. Sie können AWS Data Pipeline damit Amazon EC2-Instances in jeder Region starten, auch in Regionen, in denen dies AWS Data Pipeline nicht unterstützt wird. Weitere Informationen zu den Regionen, in denen AWS Data Pipeline unterstützt wird, finden Sie unter Regionen und Endpunkte in AWS.

Inhalt

Amazon EC2-Standardinstanzen nach AWS-Region

Wenn Sie in Ihrer Pipeline-Definition keinen Instance-Typ angeben, startet AWS Data Pipeline standardmäßig eine Instance.

In der folgenden Tabelle sind die Amazon EC2-Instances aufgeführt, die standardmäßig in den Regionen AWS Data Pipeline verwendet werden, in denen sie AWS Data Pipeline unterstützt werden.

Name der Region	Region	Instance-Typ
USA Ost (Nord-Virginia)	us-east-1	m1.small
USA West (Oregon)	us-west-2	m1.small
Asien-Pazifik (Sydney)	ap-southeast-2	m1.small
Asien-Pazifik (Tokio)	ap-northeast-1	m1.small
EU (Irland)	eu-west-1	m1.small

In der folgenden Tabelle sind die Amazon EC2-Instances aufgeführt, die standardmäßig in den Regionen AWS Data Pipeline gestartet werden, in denen sie AWS Data Pipeline nicht unterstützt werden.

Name der Region	Region	Instance-Typ
USA Ost (Ohio)	us-east-2	t2.small
USA West (Nordkalifornien)	us-west-1	m1.small
Asien-Pazifik (Mumbai)	ap-south-1	t2.small
Asien-Pazifik (Singapur)	ap-southeast-1	m1.small
Asien-Pazifik (Seoul)	ap-northeast-2	t2.small
Kanada (Zentral)	ca-central-1	t2.small
EU (Frankfurt)	eu-central-1	t2.small
EU (London)	eu-west-2	t2.small
EU (Paris)	eu-west-3	t2.small
Südamerika (São Paulo)	sa-east-1	m1.small

Zusätzliche unterstützte Amazon EC2-Instances

Neben den Standard-Instances, die erstellt werden, wenn Sie in Ihrer Pipeline-Definition keinen Instance-Typ angeben, werden auch die folgenden Instances unterstützt.

In der folgenden Tabelle sind die Amazon EC2-Instances aufgeführt, die AWS Data Pipeline unterstützt werden und erstellt werden können, sofern angegeben.

Instance-Klasse	Instance-Typen
Allgemeine Zwecke	t2.nano | t2.micro | t2.small | t2.medium | t2.large
Für Datenverarbeitung optimiert	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

Instance-Klasse	Instance-Typen
RAM-optimiert	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
Speicheroptimiert	i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge | hs1.8xlarge | g2.2xlarge | g2.8xlarge | d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge

Unterstützte Amazon EC2-Instances für Amazon EMR-Cluster

In dieser Tabelle sind die Amazon EC2-Instances aufgeführt, die Amazon EMR-Cluster AWS Data Pipeline unterstützen und für diese erstellen können, sofern angegeben. Weitere Informationen finden Sie unter Unterstützte Instance-Typen im Amazon EMR Management Guide.

Instance-Klasse	Instance-Typen
Allgemeine Zwecke	m1.small | m1.medium | m1.large | m1.xlarge | m3.xlarge | m3.2xlarge
Für Datenverarbeitung optimiert	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
RAM-optimiert	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 |

Instance-Klasse	Instance-Typen
	r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge
Speicheroptimiert	h1.4xlarge | hs1.2xlarge | hs1.4xlarge| hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4large | i2.8xlarge | d2.xlarge | d2.2xlarge| d2.4xlarge | d2.8xlarge
Beschleunigtes Computing	g2.2xlarge | cg1.4xlarge

AWS Data Pipeline-Konzepte

Bevor Sie beginnen, lesen Sie die Informationen zu den zentralen Konzepten und Komponenten von AWS Data Pipeline.

Inhalt

Pipeline-Definition

Eine Pipeline-Definition beschreibt, wie Sie Ihre Geschäftslogik an AWS Data Pipeline übermitteln. Sie umfasst die folgenden Informationen:

- Namen, Speicherorte und Formate der Datenquellen
- Aktivitäten, mit denen die Daten transformiert werden
- Den Zeitplan für diese Aktivitäten
- Ressourcen für die Ausführung Ihrer Aktivitäten und Vorbedingungen
- Voraussetzungen, die erfüllt werden müssen, bevor die Aktivitäten geplant werden können
- Möglichkeiten zur Information über Statusänderungen beim Fortschreiten der Pipeline-Durchführung

Basierend auf Ihrer Pipeline-Definition bestimmt AWS Data Pipeline die Aufgaben, plant sie und weist sie den Task-Runner-Anwendungen zu. Wenn eine Aufgabe nicht erfolgreich abgeschlossen wurde, versucht AWS Data Pipeline die Aufgabe entsprechend Ihren Anweisungen erneut durchzuführen,

und weist ihr ggf. einen anderen Task Runner zu. Sie können die Pipeline so konfigurieren, dass Sie benachrichtigt werden, wenn die Aufgabe wiederholt fehlschlägt.

In Ihrer Pipeline-Definition können Sie beispielsweise angeben, dass die von Ihrer Anwendung generierten Protokolldateien jeden Monat im Jahr 2013 in einem Amazon S3-Bucket archiviert werden. AWS Data Pipelinewürde dann 12 Aufgaben erstellen, von denen jede Daten über einen Monat kopiert, unabhängig davon, ob der Monat 30, 31, 28 oder 29 Tage umfasste.

Eine Pipeline-Definition können Sie folgendermaßen erstellen:

- Grafisch über die AWS Data Pipeline-Konsole
- Textlich, indem Sie eine JSON-Datei in dem von der Befehlszeile verwendeten Format erstellen
- Programmgesteuert, indem Sie den Webservice mit einem der AWS-SDKs oder der AWS Data Pipeline-API aufrufen

Eine Pipeline-Definition kann die folgenden Komponententypen enthalten.

Pipeline-Komponenten

Datenknoten

Den Speicherort von Eingabedaten für eine Aufgabe oder der Speicherort, an dem die Ausgabedaten gespeichert werden sollen.

Aktivitäten

Eine Definition der Arbeit, die nach einem Zeitplan mit einer Datenverarbeitungsressource und (typischerweise) Eingabe- und Ausgabedatenknoten durchgeführt werden soll.

Vorbedingungen

Eine Bedingungsaussage, die erfüllt sein muss, damit eine Aktion ausgeführt werden kann.

Ressourcen

Die Datenverarbeitungsressource, die die Arbeit ausführt, die eine Pipeline definiert.

Aktionen

Eine Aktion, die ausgelöst wird, wenn bestimmte Bedingungen erfüllt sind, z. B. wenn eine Aktivität fehlschlägt.

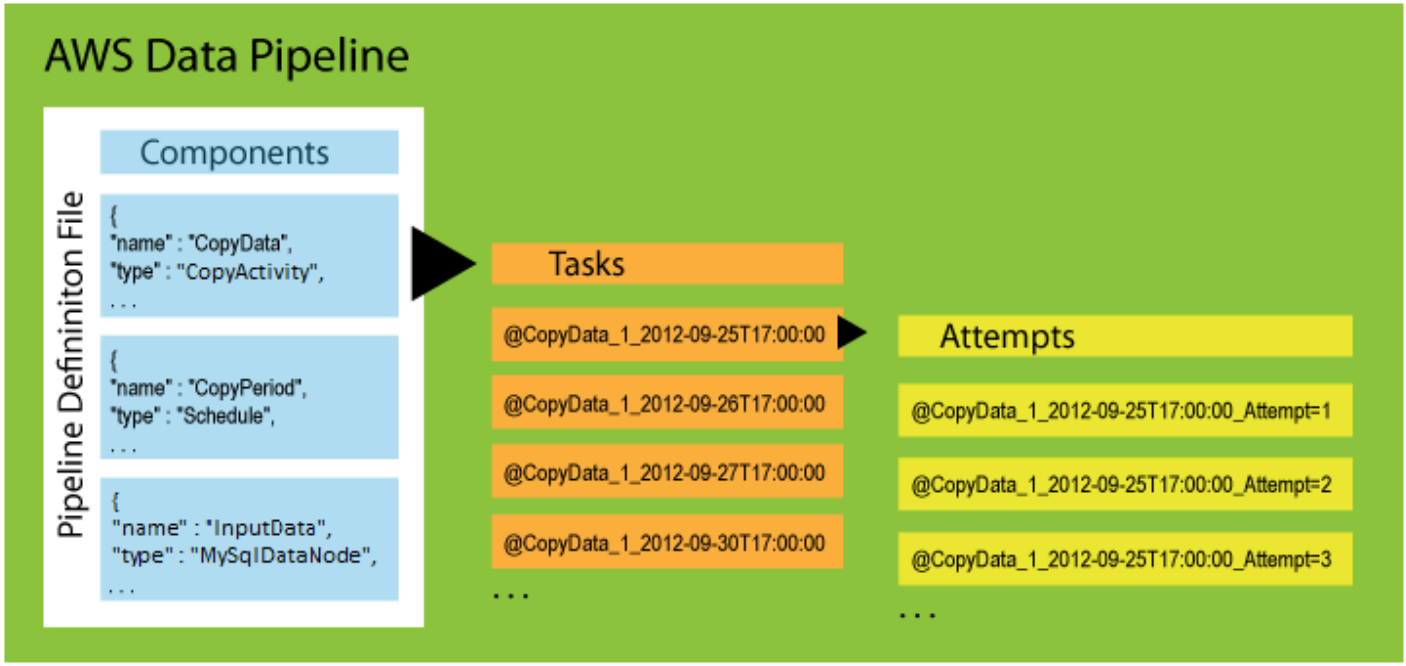
Weitere Informationen finden Sie unter Syntax der Pipeline-Definitionsdatei.

Pipeline-Komponenten, Instances und Versuche

Es gibt drei Komponententypen im Zusammenhang mit einer geplanten Pipeline:

- Pipeline-Komponenten — Pipeline-Komponenten stellen die Geschäftslogik der Pipeline dar und werden durch die verschiedenen Abschnitte einer Pipeline-Definition dargestellt. Pipeline-Komponenten geben die Datenquellen, Aktivitäten, den Zeitplan und die Vorbedingungen des Workflows an. Sie können Eigenschaften von übergeordneten Komponenten übernehmen. Beziehungen zwischen Komponenten werden durch Verweise definiert. Pipeline-Komponenten definieren die Regeln für die Datenverwaltung.
- Instances – Wenn AWS Data Pipeline eine Pipeline ausführt, werden die Pipeline-Komponenten zu einem Satz an umsetzbaren Instances zusammengestellt. Jede Instance enthält alle Informationen, die zum Ausführen einer bestimmten Aufgabe benötigt werden. Der komplette Satz an Instances stellt die To-do-Liste der Pipeline dar. AWS Data Pipeline übergibt die Instances zur Verarbeitung an Task Runner.
- Versuche – Um eine robuste Datenverwaltung sicherzustellen, wiederholt AWS Data Pipeline fehlgeschlagene Vorgänge. Diese Wiederholungen werden durchgeführt, bis die maximal erlaubte Anzahl an Wiederholungsversuchen erreicht ist. Versuchsobjekte verfolgen die einzelnen Versuche, Ergebnisse und ggf. Fehlergründe nach. Im Wesentlichen ist es die Instanz mit einem Zähler. AWS Data Pipelineführt Wiederholungsversuche durch und verwendet dieselben Ressourcen wie die vorherigen Versuche, z. B. Amazon EMR-Cluster und EC2-Instances.

> **Note**
>
> Das Wiederholen fehlgeschlagener Aufgaben ist ein wichtiger Bestandteil einer Fehlertoleranzstrategie. AWS Data Pipeline-Definitionen stellen Bedingungen und Schwellenwerte zur Steuerung der erneuten Versuche bereit. Zu viele erneute Versuche können jedoch dazu führen, dass unwiederbringliche Fehler zu spät erkannt werden, da AWS Data Pipeline Fehler erst dann meldet, wenn die festgelegte Anzahl an Wiederholungsversuchen erreicht wurde. Diese zusätzlichen Wiederholungen können für zusätzliche Gebühren sorgen, wenn sie auf AWS-Ressourcen ausgeführt werden. Daher sollten Sie genau abwägen, wann es sinnvoll ist, die AWS Data Pipeline-Standardeinstellungen für die Steuerung von erneuten Versuchen und zugehörige Einstellungen zu überschreiten.

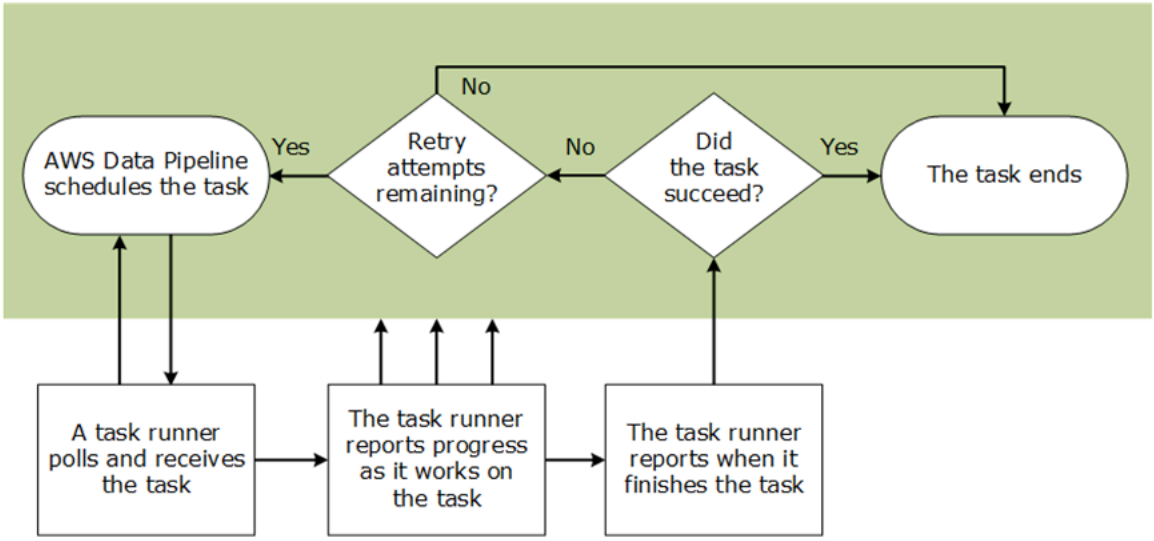


Task Runner

Ein Task Runner ist eine Anwendung, die AWS Data Pipeline-Abfragen nach Aufgaben durchführt und diese Aufgaben dann ausführt.

Task Runner ist eine Standardimplementierung eines Task Runners, der von bereitgestellt wirdAWS Data Pipeline. Wenn Task Runner installiert und konfiguriert ist, fragt es AWS Data Pipeline nach Aufgaben, die mit den von Ihnen aktivierten Pipelines verknüpft sind. Wenn Task Runner eine Aufgabe zugewiesen wird, führt er diese Aufgabe aus und meldet ihren Status anAWS Data Pipeline.

Das folgende Diagramm illustriert, wie AWS Data Pipeline und ein Task Runner interagieren, um eine geplante Aufgabe zu verarbeiten. Eine Aufgabe ist eine diskrete Arbeitseinheit, die der AWS Data Pipeline-Service mit einem Task Runner gemeinsam nutzt. Sie unterscheidet sich von einer Pipeline, die eine allgemeine Definition von Aktivitäten und Ressourcen ist, die normalerweise zu mehreren Aufgaben führt.



Es gibt zwei Möglichkeiten, Task Runner zu verwenden, um Ihre Pipeline zu verarbeiten:

- AWS Data Pipelineinstalliert Task Runner für Sie auf Ressourcen, die vom AWS Data Pipeline Webservice gestartet und verwaltet werden.
- Sie installieren Task Runner auf einer von Ihnen verwalteten Rechenressource, z. B. einer EC2-Instance mit langer Laufzeit oder einem lokalen Server.

Weitere Informationen zur Arbeit mit Task Runner finden Sie unter Arbeiten mit Task Runner.

Datenknoten

Bei AWS Data Pipeline definiert ein Datenknoten den Speicherort und den Typ der Daten, die eine Pipeline-Aktivität als Eingabe oder Ausgabe verwendet. AWS Data Pipeline unterstützt die folgenden Typen von Datenknoten:

DynamoDB DataNode

Eine DynamoDB-Tabelle, die Daten für HiveActivity oder EmrActivity zur Verwendung enthält.

SqlDataNode

Eine SQL-Tabellen- und Datenbankabfrage, die Daten zur Verwendung durch eine Pipeline-Aktivität repräsentiert.

> **Note**
>
> Bisher MySqlDataNode wurde verwendet. Verwenden Sie stattdessen SqlDataNode.

RedshiftDataNode

Eine Amazon Redshift-Tabelle, die Daten RedshiftCopyActivity zur Verwendung enthält.

S3 DataNode

Ein Amazon S3-Speicherort, der eine oder mehrere Dateien enthält, die für eine Pipeline-Aktivität verwendet werden können.

Datenbanken

AWS Data Pipeline unterstützt die folgenden Typen von Datenbanken:

JdbcDatabase

Eine JDBC-Datenbank.

RdsDatabase

Eine Amazon RDS-Datenbank.

RedshiftDatabase

Eine Amazon Redshift-Datenbank.

Aktivitäten

In AWS Data Pipeline ist eine Aktivität eine Pipeline-Komponente, die die durchzuführende Arbeit definiert. AWS Data Pipeline bietet mehrere vorkonfigurierte Aktivitäten für gängige Szenarien, z. B. das Verschieben von Daten von einem Speicherort an einen anderen, das Ausführen von Hive-Abfragen usw. Aktivitäten sind erweiterbar, sodass Sie Ihre eigenen benutzerdefinierten Skripts ausführen und so unzählige Kombinationen unterstützen können.

AWS Data Pipeline unterstützt die folgenden Typen von Aktivitäten:

CopyActivity

Kopiert Daten von einem Speicherort zu einem anderen.

EmrActivity

Führt einen Amazon EMR-Cluster aus.

HiveActivity

Führt eine Hive-Abfrage auf einem Amazon EMR-Cluster aus.

HiveCopyActivity

Führt eine Hive-Abfrage auf einem Amazon EMR-Cluster mit Unterstützung für erweiterte Datenfilterung und Unterstützung für S3 DataNode und aus. DynamoDB DataNode

PigActivity

Führt ein Pig-Skript auf einem Amazon EMR-Cluster aus.

RedshiftCopyActivity

Kopiert Daten in und aus Amazon Redshift-Tabellen.

ShellCommandActivity

Führt einen benutzerdefinierten UNIX/Linux-Shell-Befehl als Aktivität aus.

SqlActivity

Führt eine SQL-Abfrage auf einer Datenbank aus.

Einige Aktivitäten bieten spezielle Unterstützung für Staging-Daten und Datenbanktabellen. Weitere Informationen finden Sie unter Staging von Daten und Tabellen mit Pipeline-Aktivitäten.

Vorbedingungen

In AWS Data Pipeline ist eine Vorbedingung eine Pipeline-Komponente, die Bedingungsaussagen enthält, die erfüllt sein müssen, bevor eine Aktivität ausgeführt werden kann. Eine Vorbedingung kann beispielsweise überprüfen, ob Quelldaten vorhanden sind, bevor eine Pipeline-Aktivität versucht, sie zu kopieren. AWS Data Pipelinebietet mehrere vorgefertigte Vorbedingungen, die gängige Szenarien berücksichtigen, z. B. ob eine Datenbanktabelle vorhanden ist, ob ein Amazon S3-Schlüssel vorhanden ist usw. Vorbedingungen sind jedoch erweiterbar, sodass Sie Ihre eigenen benutzerdefinierten Skripts ausführen und so unzählige Kombinationen unterstützen können.

Es gibt zwei Arten von Vorbedingungen: vom System verwaltete Vorbedingungen und benutzerverwaltete Vorbedingungen. Vom System verwaltete Vorbedingungen werden vom AWS Data Pipeline-Webservice für Sie durchgeführt und erfordern keine Datenverarbeitungsressource. Benutzerverwaltete Vorbedingungen werden nur auf der Datenverarbeitungsressource ausgeführt, die Sie im Feld `runs0n` oder `workerGroup` festgelegt haben. Die `workerGroup`-Ressource ist von der Aktivität abgeleitet, die die Vorbedingung nutzt.

Vom System verwaltete Vorbedingungen

DynamoDB DataExists

Prüft, ob Daten in einer bestimmten DynamoDB-Tabelle vorhanden sind.

DynamoDB TableExists

Prüft, ob eine DynamoDB-Tabelle existiert.

S3 KeyExists

Prüft, ob ein Amazon S3-Schlüssel existiert.

S3 PrefixNotEmpty

Prüft, ob ein Amazon S3-Präfix leer ist.

Benutzerverwaltete Vorbedingungen

Vorhanden

Prüft, ob ein Datenknoten vorhanden ist.

ShellCommandPrecondition

Führt einen Unix-/Linux-Shell-Befehl als Vorbedingung aus.

Ressourcen

In AWS Data Pipeline ist eine Ressource die Datenverarbeitungsressource, die die Arbeit ausführt, die eine Pipeline-Aktivität festlegt. AWS Data Pipeline unterstützt die folgenden Ressourcentypen:

Ec2Resource

Eine EC2 Instance, welche die von einer Pipeline-Aktivität definierte Arbeit ausführt.

EmrCluster

Ein Amazon EMR-Cluster, der die durch eine Pipeline-Aktivität definierte Arbeit ausführt, wie EmrActivity z.

Ressourcen können in derselben Region mit ihrem Arbeitsdatensatz ausgeführt werden, auch in einer anderen Region als AWS Data Pipeline. Weitere Informationen finden Sie unter Verwenden einer Pipeline mit Ressourcen in mehreren Regionen.

Ressourcenlimits

AWS Data Pipeline kann skaliert werden, um eine große Anzahl von gleichzeitigen Aufgaben durchführen zu können. Sie können das System so konfigurieren, dass es automatisch die Ressourcen erstellt, die für die Verarbeitung großer Workloads erforderlich sind. Diese automatisch erstellten Ressourcen sind von Ihnen steuerbar und zählen bei Ihren Ressourcenlimits für Ihr AWS-Konto mit. Wenn Sie beispielsweise so konfigurieren, dass automatisch ein Amazon EMR-Cluster mit 20 Knoten erstellt wird, AWS Data Pipeline um Daten zu verarbeiten, und für Ihr AWS-Konto ein EC2-Instance-Limit von 20 festgelegt ist, können Sie versehentlich Ihre verfügbaren Backfill-Ressourcen erschöpfen. Daher sollten Sie diese Ressourceneinschränkungen bei Ihrem Design berücksichtigen oder Ihre Kontolimits entsprechend erweitern. Weitere Informationen zu Service Limits finden Sie unter AWS Service Limits in der Allgemeinen AWS-Referenz.

> **Note**
>
> Der Grenzwert ist eine Instance pro `Ec2Resource` Komponentenobjekt.

Unterstützte Plattformen

Pipelines können Ihre Ressourcen in den folgenden Plattformen starten:

EC2-Classic

Ihre Ressourcen werden in einem einzelnen, flachen Netzwerk ausgeführt, das Sie gemeinsam mit anderen Kunden verwenden.

EC2-VPC

Ihre Ressourcen werden in einer Virtual Private Cloud (VPC) ausgeführt, die logisch von Ihrem AWS-Konto isoliert ist.

Ihr AWS-Konto kann Ressourcen auf beiden Plattformen oder nur auf der EC2-VPC starten, je nach Region. Weitere Informationen finden Sie unter Unterstützte Plattformen im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Wenn Ihr AWS-Konto nur EC2-VPC unterstützt, erstellen wir eine Standard-VPC für Sie in jeder AWS-Region. Standardmäßig starten wir Ihre Ressourcen in einem Standard-Subnetz Ihrer Standard-VPC. Alternativ können Sie eine nicht standardmäßige VPC erstellen und bei der Konfiguration Ihrer Ressourcen eines ihrer Subnetze angeben. Dann starten wir Ihre Ressourcen in dem angegebenen Subnetz der nicht standardmäßigen VPC.

Wenn Sie eine Instance in einer VPC starten, müssen Sie eine Sicherheitsgruppe angeben, die speziell für diese VPC erstellt wurde. Wenn Sie eine Instance in einer VPC starten, können Sie keine Sicherheitsgruppe angeben, die Sie für EC2-Classic erstellt haben. Darüber hinaus müssen Sie die Sicherheitsgruppen-ID und nicht den Sicherheitsgruppennamen verwenden, um eine Sicherheitsgruppe für eine VPC festzulegen.

Amazon EC2-Spot-Instances mit Amazon EMR-Clustern und AWS Data Pipeline

Pipelines können Amazon EC2 Spot Instances für die Taskknoten in ihren Amazon EMR-Cluster-Ressourcen verwenden. Pipelines verwenden standardmäßig On-Demand-Instances. Mit Spot-Instances können Sie als Reserve vorhandene EC2-Instances verwenden und diese ausführen. Das Spot Instance-Preismodell ergänzt das On-Demand-Preismodell und das Reserved Instance-Preismodell und stellt womöglich die kosteneffizienteste Option für Rechenkapazität dar, je nach Anwendung. Weitere Informationen finden Sie auf der Produktseite zu Amazon EC2-Spot-Instances.

Wenn Sie Spot-Instances verwenden, AWS Data Pipeline übermittelt Amazon EMR den Höchstpreis für Ihre Spot-Instance, wenn Ihr Cluster gestartet wird. Es weist die Arbeit des Clusters automatisch der Anzahl an Spot-Instance-Aufgabenknoten zu, die Sie im Feld `taskInstanceCount` definiert haben. AWS Data Pipeline beschränkt Spot-Instances für Aufgabenknoten, um sicherzustellen, dass On-Demand-Core-Knoten verfügbar sind, die Ihre Pipeline ausführen.

Sie können eine fehlgeschlagene oder abgeschlossene Pipeline-Ressourcen-Instance bearbeiten und Spot-Instances hinzufügen. Wenn die Pipeline den Cluster erneut startet, nutzt er die Spot-Instances für die Aufgabenknoten.

Überlegungen zu Spot-Instances

Wenn Sie Spot-Instances mit AWS Data Pipeline verwenden, berücksichtigen Sie Folgendes:

- Ihre Spot-Instances können gekündigt werden, wenn der Spot-Instance-Preis Ihren Höchstpreis für die Instance übersteigt oder wenn Amazon EC2-Kapazitätsgründe vorliegen. Sie verlieren Ihre Daten jedoch nicht, da AWS Data Pipeline Cluster mit Core-Knoten nutzt, die immer On-Demand-Instances sind und nicht im Zusammenhang mit dem Beenden stehen.
- Spot-Instances können mehr Zeit zum Starten benötigen, weil sie Kapazität asynchron bereitstellen. Aus diesem Grund läuft eine Spot-Instance-Pipeline möglicherweise langsamer als eine äquivalente On-Demand-Instance-Pipeline.
- Ihr Cluster wird möglicherweise nicht ausgeführt, wenn Sie Ihre Spot-Instances nicht erhalten, beispielsweise, wenn Ihr Höchstpreis zu niedrig ist.

Aktionen

AWS Data Pipeline-Aktionen sind Schritte, die eine Pipeline-Komponente ausführt, wenn bestimmte Ereignisse eintreten, z. B. bei Erfolg, Fehlern oder verspäteten Aktivitäten. Das Ereignisfeld einer Aktivität bezieht sich auf eine Aktion, z. B. einen Verweis auf `snsalarm` im Feld `onLateAction` von `EmrActivity`.

AWS Data Pipelinestützt sich auf Amazon SNS-Benachrichtigungen als primäres Mittel, um den Status von Pipelines und ihren Komponenten unbeaufsichtigt anzuzeigen. Weitere Informationen finden Sie unter Amazon SNS. Zusätzlich zu den SNS-Benachrichtigungen können Sie über die AWS Data Pipeline-Konsole und -CLI Pipeline-Statusinformationen abrufen.

AWS Data Pipeline unterstützt die folgenden Aktionen:

SnsAlarm

Eine Aktion, die eine SNS-Benachrichtigung an ein Thema sendet, basierend auf den Ereignissen `onSuccess`, `OnFail` und `onLateAction`.

Beenden

Eine Aktion, die eine Stornierung von ausstehenden oder nicht abgeschlossenen Aktivitäten, Ressourcen oder Datenknoten auslöst. Sie können keine Aktionen beenden, die `onSuccess`, `OnFail` oder `onLateAction` beinhalten.

Proaktive Pipeline-Überwachung

Die beste Möglichkeit zum Erkennen von Problemen ist die proaktive Überwachung Ihrer Pipelines von Anfang an. Sie können Pipelinekomponenten so konfigurieren, dass Sie über bestimmte Situationen oder Ereignisse informiert werden, z. B. wenn eine Pipelinekomponente ausfällt oder nicht zu ihrer geplanten Startzeit startet. AWS Data Pipelineerleichtert die Konfiguration von Benachrichtigungen, indem Ereignisfelder auf Pipeline-Komponenten bereitgestellt werden, die Sie mit Amazon SNS-Benachrichtigungen verknüpfen können, wie `onSuccess`, `onFail`, und`onLateAction`.

Einrichten für AWS Data Pipeline

Bevor Sie AWS Data Pipeline zum ersten Mal verwenden können, müssen Sie die folgenden Aufgaben erledigen.

Aufgaben

- Registrieren Sie sich für AWS
- Erstellen Sie IAM-Rollen für Ressourcen AWS Data Pipeline und Pipeline-Ressourcen
- Erlauben Sie IAM-Prinzipalen (Benutzern und Gruppen), die erforderlichen Aktionen auszuführen
- Erteilen programmgesteuerten Zugriffs

Nachdem Sie diese Aufgaben durchgeführt haben, können Sie mit AWS Data Pipeline arbeiten. Ein Tutorial zum Einstieg finden Sie unter Erste Schritte mit AWS Data Pipeline.

Registrieren Sie sich für AWS

Bei der Registrierung für Amazon Web Services (AWS) wird Ihr AWS-Konto automatisch für alle Dienste in AWS einschließlich AWS Data Pipeline registriert. Berechnet werden Ihnen aber nur die Services, die Sie nutzen. Weitere Informationen zu den Nutzungsgebühren von AWS Data Pipeline finden Sie unter AWS Data Pipeline.

So melden Sie sich für ein AWS-Konto an

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

Anmeldung für ein AWS-Konto

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

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

   Wenn Sie sich für ein AWS-Konto anmelden, wird ein Root-Benutzer des AWS-Kontos erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und

verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

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

Einen Administratorbenutzer erstellen

Nachdem Sie sich für einen angemeldet habenAWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-KontosAWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Schützen Ihres Root-Benutzer des AWS-Kontos

1. Melden Sie sich bei AWS Management Console als Kontobesitzer an, indem Sie Stammbenutzer auswählen und Ihre AWS-Konto-E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-AnmeldungBenutzerhandbuch zu .
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

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

Erstellen eines Administratorbenutzers

1. Aktivieren Sie IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity CenterBenutzerhandbuch.
2. Gewähren Sie in IAM Identity Center einem Administratorbenutzer Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity CenterBenutzerhandbuch.

Als Administratorbenutzer anmelden

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

  Hilfe bei der Anmeldung mit einem IAM-Identity-Center-Benutzer finden Sie unter Anmelden beim AWS-Zugangsportal im AWS-Anmeldung Benutzerhandbuch zu.

Erstellen Sie IAM-Rollen für Ressourcen AWS Data Pipeline und Pipeline-Ressourcen

AWS Data Pipelineerfordert IAM-Rollen, die die Berechtigungen für die Ausführung von Aktionen und den Zugriff auf AWS Ressourcen festlegen. Die Pipeline-Rolle bestimmt die Berechtigungen, AWS Data Pipeline über die sie verfügt, und eine Ressourcenrolle bestimmt die Berechtigungen, über die Anwendungen verfügen, die auf Pipeline-Ressourcen wie EC2-Instances ausgeführt werden. Sie geben diese Rollen an, wenn Sie eine Pipeline erstellen. Auch wenn Sie keine benutzerdefinierte Rolle angeben und die Standardrollen `DataPipelineDefaultRole` verwenden`DataPipelineDefaultResourceRole`, müssen Sie zuerst die Rollen erstellen und Berechtigungsrichtlinien anhängen. Weitere Informationen finden Sie unter IAM-Rollen für AWS Data Pipeline.

Erlauben Sie IAM-Prinzipalen (Benutzern und Gruppen), die erforderlichen Aktionen auszuführen

Um mit einer Pipeline arbeiten zu können, muss ein IAM-Prinzipal (ein Benutzer oder eine Gruppe) in Ihrem Konto die erforderlichen AWS Data PipelineAktionen und Aktionen für andere Dienste ausführen dürfen, wie in Ihrer Pipeline definiert.

Um die Berechtigungen zu vereinfachen, können Sie die AWSDataPipeline_FullAccessverwaltete Richtlinie an IAM-Prinzipale anhängen. Diese verwaltete Richtlinie ermöglicht es dem Prinzipal, alle Aktionen auszuführen, die ein Benutzer benötigt, sowie die `iam:PassRole` Aktion für die Standardrollen, die verwendet werdenAWS Data Pipeline, wenn keine benutzerdefinierte Rolle angegeben ist.

Es wird dringend empfohlen, diese verwaltete Richtlinie sorgfältig zu prüfen und die Berechtigungen nur auf diejenigen zu beschränken, die Ihre Benutzer benötigen. Verwenden Sie bei Bedarf diese

Richtlinie als Ausgangspunkt und entfernen Sie dann die Berechtigungen, um eine restriktivere Inline-Berechtigungsrichtlinie zu erstellen, die Sie an IAM-Prinzipale anhängen können. Weitere Informationen und Beispiele für Berechtigungsrichtlinien finden Sie unter Beispielrichtlinien für AWS Data Pipeline

Eine Richtlinienanweisung, die dem folgenden Beispiel ähnelt, muss in einer Richtlinie enthalten sein, die jedem IAM-Prinzipal zugeordnet ist, der die Pipeline verwendet. Diese Anweisung ermöglicht es dem IAM-Prinzipal, die `PassRole` Aktion für die Rollen auszuführen, die eine Pipeline verwendet. Wenn Sie keine Standardrollen verwenden, ersetzen Sie *`MyPipelineRole`* und *`MyResourceRole`* durch die benutzerdefinierten Rollen, die Sie erstellen.

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

Das folgende Verfahren zeigt, wie Sie eine IAM-Gruppe erstellen, die AWSDataPipeline_FullAccessverwaltete Richtlinie an die Gruppe anhängen und dann Benutzer zur Gruppe hinzufügen. Sie können dieses Verfahren für jede Inline-Richtlinie verwenden

Um eine Benutzergruppe zu erstellen **`DataPipelineDevelopers`** und die AWSDataPipeline_FullAccessRichtlinie anzuhängen

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wechseln Sie im Navigationsbereich zu Groups und klicken Sie auf Create New Group.
3. Geben Sie beispielsweise einen Gruppennamen ein und wählen Sie dann Next Step aus. **`DataPipelineDevelopers`**
4. Geben Sie **`AWSDataPipeline_FullAccess`** Filter ein und wählen Sie ihn dann aus der Liste aus.
5. Wählen Sie Next Step (Nächster Schritt) und anschließend Create Group (Gruppe erstellen) aus.

6. Um Benutzer zur Gruppe hinzuzufügen:

   a. Wählen Sie die Gruppe, die Sie erstellt haben, aus der Gruppenliste aus.

   b. Wählen Sie Group Actions (Gruppenaktionen) und Add Users to Group (Benutzer zu Gruppe hinzufügen) aus.

   c. Wählen Sie die Benutzer, die Sie hinzufügen möchten, aus der Liste aus und klicken Sie dann auf Benutzer zur Gruppe hinzufügen.

Erteilen programmgesteuerten Zugriffs

Benutzer benötigen programmgesteuerten Zugriff, wenn sie außerhalb der AWS Management Console mit AWS interagieren möchten. Die Vorgehensweise, um programmgesteuerten Zugriff zu gewähren, hängt davon ab, welcher Benutzertyp auf zugreift AWS.

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	Verwenden Sie temporäre Anmeldeinformationen, um programmgesteuerte Anforderungen an die AWS CLI, AWS-SDKs oder AWS-APIs zu signieren.	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zur AWS CLI finden Sie unter Konfigurieren der AWS CLI für die Verwendung von AWS IAM Identity Center im AWS Command Line Interface-Benutzerhandbuch. • Informationen zu AWS-SDKs, Tools und AWS-APIs finden Sie unter IAM-Identity-Center-Authentifizierung im Referenzhandbuch zu AWS-SDKs und Tools.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmgesteuerte Anforderungen an die AWS CLI, AWS-SDKs oder AWS-APIs zu signieren.	Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS-Ressourcen im IAM-Benutzerhandbuch.
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmgesteuerte Anforderungen an die AWS CLI, AWS-SDKs oder AWS-APIs zu signieren.	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zur AWS CLI finden Sie unter Authentifizierung mit IAM-Benutzer-Anmeldeinformationen im AWS Command Line Interface-Benutzerhandbuch. • Informationen zu AWS-SDKs und Tools finden Sie unter Authentifizierung mit langfristigen Anmeldeinformationen im Referenzhandbuch zu AWS-SDKs und Tools. • Informationen zu AWS-APIs finden Sie unter Verwalten von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch.

Erste Schritte mit AWS Data Pipeline

Mit AWS Data Pipeline können Sie regelmäßige Arbeitslasten zur Datenverarbeitung sequenzieren, planen, ausführen und verwalten – zuverlässig und kosteneffizient. Dieser Service erleichtert Ihnen das Entwerfen von extract-transform-load (ETL-) Aktivitäten mithilfe strukturierter und unstrukturierter Daten, sowohl vor Ort als auch in der Cloud, auf der Grundlage Ihrer Geschäftslogik.

Um AWS Data Pipeline zu nutzen, erstellen Sie eine Pipeline-Definition, die die Geschäftslogik für die Datenverarbeitung festlegt. Eine typische Pipeline-Definition besteht aus Aktivitäten, die die auszuführende Arbeit definieren, und Datenknoten, die den Ort und Typ der Eingabe- und Ausgabedaten definieren.

In diesem Tutorial führen Sie ein Shell-Befehlsskript aus, das die Anzahl der GET-Anforderungen in Apache-Webserverprotokollen zählt. Diese Pipeline läuft eine Stunde lang alle 15 Minuten und schreibt bei jeder Iteration die Ausgabe in Amazon S3.

Voraussetzungen

Bevor Sie beginnen, führen Sie die Aufgaben in Einrichten für AWS Data Pipeline durch.

Pipeline-Objekte

Die Pipeline verwendet die folgenden Objekte:

ShellCommandActivity

Liest die Eingabeprotokolldatei und zählt die Anzahl an Fehlern.

S3 DataNode (Eingabe)

Der S3-Bucket, der die Eingabeprotokolldatei enthält.

S3 DataNode (Ausgabe)

Der S3-Bucket für die Ausgabe.

Ec2Resource

Die Datenverarbeitungsressource, mit der AWS Data Pipeline die Aktivität ausführt.

Hinweis: Wenn Sie eine große Menge an Protokolldateidaten haben, können Sie Ihre Pipeline so konfigurieren, dass zum Verarbeiten der Dateien ein EMR-Cluster anstelle einer EC2 Instance verwendet wird.

Plan

Legt fest, dass die Aktivität alle 15 Minuten eine Stunde lang ausgeführt wird.

Aufgaben

- Erstellen Sie die Pipeline
- Überwachen der ausgeführten Pipeline
- Anzeigen der Ausgabe
- Löschen der Pipeline

Erstellen Sie die Pipeline

Die schnellste Möglichkeit zum Einstieg in AWS Data Pipeline ist die Verwendung einer Pipeline-Definition namens Vorlage.

So erstellen Sie die Pipeline

1. Öffnen Sie die AWS Data Pipeline Konsole unter https://console.aws.amazon.com/datapipeline/.
2. Wählen Sie auf der Navigationsleiste eine Region aus. Sie können unabhängig von Ihrem Standort jede verfügbare Region auswählen. Viele AWS-Ressourcen sind spezifisch für eine Region, aber AWS Data Pipeline ermöglicht Ihnen die Verwendung von Ressourcen, die zu einer anderen Region gehören als die Pipeline.
3. Der erste Bildschirm, den Sie sehen, hängt davon ab, ob Sie in der aktuellen Region eine Pipeline erstellt haben.
   a. Wenn Sie in dieser Region keine Pipeline erstellt haben, zeigt die Konsole einen Einführungsbildschirm an. Wählen Sie Get started now.
   b. Wenn Sie in dieser Region bereits eine Pipeline erstellt haben, zeigt die Konsole eine Seite an, auf der Ihre Pipelines für die Region aufgeführt sind. Wählen Sie Create new pipeline (Neue Pipeline erstellen) aus.
4. Geben Sie unter Name einen Namen für Ihre Pipeline ein.
5. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für Ihre Pipeline ein.
6. Wählen Sie unter Quelle die Option Mithilfe einer Vorlage erstellen aus und wählen Sie dann die folgende Vorlage aus: Erste Schritte mit ShellCommandActivity.

7. Nach der Auswahl der Vorlage öffnet sich der Abschnitt Parameters. Behalten Sie dort die Standardwerte für S3 input folder und Shell command to run bei. Klicken Sie neben S3 output folder auf das Ordnersymbol, wählen Sie einen Ihrer Buckets oder Ordner aus und klicken Sie anschließend auf Select.
8. Behalten Sie unter Schedule die Standardwerte bei. Wenn Sie die Pipeline aktivieren, beginnen die Pipeline-Ausführungen und werden alle 15 Minuten eine Stunde lang ausgeführt.

   Wenn Sie möchten, können Sie stattdessen auch die Option Run once on pipeline activation auswählen.
9. Lassen Sie unter Pipeline-Konfiguration die Protokollierung aktiviert. Wählen Sie das Ordnersymbol unter S3-Speicherort für Protokolle, wählen Sie einen Ihrer Buckets oder Ordner aus und wählen Sie dann Auswählen.

   Wenn Sie möchten, können Sie stattdessen die Protokollierung deaktivieren.
10. Belassen Sie unter Sicherheit/Zugriff die IAM-Rollen auf Standard.
11. Klicken Sie auf Activate.

    Wenn Sie möchten, können Sie in Architect Bearbeiten wählen, um diese Pipeline zu ändern. Sie können beispielsweise Vorbedingungen hinzufügen.

Überwachen der ausgeführten Pipeline

Nachdem Sie Ihre Pipeline aktiviert haben, können Sie auf die Seite Execution details gehen, wo Sie den Fortschritt Ihrer Pipeline überwachen können.

So überwachen Sie den Fortschritt Ihrer Pipeline

1. Klicken Sie auf Update oder drücken Sie F5, um den angezeigten Status zu aktualisieren.

   > **Tip**
   >
   > Wenn keine Ausführungen aufgelistet sind, stellen Sie sicher, dass Start (in UTC) und End (in UTC) die geplante Start- und Endzeit Ihrer Pipeline abdecken, und klicken Sie dann auf Update.
2. Wenn der Status jedes Objekt in der Pipeline `FINISHED` ist, hat Ihre Pipeline die geplanten Tasks erfolgreich fertiggestellt.

3. Wenn Ihre Pipeline nicht erfolgreich abgeschlossen wurde, überprüfen Sie Ihre Pipeline-Einstellungen auf Probleme. Weitere Informationen zur Fehlerbehebung bei fehlgeschlagenen oder unvollständigen Instance-Ausführungen Ihrer Pipeline finden Sie unter Beheben typischer Probleme.

Anzeigen der Ausgabe

Öffnen Sie die Amazon S3-Konsole und navigieren Sie zu Ihrem Bucket. Wenn Sie Ihre Pipeline alle 15 Minuten eine Stunde lang ausgeführt haben, sehen Sie vier Unterordner mit Zeitstempeln. Jeder Unterordner enthält die Ausgabe in einer Datei mit dem Namen `output.txt`. Da wir das Skript jedes Mal auf derselben Eingabedatei ausgeführt haben, sind die Ausgabedateien identisch.

Löschen der Pipeline

Löschen Sie Ihre Pipeline, damit keine Gebühren mehr anfallen. Wenn Sie Ihre Pipeline löschen, werden die Pipeline-Definition und alle zugehörigen Objekte gelöscht.

Um deine Pipeline zu löschen

1. Wählen Sie auf der Seite „Pipelines auflisten“ Ihre Pipeline aus.
2. Klicken Sie auf Aktionen und wählen Sie dann Löschen.
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen).

Wenn Sie mit der Ausgabe aus diesem Tutorial fertig sind, löschen Sie die Ausgabeordner aus Ihrem Amazon S3-Bucket.

Arbeiten mit Pipelines

Sie können Pipelines mithilfe der Befehlszeilenschnittstelle (CLI) oder des SDK verwalten, erstellen und ändern. AWS Die folgenden Abschnitte beschreiben die grundlegenden AWS Data Pipeline-Konzepte und veranschaulichen, wie Sie mit Pipelines arbeiten.

> **Important**
>
> Bevor Sie beginnen, sehen Sie sich Einrichten für AWS Data Pipeline an.

Inhalt

Eine Pipeline erstellen

AWS Data Pipeline bietet Ihnen mehrere Möglichkeiten zum Erstellen von Pipelines:

- Verwenden Sie die AWS Command Line Interface (CLI) mit einer Vorlage, die der Einfachheit halber bereitgestellt wird. Weitere Informationen finden Sie unter Erstellen Sie mit der CLI eine Pipeline aus Data Pipeline-Vorlagen.

- Sie können die AWS Command Line Interface (CLI) zusammen mit einer Pipeline-Definitionsdatei im JSON-Format verwenden.
- Sie können ein AWS SDK mit einer sprachspezifischen API verwenden. Weitere Informationen finden Sie unter Arbeiten mit der API.

Erstellen Sie mit der CLI eine Pipeline aus Data Pipeline-Vorlagen

Data Pipeline bietet mehrere vorkonfigurierte Pipeline-Definitionen, sogenannte Vorlagen. Sie können Vorlagen für Ihre ersten Schritte mit AWS Data Pipeline verwenden. Diese Vorlagen sind in einem öffentlichen Bucket am Amazon S3-Standort verfügbar:`s3://datapipeline-us-east-1/templates/`. Diese vordefinierten Vorlagen wurden für bestimmte Anwendungsfälle erstellt und können zur Erstellung von Pipelines verwendet werden. Sie können `aws s3 ls --recursive "s3://datapipeline-us-east-1/templates/"` es verwenden, um alle verfügbaren Vorlagen aufzulisten.

Erstellen Sie mit der CLI eine Pipeline aus einer Vorlage

Angenommen, Sie möchten eine Pipeline erstellen, die eine DynamoDB-Tabelle nach Amazon S3 exportiert. Die in diesem Fall zu verwendende Vorlage finden Sie unter:`s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json`.

Um die JSON-Vorlage herunterzuladen und eine Pipeline mit der CLI zu erstellen

1. Laden Sie die Vorlage mit der `aws s3 cp` CLI oder Curl herunter. Beispiel:

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

2. Nehmen Sie nach Bedarf Änderungen an der heruntergeladenen Vorlage vor. Um beispielsweise die neueste EMR-Release-Version zu verwenden, ändern Sie das `releaseLabel` Feld im `EmrClusterForBackup` Objekt, ändern Sie die Master- und Core-Instance-Typen und ändern Sie die Standardwerte der Parameter in der Vorlage.
3. Erstellen Sie eine Pipeline mit der `create-pipeline` CLI. Beispiel:

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

4. Notieren Sie sich die erstellte Pipeline-ID.

5. Wird verwendet`put-pipeline-definition`, um die Definition hochzuladen. Geben Sie Werte der Parameter an, deren Standardwerte Sie mithilfe der `--parameter-values` Option überschreiben möchten.

Weitere Informationen zu Vorlagen finden Sie unter Auswahl einer Vorlage.

Auswahl einer Vorlage

Die folgenden Vorlagen können aus dem Amazon S3-Bucket heruntergeladen werden:`s3://datapipeline-us-east-1/templates/`.

Vorlagen

- Erste Schritte mit ShellCommandActivity
- AWSCLI-Befehl ausführen
- DynamoDB-Tabelle nach S3 exportieren
- DynamoDB-Backup-Daten aus S3 importieren
- Job auf einem Amazon EMR-Cluster ausführen
- Vollständige Kopie von Amazon RDS MySQL Table auf Amazon S3
- Inkrementelle Kopie der Amazon RDS-MySQL-Tabelle nach Amazon S3
- S3-Daten in die Amazon RDS-MySQL-Tabelle laden
- Vollständige Kopie der Amazon RDS-MySQL-Tabelle in Amazon Redshift
- Inkrementelles Kopieren einer Amazon RDS-MySQL-Tabelle nach Amazon Redshift
- Daten aus Amazon S3 in Amazon Redshift laden

Erste Schritte mit ShellCommandActivity

Die ShellCommandActivity Vorlage Getting Started using führt ein Shell-Befehlsskript aus, um die Anzahl der GET-Anfragen in einer Protokolldatei zu zählen. Die Ausgabe wird bei jedem geplanten Lauf der Pipeline in einen Amazon S3-Speicherort mit Zeitstempel geschrieben.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- ShellCommandActivity
- S3 InputNode
- S3 OutputNode

- Ec2Resource

AWSCLI-Befehl ausführen

Diese Vorlage führt einen vom Benutzer angegebenen AWS CLI-Befehl in festgelegten Intervallen aus.

DynamoDB-Tabelle nach S3 exportieren

Die Vorlage „DynamoDB-Tabelle in S3 exportieren“ plant einen Amazon EMR-Cluster, um Daten aus einer DynamoDB-Tabelle in einen Amazon S3-Bucket zu exportieren. Diese Vorlage verwendet einen Amazon EMR-Cluster, dessen Größe proportional zum Wert des für die DynamoDB-Tabelle verfügbaren Durchsatzes ist. Sie können IOPs in einer Tabelle zwar erhöhen, dies kann aber zu zusätzlichen Kosten beim Importieren und Exportieren führen. Bisher wurde für den Export ein verwendetHiveActivity, jetzt wird nativ verwendetMapReduce.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- EmrActivity
- EmrCluster
- DynamoDB DataNode
- S3 DataNode

DynamoDB-Backup-Daten aus S3 importieren

Die Vorlage „DynamoDB-Backup-Daten aus S3 importieren“ plant, dass ein Amazon EMR-Cluster ein zuvor erstelltes DynamoDB-Backup in Amazon S3 in eine DynamoDB-Tabelle lädt. Bestehende Elemente in der DynamoDB-Tabelle werden mit denen aus den Backup-Daten aktualisiert, und neue Elemente werden der Tabelle hinzugefügt. Diese Vorlage verwendet einen Amazon EMR-Cluster, dessen Größe proportional zum Wert des für die DynamoDB-Tabelle verfügbaren Durchsatzes ist. Sie können IOPs in einer Tabelle zwar erhöhen, dies kann aber zu zusätzlichen Kosten beim Importieren und Exportieren führen. Bisher wurde beim Import ein verwendet, HiveActivity jetzt wird nativ verwendetMapReduce.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- EmrActivity
- EmrCluster

- DynamoDB DataNode
- S3 DataNode
- S3 PrefixNotEmpty

Job auf einem Amazon EMR-Cluster ausführen

Die Vorlage Run Job on an Elastic MapReduce Cluster startet einen Amazon EMR-Cluster auf der Grundlage der bereitgestellten Parameter und beginnt mit der Ausführung von Schritten auf der Grundlage des angegebenen Zeitplans. Sobald der Auftrag abgeschlossen ist, wird der EMR-Cluster beendet. Optionale Bootstrap-Aktionen können angegeben werden, um zusätzliche Software zu installieren oder die Konfiguration der Anwendung im Cluster zu ändern.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- EmrActivity
- EmrCluster

Vollständige Kopie von Amazon RDS MySQL Table auf Amazon S3

Die Vorlage Vollständige Kopie der RDS-MySQL-Tabelle in S3 kopiert eine gesamte Amazon RDS-MySQL-Tabelle und speichert die Ausgabe an einem Amazon S3-Speicherort. Die Ausgabe wird als CSV-Datei in einem Unterordner mit Zeitstempel unter dem angegebenen Amazon S3-Speicherort gespeichert.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Inkrementelle Kopie der Amazon RDS-MySQL-Tabelle nach Amazon S3

Die Vorlage „Inkrementelle Kopie der RDS-MySQL-Tabelle in S3“ erstellt eine inkrementelle Kopie der Daten aus einer Amazon RDS-MySQL-Tabelle und speichert die Ausgabe an einem Amazon S3-Speicherort. Die Amazon RDS-MySQL-Tabelle muss eine Spalte „Zuletzt geändert“ haben.

Diese Vorlage kopiert alle Änderungen, die ab dem geplanten Startzeitpunkt zwischen festgelegten Intervallen an der Tabelle vorgenommen werden. Der Zeitplantyp ist Zeitreihe. Wenn also eine Kopie für eine bestimmte Stunde geplant wurde, werden die Tabellenzeilen AWS Data Pipeline kopiert, die mit einem Zeitstempel der letzten Änderung versehen sind, der innerhalb dieser Stunde liegt. Physische Löschvorgänge an der Tabelle werden nicht kopiert. Die Ausgabe wird bei jedem geplanten Lauf in einen Unterordner mit Zeitstempel unter dem Amazon S3-Speicherort geschrieben.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

S3-Daten in die Amazon RDS-MySQL-Tabelle laden

Die Vorlage „S3-Daten in die RDS-MySQL-Tabelle laden“ plant, dass eine Amazon EC2-Instance die CSV-Datei aus dem unten angegebenen Amazon S3-Dateipfad in eine Amazon RDS-MySQL-Tabelle kopiert. Die CSV-Datei sollte über keine Kopfzeile verfügen. Die Vorlage aktualisiert bestehende Einträge in der Amazon RDS-MySQL-Tabelle mit denen in den Amazon S3-Daten und fügt der Amazon RDS-MySQL-Tabelle neue Einträge aus den Amazon S3-Daten hinzu. Sie können die Daten in eine vorhandene Tabelle laden oder eine SQL-Abfrage zum Erstellen einer neuen Tabelle bereitstellen.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Vorlagen von Amazon RDS zu Amazon Redshift

Die folgenden beiden Vorlagen kopieren Tabellen von Amazon RDS MySQL nach Amazon Redshift. Dabei wird ein Übersetzungsskript verwendet, das eine Amazon Redshift-Tabelle unter Verwendung des Quelltabellenschemas mit den folgenden Einschränkungen erstellt:

- Wenn kein Verteilungsschlüssel angegeben ist, wird der erste Primärschlüssel aus der Amazon RDS-Tabelle als Verteilungsschlüssel festgelegt.
- Sie können keine Spalte überspringen, die in einer Amazon RDS-MySQL-Tabelle vorhanden ist, wenn Sie eine Kopie nach Amazon Redshift erstellen.
- (Optional) Sie können eine Zuordnung von Amazon RDS MySQL zu Amazon Redshift als einen der Parameter in der Vorlage angeben. Wenn dies angegeben ist, verwendet das Skript dies, um die Amazon Redshift-Tabelle zu erstellen.

Wenn der `Overwrite_Existing` Amazon Redshift-Einfügemodus verwendet wird:

- Wenn kein Verteilungsschlüssel bereitgestellt wird, wird ein Primärschlüssel in der Amazon RDS-MySQL-Tabelle verwendet.
- Wenn zusammengesetzte Primärschlüssel für die Tabelle vorhanden sind, wird der erste davon als Verteilungsschlüssel verwendet, sofern kein Verteilungsschlüssel bereitgestellt wird. Nur der erste zusammengesetzte Schlüssel ist als Primärschlüssel in der Amazon Redshift-Tabelle festgelegt.
- Wenn kein Verteilungsschlüssel bereitgestellt wird und die Amazon RDS-MySQL-Tabelle keinen Primärschlüssel enthält, schlägt der Kopiervorgang fehl.

Weitere Informationen zu Amazon Redshift finden Sie in den folgenden Themen:

- Amazon-Redshift-Cluster
- Amazon Redshift KOPIEREN
- Verteilungsstile und DISTKEY-Beispiele
- Sortierschlüssel

In der folgenden Tabelle wird beschrieben, wie das Skript die Datentypen umwandelt:

Datentypübersetzungen zwischen MySQL und Amazon Redshift

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
TINYINT, TINYINT (Größe)	SMALLINT	MySQL: -128 bis 127. Die maximale Anzahl von Ziffern kann in Klammern angegeben werden.

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
		Amazon Redshift: INT2. 2-Byte-Ganzzahl mit Vorzeichen
TINYINT UNSIGNED, TINYINT (Größe) UNSIGNED	SMALLINT	MySQL: 0 bis 255 UNSIGNED. Die maximale Anzahl von Ziffern kann in Klammern angegeben werden. Amazon Redshift: INT2. 2-Byte-Ganzzahl mit Vorzeichen
SMALLINT, SMALLINT (Größe)	SMALLINT	MySQL: -32768 bis 32767 normal. Die maximale Anzahl von Ziffern kann in Klammern angegeben werden. Amazon Redshift: INT2. 2-Byte-Ganzzahl mit Vorzeichen
SMALLINT UNSIGNED, SMALLINT(Größe) UNSIGNED,	INTEGER	MySQL: 0 bis 65535 UNSIGNED*. Die maximale Anzahl von Ziffern kann in Klammern angegeben werden. Amazon Redshift: INT4. 4-Byte-Ganzzahl mit Vorzeichen
MEDIUMINT, MEDIUMINT (Größe)	INTEGER	MySQL: 388608 bis 8388607. Die maximale Anzahl von Ziffern kann in Klammern angegeben werden. Amazon Redshift: INT4. 4-Byte-Ganzzahl mit Vorzeichen

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
MEDIUMINT UNSIGNED, MEDIUMINT (Größe) UNSIGNED	INTEGER	MySQL: 0 bis 16777215. Die maximale Anzahl von Ziffern kann in Klammern angegeben werden. Amazon Redshift: INT4. 4-Byte-Ganzzahl mit Vorzeichen
INT, INT(Größe)	INTEGER	MySQL: 147483648 bis 2147483647 Amazon Redshift: INT4. 4-Byte-Ganzzahl mit Vorzeichen
INT UNSIGNED, INT(Größe) UNSIGNED	BIGINT	MySQL: 0 bis 4294967295 Amazon Redshift: INT8. 8-Byte-Ganzzahl mit Vorzeichen
BIGINT BIGINT(Größe)	BIGINT	Amazon Redshift: INT8. 8-Byte-Ganzzahl mit Vorzeichen
BIGINT UNSIGNED BIGINT(Größe) UNSIGNED	VARCHAR(20*4)	MySQL: 0 bis 18446744073709551615 Amazon Redshift: Kein natives Äquivalent, daher wird ein Char-Array verwendet.

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
FLOAT FLOAT(Größe,d) FLOAT(Größe,d) UNSIGNED	REAL	Die maximale Anzahl von Ziffern kann im Größenparameter angegeben werden. Die maximale Anzahl von Ziffern rechts neben dem Dezimalzeichen wird im d-Parameter angegeben. Amazon Redshift: FLOAT4
DOUBLE(Größe,d)	DOUBLE PRECISION	Die maximale Anzahl von Ziffern kann im Größenparameter angegeben werden. Die maximale Anzahl von Ziffern rechts neben dem Dezimalzeichen wird im d-Parameter angegeben. Amazon Redshift: FLOAT8
DECIMAL(Größe,d)	DECIMAL(Größe,d)	Als Zeichenfolge gespeichertes DOUBLE, ermöglicht ein festes Dezimalzeichen. Die maximale Anzahl von Ziffern kann im Größenparameter angegeben werden. Die maximale Anzahl von Ziffern rechts neben dem Dezimalzeichen wird im d-Parameter angegeben. Amazon Redshift: Kein natives Äquivalent.

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
CHAR(Größe)	VARCHAR(Größe*4)	Enthält eine Zeichenfolge fester Länge, die aus Buchstaben, Ziffern und Sonderzeichen bestehen kann. Die feste Größe wird als Parameter in Klammern angegeben. Kann bis zu 255 Zeichen speichern. Rechts aufgefüllt mit Leerzeichen. Amazon Redshift: Der CHAR-Datentyp unterstützt kein Multibyte-Zeichen, daher wird VARCHAR verwendet. In Übereinstimmung mit RFC3629 sind maximal pro Zeichen 4 Byte zulässig, wodurch die Zeichentabelle auf U+10FFFF eingeschränkt wird.
VARCHAR(Größe)	VARCHAR(Größe*4)	Kann bis zu 255 Zeichen speichern. VARCHAR unterstützt nicht die folgenden ungültigen UTF-8-Codepunkte: 0xD800 – 0xDFFF, (Bytefolgen: ED A0 80 – ED BF BF), 0xFDD0 – 0xFDEF, 0xFFFE und 0xFFFF, (Bytefolgen: EF B7 90 – EF B7 AF, EF BF BE und EF BF BF)

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
TINYTEXT	VARCHAR(255*4)	Enthält eine Zeichenfolge mit einer maximalen Länge von 255 Zeichen.
TEXT	VARCHAR(max)	Enthält eine Zeichenfolge mit einer maximalen Länge von 65.535 Zeichen.
MEDIUMTEXT	VARCHAR(max)	0 bis 16.777.215 Zeichen
LONGTEXT	VARCHAR(max)	0 bis 4.294.967.295 Zeichen
BOOLEAN BOOL TINYINT(1)	BOOLEAN	MySQL: Diese Typen sind Synonyme für TINYINT (1). Der Wert Null wird als „false“ angesehen. Werte ungleich Null werden als „true“ angesehen.
BINARY[(M)]	varchar(255)	M ist 0 bis 255 Byte, FIXED
VARBINARY(M)	VARCHAR(max)	0 bis 65.535 Byte
TINYBLOB	VARCHAR(255)	0 bis 255 Byte
BLOB	VARCHAR(max)	0 bis 65.535 Byte
MEDIUMBLOB	VARCHAR(max)	0 bis 16.777.215 Byte
LONGBLOB	VARCHAR(max)	0 bis 4.294.967.295 Byte
ENUM	VARCHAR(255*2)	Die Begrenzung gilt nicht für die Länge der Literal-Aufzählungszeichenfolge, sondern vielmehr für die in der Tabelle definierte Anzahl von Aufzählungswerten.

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
SET	VARCHAR(255*2)	Wie enum.
DATUM	DATUM	(JJJJ-MM-TT) „1000-01-01“ bis „9999-12-31“
TIME	VARCHAR(10*4)	(hh:mm:ss) „-838:59:59“ bis „838:59:59“
DATETIME	TIMESTAMP	(JJJJ-MM-TT hh:mm:ss) „1000-01-01 00:00:00“ bis „9999-12-31 23:59:59“
TIMESTAMP	TIMESTAMP	(JJJJMMTThhmmss) 19700101000000 bis 2037+
JAHR	VARCHAR(4*4)	(YYYY) 1900 bis 2155
Spalte SERIAL	ID-Generierung / Dieses Attribut ist für ein OLAP-Data Warehouse nicht erforderlich, da diese Spalte kopiert wird. Das Schlüsselwort SERIAL wird bei der Umwandlung nicht hinzugefügt.	SERIAL ist tatsächlich eine Entität namens SEQUENCE. Sie existiert unabhängig von der restlichen Tabelle. Spalte GENERATED BY DEFAULT entspricht: CREATE SEQUENCE Name; CREATE TABLE Tabelle (Spalte INTEGER NOT NULL DEFAULT nextval(Name));

MySQL-Datentyp	Amazon Redshift-Datentyp	Hinweise
Spalte BIGINT UNSIGNED NOT NULL AUTO_INCR EMENT UNIQUE	ID-Generierung / Dieses Attribut ist für OLAP-Data Warehouse nicht erforderlich, da diese Spalte kopiert wird. Das Schlüsselwort SERIAL wird bei der Umwandlung daher nicht hinzugefügt.	SERIAL ist tatsächlich eine Entität namens SEQUENCE. Sie existiert unabhängig von der restlichen Tabelle. Spalte GENERATED BY DEFAULT entspricht: CREATE SEQUENCE Name; CREATE TABLE Tabelle (Spalte INTEGER NOT NULL DEFAULT nextval(Name));
ZEROFILL	Das Schlüsselwort ZEROFILL wird bei der Umwandlung nicht hinzugefügt.	INT UNSIGNED ZEROFILL NOT NULL ZEROFILL füllt den angezeigt en Wert des Feldes bis zur Anzeigenbreite in der Spaltendefinition mit Nullen auf. Werte, die länger als die Anzeigenbreite sind, werden nicht abgeschnitten. Beachten Sie, dass die Syntax von ZEROFILL auch UNSIGNED impliziert.

Vollständige Kopie der Amazon RDS-MySQL-Tabelle in Amazon Redshift

Die Vorlage „Vollständige Kopie der Amazon RDS-MySQL-Tabelle in Amazon Redshift“ kopiert die gesamte Amazon RDS-MySQL-Tabelle in eine Amazon Redshift-Tabelle, indem Daten in einem Amazon S3-Ordner bereitgestellt werden. Der Amazon S3-Staging-Ordner muss sich in derselben Region wie der Amazon Redshift-Cluster befinden. Eine Amazon Redshift-Tabelle wird mit demselben Schema wie die Amazon RDS-MySQL-Quelltabelle erstellt, sofern sie nicht bereits

vorhanden ist. Bitte geben Sie alle Datentypüberschreibungen von Amazon RDS MySQL to Amazon Redshift an, die Sie bei der Erstellung von Amazon Redshift-Tabellen anwenden möchten.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- CopyActivity
- RedshiftCopyActivity
- S3 DataNode
- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

Inkrementelles Kopieren einer Amazon RDS-MySQL-Tabelle nach Amazon Redshift

Die Vorlage „Inkrementelle Kopie der Amazon RDS-MySQL-Tabelle in Amazon Redshift“ kopiert Daten aus einer Amazon RDS-MySQL-Tabelle in eine Amazon Redshift-Tabelle, indem Daten in einem Amazon S3-Ordner bereitgestellt werden.

Der Amazon S3-Staging-Ordner muss sich in derselben Region wie der Amazon Redshift-Cluster befinden.

AWS Data Pipelineverwendet ein Übersetzungsskript, um eine Amazon Redshift-Tabelle mit demselben Schema wie die Amazon RDS-MySQL-Quelltabelle zu erstellen, sofern sie nicht bereits existiert. Sie müssen alle Datentypüberschreibungen von Amazon RDS MySQL to Amazon Redshift angeben, die Sie bei der Erstellung der Amazon Redshift-Tabelle anwenden möchten.

Diese Vorlage kopiert Änderungen, die an der Amazon RDS-MySQL-Tabelle zwischen geplanten Intervallen vorgenommen werden, beginnend mit der geplanten Startzeit. Physische Löschungen der Amazon RDS-MySQL-Tabelle werden nicht kopiert. Sie müssen den Namen der Spalte angeben, in der der Zeitpunkt der letzten Änderung gespeichert wird.

Wenn Sie die Standardvorlage verwenden, um Pipelines für inkrementelle Amazon RDS-Kopien zu erstellen, wird eine Aktivität mit dem Standardnamen `RDSToS3CopyActivity` erstellt. Sie können sie umbenennen.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- CopyActivity

- RedshiftCopyActivity
- S3 DataNode
- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

Daten aus Amazon S3 in Amazon Redshift laden

Die Vorlage „Daten aus S3 in Redshift laden“ kopiert Daten aus einem Amazon S3-Ordner in eine Amazon Redshift-Tabelle. Sie können die Daten in eine vorhandene Tabelle laden oder eine SQL-Abfrage zum Erstellen der Tabelle bereitstellen.

Die Daten werden auf der Grundlage der Amazon `COPY` Redshift-Optionen kopiert. Die Amazon Redshift-Tabelle muss dasselbe Schema haben wie die Daten in Amazon S3. `COPY`Optionen finden Sie unter COPY im Amazon Redshift Database Developer Guide.

Die Vorlage verwendet die folgenden Pipeline-Objekte:

- CopyActivity
- RedshiftCopyActivity
- S3 DataNode
- RedshiftDataNode
- RedshiftDatabase
- Ec2Resource

Erstellen einer Pipeline mithilfe parametrisierter Vorlagen

Sie können eine Pipeline-Definition anhand einer parametrisierten Vorlage anpassen. Auf diese Weise können Sie eine gemeinsame Pipeline-Definition erstellen, aber unterschiedliche Parameter konfigurieren, wenn Sie die Pipeline-Definition zu einer neuen Pipeline hinzufügen.

Inhalt

- MyVariables zur Pipeline-Definition hinzufügen
- Definieren Sie Parameterobjekte
- Definieren von Parameterwerten

- Einreichung der Pipeline-Definition

MyVariables zur Pipeline-Definition hinzufügen

Geben Sie beim Erstellen der Pipeline-Definitionsdatei mithilfe der folgenden Syntax Variablen an: #{my*Variable*}. Der Variablen muss `my` vorangestellt werden. Die folgende Pipeline-Definitionsdatei`pipeline-definition.json`, enthält beispielsweise die folgenden Variablen: *`myShellCmd`*, *`myS3 InputLoc`* und *`OutputLocmyS3`*.

> **Note**
>
> In einer Pipeline-Definition sind maximal 50 Parameter zulässig.

```
{
  "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"
    }
  ]
}
```

Definieren Sie Parameterobjekte

Sie können eine separate Datei mit Parameterobjekten erstellen, um die Variablen in Ihrer Pipeline-Definition zu definieren. Die folgende JSON-Datei,`parameters.json`, enthält beispielsweise Parameterobjekte für die OutputLoc Variablen *myShellCmd*, *myS3 InputLoc* und *myS3* aus der obigen Beispiel-Pipeline-Definition.

```
{
  "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**
>
> Sie können diese Objekte der Pipeline-Definitionsdatei anstatt über eine separate Datei auch direkt hinzufügen.

Die folgende Tabelle beschreibt die Attribute für Parameterobjekte.

Parameterattribute

Attribut	Typ	Beschreibung
`id`	Zeichenfolge	Der eindeutige Bezeichner des Parameters. Wenn der Wert bei der Eingabe oder Anzeige maskiert werden soll, fügen Sie als Präfix ein Sternchen ('*') hinzu. Zum Beispiel `*myVariable` —. Dabei ist zu beachten, dass der Wert dadurch auch

Attribut	Typ	Beschreibung
		verschlüsselt wird, bevor er durch AWS Data Pipeline gespeichert wird.
description	Zeichenfolge	Eine Beschreibung des Parameters.
type	Zeichenfolge, Ganzzahl, Double oder AWS::S3::ObjectKey	Der Parametertyp zur Definition des zulässigen Bereichs von Eingabewerten und Validierungsregeln. Der Standardwert ist eine Zeichenfolge.
optional	Boolesch	Gibt an, ob der Parameter optional oder erforderlich ist. Der Standardwert ist `false`.
allowedValues	Liste von Zeichenfolgen	Listet alle zulässigen Werte für den Parameter auf.
default	Zeichenfolge	Der Standardwert für den Parameter. Wenn Sie mithilfe von Parameterwerten einen Wert für diesen Parameter angeben, wird der Standardwert durch ihn überschrieben.
isArray	Boolesch	Gibt an, ob der Parameter ein Array ist.

Definieren von Parameterwerten

Sie können eine separate Datei zur Definition von Variablen mithilfe von Parameterwerten erstellen. Die folgende JSON-Datei,`file://values.json`, enthält beispielsweise den Wert für die *`OutputLocmyS3-Variable`* aus der obigen Beispiel-Pipeline-Definition.

```
{
```

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

Einreichung der Pipeline-Definition

Wenn Sie Ihre Pipeline-Definition senden, können Sie Parameter, Parameterobjekte und Parameterwerte angeben. Sie können den put-pipeline-definitionAWS CLIBefehl beispielsweise wie folgt verwenden:

```
$ 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**
>
> In einer Pipeline-Definition sind maximal 50 Parameter zulässig. Die Größe der Datei für `parameter-values-uri` darf maximal 15 kB betragen.

Anzeigen Ihrer Pipelines

Sie können Ihre Pipelines mithilfe der Befehlszeilenschnittstelle (CLI) anzeigen.

So zeigen Sie Ihre Pipelines über die AWS CLI an

- Verwenden Sie den folgenden list-pipelines-Befehl, um Ihre Pipelines aufzulisten:

  ```
  aws datapipeline list-pipelines
  ```

Interpretieren der Pipeline-Statuscodes

Die in der AWS Data Pipeline-Konsole und Befehlszeilenschnittstelle (CLI) angezeigten Statuswerte geben den Zustand einer Pipeline und ihrer Komponenten an. Der Pipeline-Status ist vereinfacht ausgedrückt ein Überblick über eine Pipeline. Wenn Sie weitere Informationen benötigen, zeigen Sie den Status der einzelnen Pipeline-Komponenten an.

Der Status einer Pipeline lautet `SCHEDULED`, wenn sie bereit ist (die Pipeline-Definition hat die Validierung bestanden), aktuell Arbeiten ausführt oder die Ausführung von Arbeiten beendet hat. Der Status einer Pipeline lautet `PENDING`, wenn sie nicht aktiviert ist oder keine Arbeiten ausführen kann (z. B. hat die Pipeline-Definition die Validierung nicht bestanden).

Eine Pipeline gilt als inaktiv, wenn ihr Status `PENDING`, `INACTIVE`oder `FINISHED` lautet. Für inaktive Pipelines fallen Gebühren an (weitere Informationen finden Sie unter – Preise).

Statuscodes

`ACTIVATING`

Die Komponente oder Ressource wird gestartet, z. B. eine EC2-Instance.

`CANCELED`

Die Komponente wurde von einem Benutzer oder AWS Data Pipeline bevor sie ausgeführt werden konnte, storniert. Dies kann automatisch geschehen, wenn ein Fehler in einer anderen Komponente oder Ressource auftritt, von der diese Komponente abhängt.

`CASCADE_FAILED`

Die Komponente oder Ressource wurde aufgrund eines Kaskadenausfalls aus einer ihrer Abhängigkeiten storniert, aber die Komponente war wahrscheinlich nicht die ursprüngliche Ursache des Fehlers.

`DEACTIVATING`

Die Pipeline wird deaktiviert.

`FAILED`

Bei der Komponente oder Ressource ist ein Fehler aufgetreten und sie funktioniert nicht mehr. Wenn eine Komponente oder Ressource ausfällt, kann dies zu Abbrüchen und Ausfällen führen, die sich auf andere Komponenten auswirken, die von ihr abhängen.

`FINISHED`

Die Komponente hat ihre zugewiesene Arbeit abgeschlossen.

`INACTIVE`

Die Pipeline wurde deaktiviert.

`PAUSED`

Die Komponente wurde angehalten und führt derzeit ihre Arbeit nicht aus.

`PENDING`

Die Pipeline ist bereit, zum ersten Mal aktiviert zu werden.

`RUNNING`

Die Ressource läuft und ist bereit, Arbeit anzunehmen.

`SCHEDULED`

Die Ausführung der Ressource ist geplant.

`SHUTTING_DOWN`

Die Ressource wird heruntergefahren, nachdem sie ihre Arbeit erfolgreich abgeschlossen hat.

`SKIPPED`

Die Komponente hat Ausführungsintervalle übersprungen, nachdem die Pipeline aktiviert wurde. Dabei wurde ein Zeitstempel verwendet, der nach dem aktuellen Zeitplan liegt.

`TIMEDOUT`

Die Ressource hat den `terminateAfter` Schwellenwert überschritten und wurde angehaltenAWS Data Pipeline. Nachdem die Ressource diesen Status erreicht hat, AWS Data Pipeline werden die `retryTimeout` Werte `actionOnResourceFailureretryDelay`, und für diese Ressource ignoriert. Dieser Status gilt nur für Ressourcen.

`VALIDATING`

Die Pipeline-Definition wird von AWS Data Pipeline validiert.

`WAITING_FOR_RUNNER`

Die Komponente wartet darauf, dass ihr Worker-Client ein Arbeitselement abruft. Die Beziehung zwischen Komponente und Mitarbeiter und Kunde wird durch die `runsOn` `workerGroup` Oder-Felder gesteuert, die von dieser Komponente definiert werden.

`WAITING_ON_DEPENDENCIES`

Die Komponente überprüft, ob die standardmäßigen und vom Benutzer konfigurierten Vorbedingungen erfüllt sind, bevor sie ihre Arbeit ausführt.

Interpretieren des Pipeline- und Komponenten-Zustands

Jede Pipeline und Komponente innerhalb der betreffenden Pipeline geben als Zustand `HEALTHY`, `ERROR`, `"-"`, `No Completed Executions` oder `No Health Information Available` zurück.

Für eine Pipeline wird erst dann ein Zustand angegeben, nachdem eine Pipeline-Komponente erstmals ausgeführt wurde oder wenn die Vorbedingungen einer Komponente fehlgeschlagen sind. Der Zustand einzelner Komponenten wird im Pipeline-Zustand aggregiert. Der Fehlerstatus ist erst sichtbar, wenn Sie Ihre Pipeline-Ausführungsdetails anzeigen.

Pipeline-Zustände

`HEALTHY`

Der aggregierte Zustand aller Komponenten ist `HEALTHY`. Dies bedeutet, dass mindestens eine Komponente erfolgreich abgeschlossen worden sein muss. Sie können auf den Status `HEALTHY` klicken, um die neueste erfolgreich abgeschlossene Pipeline-Komponenten-Instance auf der Seite Execution Details zu sehen.

`ERROR`

Mindestens eine Komponente in der Pipeline verfügt über den Zustand `ERROR`. Sie können auf den Status `ERROR` klicken, um die neueste fehlgeschlagene Pipeline-Komponenten-Instance auf der Seite Execution Details zu sehen.

`No Completed Executions` oder `No Health Information Available`.

Für diese Pipeline wurde kein Zustand gemeldet.

> **Note**
>
> Während Komponenten ihren Zustand beinahe sofort aktualisieren, kann es bis zu fünf Minuten dauern, bis der Pipeline-Zustand aktualisiert wird.

Komponenten-Zustand

`HEALTHY`

Der Zustand einer Komponente (`Activity` oder `DataNode`) ist `HEALTHY`, wenn sie nach einer erfolgreichen Ausführung mit dem Status `FINISHED` oder `MARK_FINISHED` markiert wurde. Sie können auf den Namen der Komponente oder auf den Status `HEALTHY` klicken, um auf der Seite Execution Details die neuesten erfolgreich abgeschlossenen Pipeline-Komponenten-Instances anzuzeigen.

`ERROR`

Es ist ein Fehler auf Komponentenebene aufgetreten oder eine ihrer Vorbedingungen ist fehlgeschlagen. Dieser Fehler wird durch einen Status von `FAILED`, `TIMEOUT` oder `CANCELED` ausgelöst. Sie können auf den Namen der Komponente oder auf den Status `ERROR` klicken, um auf der Seite Execution Details die neueste fehlgeschlagene Pipeline-Komponenten-Instance anzuzeigen.

`No Completed Executions` oder `No Health Information Available`

Für diese Komponente wurde kein Zustand gemeldet.

Anzeigen Ihrer Pipeline-Definitionen

Verwenden Sie die Befehlszeilenschnittstelle (CLI), um Ihre Pipeline-Definition einzusehen. Die CLI druckt eine Pipeline-Definitionsdatei im JSON-Format. Weitere Informationen zur Syntax und Nutzung von Pipeline-Definitionsdateien finden Sie unter Syntax der Pipeline-Definitionsdatei.

Wenn Sie die CLI verwenden, ist es eine gute Idee, die Pipeline-Definition abzurufen, bevor Sie Änderungen einreichen, da es möglich ist, dass ein anderer Benutzer oder Prozess die Pipeline-Definition geändert hat, nachdem Sie das letzte Mal damit gearbeitet haben. Durch Herunterladen einer Kopie der aktuellen Definition, auf der Sie Ihre Änderungen basieren, können Sie sicher sein, dass Sie an der neuesten Pipeline-Definition arbeiten. Auch wird empfohlen, die Pipeline-Definition nach den vorgenommenen Änderungen erneut abzurufen, um sicherzugehen, dass die Aktualisierung erfolgreich war.

Wenn Sie die CLI verwenden, können Sie zwei verschiedene Versionen Ihrer Pipeline abrufen. Die Version `active` ist die Pipeline, die gerade ausgeführt wird. Die Version `latest` ist eine Kopie, die beim Bearbeiten einer gerade ausgeführten Pipeline erstellt wird. Wenn Sie die bearbeitete Pipeline hochladen, ist sie nun die Version `active` und die vorherige Version `active` ist nicht mehr verfügbar.

So rufen Sie eine Pipeline-Definition über die AWS CLI ab

Verwenden Sie den get-pipeline-definitionBefehl, um die vollständige Pipeline-Definition abzurufen. Die Pipeline-Definition wird zur Standardausgabe (stdout) ausgegeben.

Das folgende Beispiel ruft die Pipeline-Definition für die angegebene Pipeline ab.

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

Zum Abrufen einer bestimmten Version einer Pipeline verwenden Sie die Option `--version`. Das folgende Beispiel ruft die Version `active` der angegebenen Pipeline ab.

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

Anzeigen von Pipeline-Instance Details

Sie können den Fortschritt Ihrer Pipeline überwachen. Weitere Informationen zum Instance-Status finden Sie unter Interpretieren der Pipeline-Statusdetails. Weitere Informationen zur Fehlerbehebung bei fehlgeschlagenen oder unvollständigen Instance-Ausführungen Ihrer Pipeline finden Sie unter Beheben typischer Probleme.

So überwachen Sie den Fortschritt einer Pipeline über die AWS CLI

Um Pipeline-Instance-Details abzurufen, wie z. B. wie oft eine Pipeline ausgeführt wurde, verwenden Sie den Befehl list-runs. Mit diesem Befehl können Sie die Liste der zurückgegebenen Ausführungen nach ihrem aktuellem Status oder dem Datumsbereich filtern, in dem sie gestartet wurden. Das Filtern der Ergebnisse ist hilfreich, da der Ausführungsverlauf je nach Alter und Zeitplanung der Pipeline groß sein kann.

Das folgende Beispiel ruft Informationen zu allen Ausführungen ab.

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

Das folgende Beispiel ruft Informationen zu allen abgeschlossenen Ausführungen ab.

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

Das folgende Beispiel ruft Informationen zu allen in einem bestimmten Zeitrahmen gestarteten Ausführungen ab.

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

Anzeigen von Pipeline-Protokollen

Die Protokollierung auf Pipelineebene wird bei der Pipelineerstellung unterstützt, indem ein Amazon S3-Standort entweder in der Konsole oder mit einem `pipelineLogUri` im Standardobjekt in SDK/

CLI angegeben wird. Die Verzeichnisstruktur für jede Pipeline in diesem URI wird nachstehend beschrieben:

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

Für die Pipeline `df-00123456ABC7DEF8HIJK` sieht die Verzeichnisstruktur folgendermaßen aus:

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

Für `ShellCommandActivity` werden Protokolle für `stderr` und `stdout`, die diesen Aktivitäten zugeordnet sind, bei jedem Versuch im Verzeichnis gespeichert.

Für Ressourcen wie `EmrCluster` mit festgelegtem `emrLogUri` hat dieser Wert Vorrang. Andernfalls folgen Ressourcen (einschließlich der TaskRunner Protokolle für diese Ressourcen) der obigen Pipeline-Protokollierungsstruktur.

Um die Protokolle für einen bestimmten Pipeline-Lauf anzuzeigen:

1. Rufen Sie die ab, `ObjectId` indem Sie aufrufen`query-objects`, um die genaue Objekt-ID zu erhalten. Beispiel:

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

   `query-objects`ist eine paginierte CLI und kann ein Paginierungstoken zurückgeben, wenn es mehr Ausführungen für die angegebene gibt. `pipeline-id` Sie können das Token verwenden, um alle Versuche zu durchlaufen, bis Sie das erwartete Objekt gefunden haben. Eine Rückgabe ObjectId würde beispielsweise wie folgt aussehen:`@TableBackupActivity_2023-05-020T18:05:18_Attempt=1`.
2. Rufen Sie mithilfe von den ObjectId den folgenden Schritten den Speicherort des Protokolls ab:

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

Fehlermeldung einer fehlgeschlagenen Aktivität

Um die Fehlermeldung zu erhalten, müssen Sie zuerst die ObjectId Verwendung verwendenquery-objects.

Verwenden Sie nach dem Abrufen des Fehlgeschlagenen die describe-objects CLIObjectId, um die eigentliche Fehlermeldung abzurufen.

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

Ein Objekt stornieren oder erneut ausführen oder als abgeschlossen markieren

Verwenden Sie die set-status CLI, um ein laufendes Objekt abzubrechen oder ein ausgefallenes Objekt erneut auszuführen oder ein laufendes Objekt als Fertig zu markieren.

Rufen Sie zunächst die Objekt-ID mit der query-objects CLI ab. Beispiel:

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

Verwenden Sie die set-status CLI, um den Status des gewünschten Objekts zu ändern. Beispiel:

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

Bearbeiten Ihrer Pipeline

Wenn Sie bestimmte Aspekte einer Ihrer Pipelines ändern müssen, können Sie die Definition der Pipeline entsprechend aktualisieren. Wenn Änderungen an einer derzeit ausgeführten Pipeline vorgenommen wurden, müssen Sie die Pipeline erneut aktivieren, damit die Änderungen wirksam werden. Zudem ist es möglich, eine oder mehrere Pipeline-Komponenten erneut auszuführen.

Inhalt

Einschränkungen

Solange sich die Pipeline im `PENDING` Status befindet und nicht aktiviert ist, können Sie keine Änderungen daran vornehmen. Nachdem eine Pipeline aktiviert wurde, gelten beim Bearbeiten der Pipeline die folgenden Einschränkungen. Die von Ihnen vorgenommenen Änderungen werden für neue Ausführungen der Pipeline-Objekte übernommen, nachdem Sie sie speichern und die Pipeline dann erneut aktivieren.

- Ein Objekt kann nicht entfernt werden
- Der Planungszeitraum eines vorhandenen Objekts kann nicht geändert werden
- Referenzfelder in einem vorhandenen Objekt können nicht hinzugefügt, gelöscht oder abgeändert werden
- In dem Ausgabefeld eines neuen Objekts kann nicht auf ein vorhandenes Objekt verwiesen werden
- Das geplante Anfangsdatum eines Objekts kann nicht geändert werden (aktivieren Sie stattdessen die Pipeline mit bestimmten Angaben für Datum und Uhrzeit)

Bearbeiten einer Pipeline über die AWS CLI

Sie können eine Pipeline mit Befehlszeilen-Tools bearbeiten.

Laden Sie zunächst mit dem get-pipeline-definitionBefehl eine Kopie der aktuellen Pipeline-Definition herunter. Auf diese Weise können Sie sicher sein, dass Sie die neuesten Pipeline-Definition abändern. In dem folgenden Beispiel wird die Pipeline-Definition zur Standardausgabe (stdout) ausgegeben.

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

Speichern Sie die Pipeline-Definition in einer Datei und bearbeiten Sie sie nach Bedarf. Aktualisieren Sie Ihre Pipeline-Definition mit dem put-pipeline-definitionBefehl. Im folgenden Beispiel wird die Pipeline-Definitionsdatei hochgeladen.

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

Sie können die Pipeline-Definition mit dem Befehl `get-pipeline-definition` erneut abrufen, um sicherzustellen, dass die Aktualisierung erfolgreich war. Verwenden Sie zum Aktivieren der Pipeline den folgenden activate-pipeline-Befehl:

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

Wenn Sie möchten, können Sie die Pipeline ab einem bestimmten Datum und einer bestimmten Uhrzeit aktivieren. Verwenden Sie hierzu folgendermaßen die Option `--start-timestamp`:

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

Mit dem Befehl set-status können Sie eine oder mehrere Pipeline-Komponenten erneut ausführen.

Klonen Ihrer Pipeline

Klonen erstellt eine Kopie einer Pipeline und ermöglicht Ihnen einen Namen für die neue Pipeline anzugeben. Pipelines können in jedem Zustand geklont werden, sogar mit Fehlern. Die neue Pipeline verbleibt jedoch im Zustand `PENDING`, bis Sie sie manuell aktivieren. Der Clone-Vorgang verwendet für die neue Pipeline die neueste Version der ursprünglichen Pipeline-Definition und nicht die aktive Version. Bei dem Klonvorgang wird nicht der vollständige Zeitplan aus der ursprünglichen Pipeline in die neue Pipeline kopiert, sondern nur der festgelegte Zeitraum.

So klonen Sie eine Pipeline mit der AWS CLI:

1. Erstellen Sie eine neue Pipeline mit einem neuen Namen und einer eindeutigen ID. Notieren Sie sich die zurückgegebene Pipeline-ID.
2. Verwenden Sie die `get-pipeline-definition` CLI, um die Pipeline-Definition der vorhandenen Pipeline abzurufen, die geklont werden soll, und schreiben Sie sie in eine temporäre Datei. Notieren Sie sich den absoluten Pfad der Datei.
3. Verwenden Sie die `put-pipeline-definition` CLI, um die Pipeline-Definition von der vorhandenen Pipeline in die neue Pipeline zu kopieren.
4. Verwenden Sie die `get-pipeline-definition` CLI, um die Definition der neuen Pipeline abzurufen und die Pipeline-Definition zu überprüfen.

```
# 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
```

Tagging Ihrer Pipeline

Tags sind Schlüssel/Wert-Paare mit Unterscheidung nach Groß-/Kleinschreibung, die aus einem Schlüssel und einem optionalen Wert bestehen, die beide vom Benutzer definiert wurden. Sie können auf jede Pipeline bis zu 10 Tags anwenden. Tag-Schlüssel müssen für jede Pipeline eindeutig sein. Wenn Sie ein Tag mit einem Schlüssel hinzufügen, der der Pipeline bereits zugeordnet ist, ändert sich der Wert dieses Tags.

Durch das Anwenden eines Tags auf eine Pipeline werden die Tags auch an die zugrunde liegenden Ressourcen weitergegeben (z. B. Amazon EMR-Cluster und Amazon EC2-Instances). Diese Tags werden allerdings nicht auf Ressourcen im Zustand `FINISHED` oder in einem anderen beendeten Zustand übertragen. Sie können mit der CLI bei Bedarf Tags auf diese Ressourcen anwenden.

Wenn Sie ein Tag nicht mehr benötigen, können Sie es von Ihrer Pipeline entfernen.

So versehen Sie Ihre Pipeline mit Tags über die AWS CLI

Um einer neuen Pipeline Tags hinzuzufügen, fügen Sie die `--tags` Option zu Ihrem Befehl create-pipeline hinzu. Die folgende Option erstellt z. B. eine Pipeline mit zwei Tags, einem `environment`-Tag mit dem Wert `production` und einem `owner`-Tag mit dem Wert `sales`.

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

Um Tags zu einer vorhandenen Pipeline hinzuzufügen, verwenden Sie den Befehl add-tags folgendermaßen:

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

Um Tags von einer vorhandenen Pipeline zu entfernen, verwenden Sie den Befehl remove-tags folgendermaßen:

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

Deaktivieren Ihrer Pipeline

Wenn eine gerade ausgeführte Pipeline deaktiviert wird, wird die Ausführung der Pipeline angehalten. Um die Ausführung der Pipeline fortzusetzen, können Sie die Pipeline aktivieren. Dies ermöglicht Ihnen, die folgenden Änderungen vorzunehmen. Wenn Sie beispielsweise Daten in eine Datenbank schreiben, für die eine Wartung geplant ist, können Sie die Pipeline deaktivieren, und warten, bis die Wartung abgeschlossen ist, und die Pipeline dann aktivieren.

Beim Deaktivieren der Pipeline können Sie den Umgang mit laufenden Aktivitäten festlegen. Standardmäßig werden diese Aktivitäten sofort abgebrochen. Alternativ können Sie AWS Data Pipeline veranlassen, mit dem Deaktivieren der Pipeline zu warten, bis die Aktivitäten abgeschlossen wurden.

Beim Aktivieren einer deaktivierten Pipeline können Sie festlegen, wann sie fortgesetzt wird. Über die AWS CLI oder die API wird die Pipeline standardmäßig ab der letzten Ausführung fortgesetzt. Sie können auch angeben, an welchem Datum und um welche Uhrzeit die Pipeline fortgesetzt werden soll.

Deaktivieren Ihrer Pipeline über die AWS CLI

Verwenden Sie den folgenden deactivate-pipeline-Befehl zum Deaktivieren einer Pipeline:

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

Um die Pipeline erst nach der Ausführung aller Aktivitäten zu deaktivieren, fügen Sie die Option `--no-cancel-active` wie folgt hinzu:

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

Wenn Sie bereit sind, können Sie die Ausführung der Pipeline mit dem folgenden activate-pipeline-Befehl an der Stelle fortsetzen, an der sie unterbrochen wurde:

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

Um die Pipeline ab einem bestimmten Datum und einer bestimmten Uhrzeit zu starten, fügen Sie die Option `--start-timestamp` wie folgt hinzu:

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

Löschen Ihrer Pipeline

Wenn Sie eine Pipeline nicht mehr benötigen, z. B. eine Pipeline, die beim Testen der Anwendung erstellt wurde, sollten Sie sie löschen, damit sie nicht mehr aktiv genutzt wird. Beim Löschen wird eine Pipeline in den gelöschten Zustand versetzt. Im gelöschten Zustand besitzen Pipelines keine Pipeline-Definition und keinen Ausführungsverlauf mehr. Für solche Pipeline können daher keine Vorgänge mehr durchgeführt werden, einschließlich deren Beschreibung.

> **Important**
>
> Gelöschte Pipelines können nicht wiederhergestellt werden. Stellen Sie daher sicher, dass Sie die Pipeline zukünftig nicht mehr benötigen, bevor Sie sie löschen.

So löschen Sie eine Pipeline über die AWS CLI

Verwenden Sie zum Löschen einer Pipeline den Befehl delete-pipeline. Der folgende Befehl löscht die angegebene Pipeline.

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

Staging von Daten und Tabellen mit Pipeline-Aktivitäten

AWS Data Pipeline kann Eingabe- und Ausgabedaten in Ihren Pipelines bereitstellen, so dass bestimmte Aktivitäten, wie z. B `ShellCommandActivity` und `HiveActivity`, leichter zu verwenden sind.

Daten-Staging gibt Ihnen die Möglichkeit, Daten von einem Eingabedatenknoten in die Ressource zu kopieren, die die Aktivität ausführt, und anschließend genauso von der Resource in den Ausgabedatenknoten.

Die bereitgestellten Daten auf der Amazon EMR- oder Amazon EC2-Ressource sind verfügbar, indem spezielle Variablen in den Shell-Befehlen oder Hive-Skripten der Aktivität verwendet werden.

Tabellen-Staging ist mit Daten-Staging vergleichbar, außer dass die Daten speziell in Form von Datenbanktabellen bereitgestellt werden.

AWS Data Pipeline unterstützt die folgenden Staging-Szenarien:

- Daten-Staging mit `ShellCommandActivity`
- Tabellen-Staging mit Hive und zum Staging fähigen Datenknoten
- Tabellen-Staging mit Hive und nicht zum Staging fähigen Datenknoten

> **Note**
>
> Staging funktioniert nur, wenn das Feld `stage` für eine Aktivität wie etwa `ShellCommandActivity` auf `true` festgelegt ist. Weitere Informationen finden Sie unter ShellCommandActivity.

Darüber hinaus sind zwischen Datenknoten und Aktivitäten vier verschiedene Beziehungen möglich:

Staging von Daten lokal auf einer Ressource

Die Eingabedaten werden automatisch in das lokale Dateisystem der Ressource kopiert. Die Ausgabedaten werden automatisch aus dem lokalen Dateisystem auf den Ausgabedatenknoten kopiert. Wenn Sie z. B. `ShellCommandActivity`-Eingaben und -Ausgaben mit Staging = true konfigurieren, sind die Eingabedaten als INPUTx_STAGING_DIR und die Ausgabedaten als OUTPUTx_STAGING_DIR verfügbar, wobei x die Anzahl der Ein- oder Ausgaben ist.

Staging von Eingabe- und -Ausgabedefinitionen für eine Aktivität

Das Format der Eingabedaten (Spaltennamen und Tabellennamen) wird automatisch auf die Ressource der Aktivität kopiert. Beispielsweise, wenn Sie `HiveActivity` mit Staging = true konfigurieren. Das auf dem Eingabedatenknoten `S3DataNode` angegebene Datenformat wird zum Bereitstellen der Tabellendefinition aus der Hive-Tabelle verwendet.

Staging nicht aktiviert

Die Eingabe- und Ausgabe-Objekte und ihre Felder sind für die Aktivität zwar verfügbar, die Daten selbst jedoch nicht. Dies trifft z. B. bei `EmrActivity` standardmäßig zu. Andere Aktivitäten

müssen Sie dazu mit Staging = false konfigurieren. In einer solchen Konfiguration sind die Datenfelder verfügbar und die Aktivität kann mit der AWS Data Pipeline-Ausdruckssyntax auf sie verweisen, sofern die Abhängigkeit erfüllt ist. Dies dient alleinig der Abhängigkeitsprüfung. Der Code in der Aktivität veranlasst das Kopieren der Daten von der Eingabe auf die Ressource, auf der die Aktivität ausgeführt wird.

Abhängigkeitsbeziehung zwischen Objekten

Zwischen zwei Objekten besteht eine Abhängigkeitsbeziehung, die zu einer ähnlichen Situation führt, wie wenn Staging nicht aktiviert ist. In diesem Fall fungiert ein Datenknoten oder eine Aktivität als Vorbedingung für die Ausführung einer anderen Aktivität.

Data Staging mit ShellCommandActivity

Stellen Sie sich ein Szenario mit einer `ShellCommandActivity` mit `S3DataNode`-Objekten als Dateneingabe und -ausgabe vor. AWS Data Pipeline stellt die Datenknoten mit den Umgebungsvariablen `${INPUT1_STAGING_DIR}` und `${OUTPUT1_STAGING_DIR}` automatisch bereit, um sie für den Shell-Befehl zugänglich zu machen, als ob sie lokale Dateiordner wären. Dies wird im folgenden Beispiel veranschaulicht. Der numerische Teil der Variablen namens `INPUT1_STAGING_DIR` und `OUTPUT1_STAGING_DIR` wird abhängig von der Anzahl der Datenknoten erhöht, auf die Ihre Aktivität verweist.

> **Note**
>
> Dieses Szenario funktioniert nur wie beschrieben, wenn Ihre Dateneingabe- und -ausgaben `S3DataNode`-Objekte sind. Außerdem ist das Staging von Ausgabedaten nur zulässig, wenn für das `S3DataNode`-Ausgabeobjekt `directoryPath` festgelegt ist.

```
{
  "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')}"
  }
},
...
```

Tabellen-Staging mit Hive und zum Staging fähigen Datenknoten

Stellen Sie sich ein Szenario mit einer `HiveActivity` mit `S3DataNode`-Objekten als Dateneingabe und -ausgabe vor. AWS Data Pipeline stellt die Datenknoten mit den Variablen `${input1}` und `${output1}` automatisch bereit, um sie für das Hive-Skript zugänglich zu machen, als ob sie Hive-Tabellen wären. Dies wird im folgenden Beispiel für `HiveActivity` veranschaulicht. Der numerische Teil der Variablen namens `input` und `output` wird abhängig von der Anzahl der Datenknoten erhöht, auf die Ihre Aktivität verweist.

> **Note**
>
> Dieses Szenario funktioniert nur wie beschrieben, wenn die Dateneingaben und -ausgaben `S3DataNode`- oder `MySqlDataNode`-Objekte sind. Tabelle-Staging wird für `DynamoDBDataNode` nicht unterstützt.

```
{
```

```
  "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"
  }
},
...
```

Tabellen-Staging mit Hive und nicht zum Staging fähigen Datenknoten

Angenommen, eine `HiveActivity` wird für die Dateneingabe mit `DynamoDBDataNode` und für die Datenausgabe mit einem `S3DataNode`-Objekt verwendet. Da kein Data Staging verfügbar ist`DynamoDBDataNode`, müssen Sie die Tabelle zunächst manuell in Ihrem Hive-Skript erstellen und dabei den Variablennamen verwenden, um auf die DynamoDB-Tabelle `#{input.tableName}` zu

verweisen. Eine ähnliche Nomenklatur gilt, wenn die DynamoDB-Tabelle die Ausgabe ist, außer Sie verwenden eine Variable. `#{output.tableName}` Da in diesem Beispiel für das Ausgabeobjekt `S3DataNode` Staging verfügbar ist, können Sie unter `${output1}` auf den Ausgabedatenknoten verweisen.

> **Note**
>
> In diesem Beispiel verfügt die Variable für den Tabellennamen über das #-(Hash-)Zeichen als Präfix, da AWS Data Pipeline zum Zugriff auf `tableName` oder `directoryPath` Ausdrücke verwendet. Weitere Informationen darüber, wie die Ausdrucksauswertung in AWS Data Pipeline funktioniert, finden Sie unter Ausdrucksauswertung.

```
{
  "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"
  }
},
...
```

Verwenden einer Pipeline mit Ressourcen in mehreren Regionen

Standardmäßig werden die Ressourcen `Ec2Resource` und `EmrCluster` in derselben Region wie AWS Data Pipeline ausgeführt. AWS Data Pipeline unterstützt jedoch die Möglichkeit, Datenflüsse über mehrere Regionen hinweg zu orchestrieren, wie beispielsweise das Ausführen von Ressourcen in einer Region, die Eingabedaten aus einer anderen Region konsolidieren. Da Sie Ressourcen die Ausführung in einer bestimmten Region erlauben können, haben Sie auch die Flexibilität, Ressourcen zusammen mit ihren abhängigen Datensätzen in derselben Region anzusiedeln und die Leistung zu maximieren, indem Latenzen verringert und Kosten für regionsübergreifende Datenübertragungen vermieden werden. Sie können mit dem Feld `region` in `Ec2Resource` und `EmrCluster` Ressourcen so konfigurieren, dass sie in einer anderen Region als AWS Data Pipeline ausgeführt werden.

Die folgende Beispiel-Pipeline-JSON-Datei zeigt, wie eine `EmrCluster` Ressource in der Region Europa (Irland) ausgeführt wird, wobei davon ausgegangen wird, dass in derselben Region eine große Datenmenge vorhanden ist, an der der Cluster arbeiten soll. In diesem Beispiel ist der einzige Unterschied zu einer typischen Pipeline, dass der Wert des Feldes `region` für `EmrCluster` auf `eu-west-1` eingestellt ist.

```
{
  "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"
    }
  ]
}
```

In der folgenden Tabelle werden die Regionen aufgelistet, die Sie auswählen können, sowie die im Feld `region` zu verwendenden zugehörigen Regionscodes.

> **Note**
>
> Die folgende Liste enthält Regionen, in denen Workflows orchestriert und Amazon EMR- oder Amazon EC2-Ressourcen gestartet werden AWS Data Pipeline können. AWS Data Pipelinewird in diesen Regionen möglicherweise nicht unterstützt. Weitere Informationen zu den Regionen, in denen AWS Data Pipeline unterstützt wird, finden Sie unter Regionen und Endpunkte in AWS.

Name der Region	Regionscode
USA Ost (Nord-Virginia)	us-east-1
USA Ost (Ohio)	us-east-2
USA West (Nordkalifornien)	us-west-1
US West (Oregon)	us-west-2
Kanada (Zentral)	ca-central-1
Europa (Irland)	eu-west-1
Europe (London)	eu-west-2
Europa (Frankfurt)	eu-central-1
Asien-Pazifik (Singapur)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asien-Pazifik (Mumbai)	ap-south-1
Asien-Pazifik (Tokio)	ap-northeast-1
Asien-Pazifik (Seoul)	ap-northeast-2
South America (São Paulo)	sa-east-1

Cascading-Ausfälle und erneute Ausführungen

Mit AWS Data Pipeline können Sie konfigurieren, wie sich Pipeline-Objekte verhalten, wenn eine Abhängigkeit ausfällt oder von einem Benutzer storniert wird. Sie können sicherstellen, dass Ausfälle zu anderen Pipeline-Objekten (Verbrauchern) kaskadieren, um ein unendlich langes Warten zu verhindern. Für alle Aktivitäten, Datenknoten und Vorbedingungen ist ein Feld namens `failureAndRerunMode` mit dem Standardwert `none` vorhanden. Um das Kaskadieren von Ausfällen zu aktivieren, stellen Sie das Feld `failureAndRerunMode` auf `cascade` ein.

Wenn dieses Feld aktiviert ist, treten kaskadierende Ausfälle auf, wenn ein Pipeline-Objekt im Zustand `WAITING_ON_DEPENDENCIES` blockiert wird und alle Abhängigkeiten ohne ausstehenden Befehl fehlgeschlagen sind. Während eines kaskadierenden Ausfalls treten die folgenden Ereignisse ein:

- Wenn ein Objekt fehlschlägt, wird für seine Verbraucher `CASCADE_FAILED` eingestellt und für die Vorbedingungen des Originalobjekts und seiner Verbraucher wird `CANCELED` festgelegt.
- Alle Objekte, die sich bereits im Zustand `FINISHED`, `FAILED`oder `CANCELED` befinden, werden ignoriert.

Das Kaskadieren von Ausfällen funktioniert nicht bei Abhängigkeiten (stromaufwärts) eines ausgefallenen Objekts, außer bei zugehörigen Vorbedingungen des ursprünglichen ausgefallenen Objekts. Pipeline-Objekte, die vom Kaskadieren eines Ausfalls betroffen sind, können beliebig viele Wiederholungen oder Post-Aktionen, wie z. B. `onFail`, auslösen.

Die detaillierten Auswirkungen eines kaskadierenden Ausfalls sind vom Objekttyp abhängig.

Aktivitäten

Der Zustand einer Aktivität ändert sich in `CASCADE_FAILED`, wenn irgendwelche seiner Abhängigkeiten ausfallen, und es löst anschließend einen kaskadierenden Ausfall bei den Verbrauchern der Aktivität aus. Falls eine Ressource ausfällt, von der die Aktivität abhängig ist, befindet sich die Aktivität im Zustand `CANCELED` und der Zustand aller Verbraucher ändert sich in `CASCADE_FAILED`.

Datenknoten und Voraussetzungen

Wenn als Ausgabe einer ausgefallenen Aktivität ein Datenknoten konfiguriert ist, ändert sich der Zustand des Datenknotens in `CASCADE_FAILED`. Der Ausfall eines Datenknotens wird auf alle zugehörigen Vorbedingungen übertragen, deren Zustand sich in `CANCELED` ändert.

Ressourcen

Wenn sich die Objekte, die von einer Ressource abhängig sind, im Zustand `FAILED` befinden, während sich die Ressource im Zustand `WAITING_ON_DEPENDENCIES` befindet, dann ändert sich der Zustand der Ressource in `FINISHED`.

Objekte mit kaskadierendem Ausfall erneut ausführen

Standardmäßig wird bei der erneuten Ausführung einer Aktivität oder eines Datenknotens nur die zugehörige Ressource erneut ausgeführt. Wird jedoch das Feld `failureAndRerunMode` in einem Pipeline-Objekt auf `cascade` eingestellt, ermöglicht dies unter den folgenden Bedingungen, dass die erneute Ausführung auf einem Zielobjekt auf alle Verbraucher übertragen wird:

- Die Verbraucher des Zielobjekts befinden sich im Zustand `CASCADE_FAILED`.
- Für die Abhängigkeiten des Zielobjekts stehen keine Befehle für eine erneute Ausführung an.
- Die Abhängigkeiten des Zielobjekts befinden sich nicht im Zustand `FAILED`, `CASCADE_FAILED` oder `CANCELED`.

Bei dem Versuch, ein `CASCADE_FAILED`-Objekt erneut auszuführen, bei dem sich irgendwelche seiner Abhängigkeiten im Zustand `FAILED`, `CASCADE_FAILED` oder `CANCELED` befinden, schlägt die erneute Ausführung fehl und das Objekt kehrt wieder in den Zustand `CASCADE_FAILED` zurück. Zur erfolgreichen erneuten Ausführung des ausgefallenen Objekts müssen Sie den Ausfall in der Kette der Abhängigkeiten nach oben verfolgen, um die ursprüngliche Quelle des Ausfalls zu finden, und statt dessen das betreffende Objekt erneut ausführen. Wenn Sie den Befehl für eine erneute Ausführung auf einer Ressource ausgeben, versuchen Sie, alle von ihr abhängigen Ressourcen ebenfalls erneut auszuführen.

Kaskadenausfall und Füllungen

Wenn Sie kaskadierende Ausfälle aktivieren und eine Pipeline viele Abgleichungen erstellt, kann es aufgrund von Laufzeitfehlern der Pipeline sein, dass Ressourcen in rascher Abfolge erstellt und gelöscht werden, ohne effektive Arbeit zu leisten. AWS Data Pipeline versucht, Sie beim Speichern einer Pipeline mit der folgenden Warnmeldung auf diese Situation aufmerksam zu machen: *`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.` Dies geschieht, da kaskadierende Ausfälle schnell als Zustand von Downstream-Aktivitäten `CASCADE_FAILED` einstellen und nicht mehr benötigte EMR-Cluster und EC2-Ressourcen herunterfahren können. Wir empfehlen, dass Sie Pipelines mit kurzen Zeitbereichen testen, um die Auswirkungen dieser Situation zu begrenzen.

Syntax der Pipeline-Definitionsdatei

Die Anleitungen in diesem Abschnitt sind für das manuelle Arbeiten mit Pipeline-Definitionsdateien über die AWS Data Pipeline-Befehlszeilenschnittstelle (CLI) bestimmt. Dies ist eine Alternative zur interaktiven Entwicklung einer Pipeline über die AWS Data Pipeline-Konsole.

Sie können Pipeline-Definitionsdateien mit jedem beliebigen Texteditor manuell erstellen, der das Speichern von Dateien im UTF-8-Dateiformat und das Senden der Dateien über die AWS Data Pipeline-Befehlszeilenschnittstelle unterstützt.

AWS Data Pipeline unterstützt in Pipeline-Definitionen auch eine Vielzahl von komplexen Ausdrücken und Funktionen. Weitere Informationen finden Sie unter Pipeline-Ausdrücke und -Funktionen.

Dateistruktur

Der erste Schritt bei der Erstellung einer Pipeline besteht darin, Pipeline-Definitionsobjekte in einer Pipeline-Definitionsdatei zu verfassen. Im folgenden Beispiel wird die allgemeine Struktur einer Pipeline-Definitionsdatei veranschaulicht. Diese Datei definiert zwei Objekte, die durch '{' und '}' begrenzt und durch ein Komma getrennt werden.

Im folgenden Beispiel definiert das erste Objekt zwei Namen-Wert-Paare, die als Felder bezeichnet werden. Das zweite Objekt definiert drei Felder.

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

Beim Erstellen einer Pipeline-Definitionsdatei müssen Sie die erforderlichen Typen von Pipeline-Objekten auswählen, sie der Pipeline-Definitionsdatei hinzufügen und dann die entsprechenden Felder hinzufügen. Weitere Informationen zu Pipeline-Objekten finden Sie unter Pipeline-Objektreferenz.

So können Sie beispielsweise ein Pipeline-Definitionsobjekt für einen Eingabedatenknoten und ein anderes für den Ausgabedatenknoten erstellen. Erstellen Sie dann ein weiteres Pipeline-Definitionsobjekt für eine Aktivität, z. B. die Verarbeitung der Eingabedaten mit Amazon EMR.

Pipeline-Felder

Wenn Sie wissen, welche Objekttypen in Ihre Pipeline-Definitionsdatei aufzunehmen sind, fügen Sie die betreffenden Felder zur Definition der einzelnen Pipeline-Objekte hinzu. Feldnamen stehen in Anführungszeichen und sind wie im folgenden Beispiel veranschaulicht durch ein Leerzeichen, einen Doppelpunkt und ein Leerzeichen von den Feldwerten getrennt.

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

Bei dem Feldwert kann es sich um eine Zeichenfolge, einen Verweis auf ein anderes Objekt, einen Funktionsaufruf, einen Ausdruck oder eine geordnete Liste beliebiger der oben genannten Typen handeln. Weitere Informationen über die Arten von Daten, die für Feldwerte verwendet werden können, finden Sie unter Einfache Datentypen. Weitere Informationen zu Funktionen, die zur Auswertung von Feldwerten verwendet werden können, finden Sie unter Ausdrucksauswertung.

Felder sind auf 2048 Zeichen begrenzt. Da die Größe von Objekten 20 KB betragen kann, können einem Objekt nicht viele große Feder hinzugefügt werden.

Jedes Pipeline-Objekt muss die folgenden Felder enthalten: `id` und `type`, wie im folgenden Beispiel dargestellt. Je nach Objekttyp werden möglicherweise noch andere Felder benötigt. Wählen Sie einen Wert für `id` aus, der für Sie bedeutsam und in der Pipeline-Definition eindeutig ist. Der Wert für `type` gibt den Typ des Objekts an. Geben Sie einen der unterstützten Objekttypen für die Pipeline-Definition ein, die im Thema Pipeline-Objektreferenz aufgelistet werden.

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

Weitere Informationen zu den erforderlichen und optionalen Feldern eines jeden Objekts finden Sie in der Dokumentation für das Objekt.

Wenn Sie Felder aus einem Objekt in ein anderes Objekt einschließen möchten, verwenden Sie das Feld `parent` mit einem Verweis auf das Objekt. Beispiel: Objekt „B“ enthält seine Felder, „B1“ und „B2“, sowie die Felder von Objekt „A“, „A1“ und „A2“.

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

Sie können gemeinsame Felder in einem Objekt mit der ID „Default“ definieren. Diese Felder werden automatisch in die Pipeline-Definitionsdatei eines jeden Objekts eingeschlossen, sofern für das Feld `parent` nicht ausdrücklich ein Verweis auf ein anderes Objekt festgelegt ist.

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

Benutzerdefinierte Felder

Sie können für Ihre Pipeline-Komponenten benutzerdefinierte Felder erstellen und mit Ausdrücken auf sie verweisen. Das folgende Beispiel zeigt ein benutzerdefiniertes Feld, das einem DataNode S3-Objekt benannt `myCustomField` und diesem `my_customFieldReference` hinzugefügt wurde:

```
{
  "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"}
  },
```

Der Name eines benutzerdefinierten Feldes muss als Präfix das Wort „my“ in Kleinbuchstaben gefolgt von einem Großbuchstaben und einem Unterstrich besitzen. Darüber hinaus kann ein benutzerdefiniertes Feld wie im voranstehenden `myCustomField`-Beispiel ein Zeichenfolgenwert

oder wie im voranstehenden `my_customFieldReference`-Beispiel ein Verweis auf eine andere Pipeline-Komponente sein.

> **Note**
>
> AWS Data Pipeline prüft benutzerdefinierte Felder nur auf gültige Verweise auf andere Pipeline-Komponenten und nicht auf von Ihnen in benutzerdefinierten Feldern hinzugefügte Zeichenfolgenwerte.

Arbeiten mit der API

> **Note**
>
> Wenn Sie keine Anwendungen programmieren, die mit AWS Data Pipeline interagieren, müssen Sie keine AWS-SDKs installieren. Sie können über die Konsole oder die Befehlszeile Pipelines erstellen und ausführen. Weitere Informationen finden Sie unterEinrichten für AWS Data Pipeline

Am einfachsten können Sie Anwendungen programmieren, die mit interagierenAWS Data Pipelineoder einen benutzerdefinierten Task Runner implementieren, indem Sie eines der AWS-SDKs verwenden. Die Funktionen der AWS-SDKs vereinfachen das Aufrufen der Web-Service-APIs von Ihrer bevorzugten Programmierumgebung. Weitere Informationen finden Sie unter Installieren des AWS-SDKs .

Installieren des AWS-SDKs

Die AWS-SDKs stellen Wrapper-Funktionen für die API bereit und übernehmen viele der Verbindungsdetails, wie Berechnen der Signaturen, Umgang mit Anforderungswiederholungen und Fehlerbehandlung. Die SDKs enthalten außerdem Beispiel-Code, Tutorials und weitere Ressourcen, die Sie beim Schreiben von Anwendungen unterstützen. die AWS aufrufen. Durch Aufrufen der Wrapper-Funktionen in einem SDK kann der Prozess zum Schreiben einer AWS-Anwendung erheblich vereinfacht werden. Weitere Informationen zum Herunterladen und Verwenden von AWS-SDKs finden Sie unter Beispiel-Code und Bibliotheken.

AWS Data Pipeline-Unterstützung ist in SDKs für die folgenden Plattformen verfügbar:

- AWS SDK für Java

- AWS SDK for Node.js
- AWS SDK für PHP
- AWS SDK für Python (Boto)
- AWS SDK für Ruby
- AWS SDK für .NET

Erstellen einer HTTP-Anforderung an AWS Data Pipeline

Eine vollständige Beschreibung der programmgesteuerten Objekte in AWS Data Pipeline finden Sie in der AWS Data Pipeline-API-Referenz.

Wenn Sie keinen AWS-SDK verwenden, können Sie AWS Data Pipeline-Operationen über HTTP ausführen, indem Sie die POST-Anforderungsmethode verwenden. Bei der POST-Methode müssen Sie den Vorgang im Header der Anforderung festlegen und im Anforderungstext die Daten für den Vorgang im JSON-Format angeben.

Inhalt des HTTP-Headers

AWS Data Pipeline benötigt die folgenden Informationen im Header einer HTTP-Anforderung:

- `host` Den AWS Data Pipeline-Endpunkt.

  Weitere Informationen zu Endpunkten finden Sie unter Regionen und Endpunkte.
- `x-amz-date` Sie müssen den Zeitstempel entweder im HTTP-Datums-Header oder im AWS-x-amz-date-Header angeben. (Einige HTTP-Client-Bibliotheken lassen den Datums-Header nicht zu.) Ist der x-amz-date-Header vorhanden, ignoriert das System bei der Anforderungsauthentifizierung alle Datums-Header.

  Das Datum muss in einem der folgenden drei Formate angegeben werden, wie in HTTP/1.1 RFC festgelegt:
  - Sun, 06 Nov 1994 08:49:37 GMT (RFC 822, aktualisiert durch RFC 1123)
  - Sunday, 06-Nov-94 08:49:37 GMT (RFC 850, abgelöst durch RFC 1036)
  - Sun Nov 6 08:49:37 1994 (ANSI C asctime()-Format)
- `Authorization` Der Satz an Autorisierungsparametern, mit denen AWS die Gültigkeit und Authentizität der Anforderung sicherstellt. Weitere Informationen zum Aufbau dieses Headers finden Sie unter Signature Version 4-Signaturprozess.

- `x-amz-target` Der Zieldienst der Anforderung und des Vorgangs für die Daten, in folgendem Format: `<<serviceName>>_<<API version>>.<<operationName>>`

  Beispiel, `DataPipeline_20121129.ActivatePipeline`
- `content-type` Gibt JSON und die Version an. Beispiel, `Content-Type: application/x-amz-json-1.0`

Nachfolgend finden Sie einen Beispiel-Header für eine HTTP-Anforderung zum Aktivieren einer 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
```

HTTP-Textinhalt

Der Textkörper einer HTTP-Anforderung enthält die Daten für den Vorgang, der im Header der HTTP-Anforderung festgelegt ist. Die Daten müssen für jede AWS Data Pipeline-API entsprechend dem JSON-Datenschema formatiert werden. Das AWS Data Pipeline-JSON-Datenschema definiert die Datentypen und Parameter (z. B. Vergleichsoperatoren und Aufzählungskonstanten), die für die einzelnen Vorgänge verfügbar sind.

Format des Textkörpers einer HTTP-Anforderung

Verwenden Sie das JSON-Datenformat zur gleichzeitigen Übermittlung von Datenwerten und -strukturen. Elemente können mit der Klammerschreibweise innerhalb anderer Elemente verschachtelt werden. Das folgende Beispiel zeigt eine Anforderung für das Erstellen einer Pipeline-Definition, die aus drei Objekten und ihren entsprechenden Slots besteht.

```
{
 "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"}

      ]
    }
  ]
}
```

Handhaben der HTTP-Antwort

Nachfolgend finden Sie einige wichtige Header in der HTTP-Antwort und Informationen dazu, wie Sie diese Header in Ihrer Anwendung behandeln sollten:

- HTTP/1.1— Auf diesen Header folgt ein Statuscode. Ein Code-Wert von 200 gibt an, dass ein Vorgang erfolgreich war. Jeder andere Wert weist auf einen Fehler hin.
- x-amzn-RequestId – Dieser Header enthält eine Anforderungs-ID, die Sie für die Fehlerbehebung bei einer Anfrage bei AWS Data Pipeline nutzen können. Ein Beispiel für eine Anforderungs-ID ist K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG.
- x-amz-crc32 – AWS Data Pipeline berechnet eine CRC32-Prüfsumme der HTTP-Nutzlast und gibt diese Prüfsumme an den x-amz-crc32-Header zurück. Wir empfehlen Ihnen, Ihre eigene CRC32-Prüfsumme clientseitig zu berechnen und sie mit dem x-amz-crc32-Header zu vergleichen. Wenn die Prüfsummen nicht übereinstimmen, kann dies ein Hinweis darauf sein, dass die Daten während der Übertragung beschädigt wurden. Wenn dies der Fall ist, sollten Sie Ihre Anforderung erneut übermitteln.

AWS-SDK-Benutzer müssen diese Verifizierung nicht manuell ausführen, da die SDKs die Prüfsumme jeder Antwort von Amazon DynamoDB berechnen und den Versuch automatisch wiederholen, wenn keine Übereinstimmung vorliegt.

Beispiel AWS Data Pipeline, JSON-Anforderung und -Antwort

Die folgenden Beispiele zeigen eine Anforderung zum Erstellen einer neuen Pipeline. Anschließend wird die AWS Data Pipeline-Antwort angezeigt, einschließlich der Pipeline-ID der neu erstellten Pipeline.

HTTP-POST-Anforderung

```
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"}
```

AWS Data Pipeline-Antwort

```
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"}
```

Sicherheit in AWS Data Pipeline

Die Sicherheit in der Cloud hat AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud selbst – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die für AWS Data Pipeline gelten, finden Sie unter Betroffene AWS-Services nach Compliance-Programm.
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation erläutert, wie das Modell der geteilten Verantwortung bei der Verwendung von AWS Data Pipeline zum Tragen kommt. Die folgenden Themen veranschaulichen, wie Sie AWS Data Pipeline zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren auch, wie Sie andere AWS-Services nutzen können, die Ihnen bei der Überwachung und Sicherung Ihrer AWS Data Pipeline-Ressourcen helfen.

Themen

Datenschutz in AWS Data Pipeline

Das Modell der geteilten Verantwortung von AWS gilt für den Datenschutz in AWS Data Pipeline. Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieser Inhalt enthält die Sicherheitskonfigurations- und Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS-Modell der geteilten Verantwortung und die GDPR im Blog zur AWS-Sicherheit.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Factor Authentication (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir empfehlen TLS 1.2 oder höher.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.
- AWS Data Pipelineunterstützt IMDSv2 für Amazon EMR- und Amazon EC2 EC2-Ressourcen. Um IMDSv2 mit Amazon EMR zu verwenden, verwenden Sie die Versionen 5.23.1, 5.27.1 oder 5.32 oder höher oder Version 6.2 oder höher. Weitere Informationen finden Sie unter Konfigurieren von Metadatendienstanforderungen für Amazon EC2 EC2-Instances und Verwenden von IMDSv2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie unter Verwendung der Konsole, der API, AWS CLI oder AWS SDKs mit AWS Data Pipeline oder

anderen AWS-Services arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Identity and Access Management für AWS Data Pipeline

Mit Ihren Sicherheitsanmeldeinformationen identifizieren Sie sich bei den Services in AWS und erhalten Berechtigungen zur Verwendung von AWS-Ressourcen, wie etwa Ihre Pipelines. Sie können Funktionen vonAWS Data Pipeline undAWS Identity and Access Management (IAM) verwenden, um anderen Benutzern den Zugriff auf IhreAWS Data Pipeline Ressourcen zu erlaubenAWS Data Pipeline, ohne Ihre Sicherheitsanmeldeinformationen freizugeben.

In Unternehmen können bestimmte Pipelines für mehrere Mitarbeiter freigegeben werden, damit diese gemeinsam damit arbeiten können. Allerdings sollten in diesem Fall Maßnahmen wie die folgenden ergriffen werden:

- Steuern Sie, welche Benutzer auf bestimmte Pipelines zugreifen können
- Schützen einer Produktions-Pipeline vor versehentlichen Änderungen
- Erlauben des Lesezugriffs von Auditoren auf Pipelines und gleichzeitiges Verhindern von Änderungen

AWS Data Pipelineist inAWS Identity and Access Management (IAM) integriert, das eine Vielzahl von Funktionen bietet:

- Erstellen von Benutzern und Gruppen in IhremAWS-Konto.
- Teilen Sie IhreAWS Ressourcen ganz einfach zwischen den Benutzern in IhremAWS-Konto.
- Zuweisen eindeutiger Sicherheitsanmeldeinformationen zu jedem Benutzer.
- Kontrollieren Sie den Zugriff jedes Benutzers auf Services und Ressourcen.
- Erhalten Sie eine einzige Rechnung für alle Benutzer in IhremAWS-Konto.

Wenn Sie IAM zusammen mit verwendenAWS Data Pipeline, können Sie steuern, ob Benutzer im Unternehmen Aufgaben mit bestimmten -API-Aktionen ausführen und spezifische AWS-Ressourcen verwenden können. Sie können IAM-Richtlinien verwenden, die auf Pipeline-Tags und Worker-

Gruppen basieren, um Ihre Pipelines mit anderen Benutzern zu teilen und deren Zugriffsebene zu kontrollieren.

Inhalt

IAM-Richtlinien für AWS Data Pipeline

IAM-Entitäten verfügen standardmäßig nicht über die Berechtigung zur Erstellung oder Änderung von AWS-Ressourcen. Damit IAM-Benutzer Ressourcen erstellen oder ändern und Aufgaben ausführen können, müssen Sie IAM-Richtlinien erstellen und so den IAM-Benutzern die Berechtigung zur Nutzung der benötigten Ressourcen und API-Aktionen erteilen. Diese Richtlinien ordnen Sie dann den IAM-Benutzern zu, die diese Berechtigungen benötigen.

Wenn Sie einem Benutzer oder einer Benutzergruppe eine Richtlinie zuordnen, wird den Benutzern die Ausführung der angegebenen Aufgaben für die angegebenen Ressourcen gestattet oder verweigert. Allgemeine Informationen zu IAM-Richtlinien finden Sie unter Berechtigungen und Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen zum Verwalten und Erstellen von benutzerdefinierten IAM-Richtlinien finden Sie unter Verwalten von IAM-Richtlinien.

Inhalt

Richtliniensyntax

Eine IAM-Richtlinie ist ein JSON-Dokument, das eine oder mehrere Anweisungen enthält. Jede Anweisung ist folgendermaßen strukturiert:

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"*",
```

```
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
  ]
}
```

Eine Anweisung in einer Richtlinie besteht aus folgenden Elementen:

- Effect: Der effect-Wert kann `Allow` oder `Deny` lauten. IAM-Entitäten verfügen standardmäßig nicht über die Berechtigung zur Verwendung von Ressourcen und API-Aktionen. Daher werden alle Anfragen abgelehnt. Dieser Standardwert kann durch eine explizite Zugriffserlaubnis überschrieben werden. Eine explizite Zugriffsverweigerung überschreibt jedwede Zugriffserlaubnis.
- Action: Mit action wird die API-Aktion spezifiziert, für die Sie Berechtigungen erteilen oder verweigern. Eine Liste der Aktionen fürAWS Data Pipeline finden Sie unter Aktionen in der AWS Data PipelineAPI-Referenz.
- Resource: Die von einer Aktion betroffene Ressource. Der einzige hier zulässige Wert lautet `"*"`.
- Condition: Bedingungen sind optional. Mit ihrer Hilfe können Sie bestimmen, wann Ihre Richtlinie wirksam wird.

  AWS Data Pipeline implementiert die AWS-weiten Kontextschlüssel (siehe Verfügbare Schlüssel für Bedingungen) und zusätzlich die folgenden servicespezifischen Schlüssel.

  - `datapipeline:PipelineCreator`— Um dem Benutzer, der die Pipeline erstellt hat, Zugriff zu erteilen. Ein Beispiel hierzu finden Sie unter Gewähren des vollen Zugriffs für Pipeline-Eigentümer.
  - `datapipeline:Tag`— Um Zugriff auf der Grundlage von Pipeline-Tagging zu gewähren. Weitere Informationen finden Sie unter Steuern des Zugriffs auf Pipelines mithilfe von Tags.
  - `datapipeline:workerGroup`— Um Zugriff auf der Grundlage des Namens der Arbeitsgruppe zu gewähren. Weitere Informationen finden Sie unter Steuern des Zugriffs auf Pipelines mithilfe von Worker-Gruppen.

Steuern des Zugriffs auf Pipelines mithilfe von Tags

Sie können IAM-Richtlinien erstellen, die auf die Tags für Ihre Pipeline verweisen. Dadurch lässt sich mithilfe von Pipeline-Tags Folgendes durchführen:

- Gewähren des Lesezugriffs auf eine Pipeline
- Gewähren des Lese-/Schreibzugriffs auf eine Pipeline
- Blockieren des Zugriffs auf eine Pipeline

Nehmen wir an, dass es in einem Unternehmen zwei Pipeline-Umgebungen (Produktion und Entwicklung) und für jede Umgebung eine IAM-Gruppe gibt. Für Pipelines in der Produktionsumgebung gewährt der Manager Benutzern in der Produktions-IAM-Gruppe Lese-/Schreibzugriff, Benutzern in der Entwickler-IAM-Gruppe jedoch nur Lesezugriff. Für Pipelines in der Entwicklungsumgebung gewährt der Manager Lese-/Schreibzugriff sowohl für die Produktions- als auch für die Entwickler-IAM-Gruppe.

Um dieses Szenario zu erreichen, kennzeichnet der Manager die Produktionspipelines mit dem Tag „environment=production“ und fügt der Entwickler-IAM-Gruppe die folgende Richtlinie hinzu. Die erste Anweisung gewährt Lesezugriff auf alle Pipelines. Die zweite Anweisung gewährt Lese-/Schreibzugriff auf Pipelines, die nicht mit dem Tag "environment=production" gekennzeichnet sind.

```
{
  "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"}
      }
    }
  ]
}
```

Darüber hinaus verknüpft der Manager die folgende Richtlinie mit der Produktions-IAM-Gruppe. Diese Anweisung gewährt vollständigen Zugriff auf alle Pipelines.

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

Weitere Beispiele finden Sie unter Gewähren des Lesezugriffs für Benutzer basierend auf einem Tag und Gewähren des vollständigen Zugriffs für Benutzer basierend auf einem Tag.

Steuern des Zugriffs auf Pipelines mithilfe von Worker-Gruppen

Sie können IAM-Richtlinien erstellen, die auf Arbeitsgruppennamen verweisen.

Nehmen wir an, dass es in einem Unternehmen zwei Pipeline-Umgebungen (Produktion und Entwicklung) und für jede Umgebung eine IAM-Gruppe gibt. Außerdem sind drei Datenbankserver mit Task Runner-Anwendungen für die Produktions-, die Vorproduktions- und die Entwicklungsumgebung vorhanden. Der Manager möchte sicherstellen, dass Benutzer in der Produktions-IAM-Gruppe Pipelines erstellen können, die Aufgaben an Produktionsressourcen weiterleiten, und dass Benutzer in der Entwicklungs-IAM-Gruppe Pipelines erstellen können, die Aufgaben sowohl an Vorproduktions- als auch an Entwicklerressourcen weiterleiten.

Um dies zu erreichen, installiert er Task Runner auf den Produktionsressourcen mit den Anmeldeinformationen der Produktionsgruppe und weist `workerGroup` den Wert "prodresource" zu. Außerdem installiert der Manager Task Runner auf den Entwicklungsressourcen mit den Anmeldeinformationen der Entwicklungsgruppe und weist `workerGroup` die Werte "pre-production" und "development" zu. Der Manager fügt der Entwickler-IAM-Gruppe die folgende Richtlinie hinzu, um den Zugriff auf „Prodresource“ -Ressourcen zu blockieren. Die erste Anweisung gewährt Lesezugriff auf alle Pipelines. Die zweite Anweisung erteilt der Worker-Gruppe mit dem Namenspräfix "dev" oder "pre-prod" den Lese-/Schreibzugriff auf die Pipelines.

```
{
  "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*"]
        }
      }
    }
  ]
}
```

Darüber hinaus fügt der Manager der Produktions-IAM-Gruppe die folgende Richtlinie hinzu, um Zugriff auf „Prodresource“ -Ressourcen zu gewähren. Die erste Anweisung gewährt Lesezugriff auf alle Pipelines. Die zweite Anweisung gewährt der Worker-Gruppe mit dem Namenspräfix "prod" den Lese-/Schreibzugriff.

```
{
  "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*"}
      }
    }
  ]
}
```

Beispielrichtlinien für AWS Data Pipeline

Die folgenden Beispiele zeigen, wie Sie Benutzern vollständigen oder eingeschränkten Zugriff auf Pipelines gewähren.

Inhalt

Beispiel 1: Gewähren des Lesezugriffs für Benutzer basierend auf einem Tag

Die folgende Richtlinie gestattet Benutzern die Verwendung von AWS Data Pipeline-API-Aktionen, die Lesezugriffe durchführen, allerdings nur bei Pipelines mit dem Tag "environment=production".

Die ListPipelines API-Aktion unterstützt keine Tag-basierte Autorisierung.

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

```
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "datapipeline:Tag/environment": "production"
        }
      }
    }
  ]
}
```

Beispiel 2: Gewähren des vollständigen Zugriffs für Benutzer basierend auf einem Tag

Die folgende Richtlinie ermöglicht es Benutzern, alleAWS Data Pipeline API-Aktionen zu verwenden, mit Ausnahme von ListPipelines, aber nur mit Pipelines, die das Tag „environment=test“ haben.

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

Beispiel 3: Gewähren des vollen Zugriffs für Pipeline-Eigentümer

Die folgende Richtlinie gestattet Benutzern die Verwendung aller AWS Data Pipeline-API-Aktionen, allerdings nur bei ihren eigenen Pipelines.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "datapipeline:PipelineCreator": "${aws:userid}"
        }
      }
    }
  ]
}
```

Beispiel 4: Benutzern den Zugriff auf die AWS Data Pipeline-Konsole gewähren

Die folgende Richtlinie gestattet Benutzern, mit der AWS Data Pipeline-Konsole eine Pipeline zu erstellen und zu verwalten.

Die Richtlinie enthält die Aktion für `PassRole`-Berechtigungen für spezifische Ressourcen, die an den `role`ARN gebunden sind, den AWS Data Pipeline benötigt. Weitere Informationen zur identitätsbasierten (IAM)`PassRole` -Berechtigung finden Sie im Blogbeitrag Granting Permission to Launch EC2 Instances with IAM Roles (PassRolePermission).

```
{
 "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"
   ]
  }
 ]
}
```

IAM-Rollen für AWS Data Pipeline

AWS Data PipelineverwendetAWS Identity and Access Management Rollen. Die den IAM-Rollen zugeordneten Berechtigungsrichtlinien bestimmen, welche AktionenAWS Data Pipeline und Ihre Anwendungen ausführen können und auf welcheAWS Ressourcen sie zugreifen können. Weitere Informationen finden Sie unter IAM-Rollen im IAM-Benutzerhandbuch.

AWS Data Pipelineerfordert zwei IAM-Rollen:

- Die Pipeline-Rolle steuertAWS Data Pipeline den Zugriff auf Ihre AWS-Ressourcen. In Pipeline-Objektdefinitionen gibt das`role` Feld diese Rolle an.
- Die EC2-Instance-Rolle steuert den Zugriff, den Anwendungen, die auf EC2-Instances ausgeführt werden, einschließlich der EC2-Instances in Amazon EMR-Clustern, aufAWS Ressourcen haben. In Pipeline-Objektdefinitionen gibt das`resourceRole` Feld diese Rolle an.

> **Important**
>
> Wenn Sie vor dem 3. Oktober 2022 eine Pipeline mithilfe derAWS Data Pipeline Konsole mit StandardrollenAWS Data Pipeline erstellt haben, haben Sie die`DataPipelineDefaultRole` für Sie erstellt und die`AWSDataPipelineRole` verwaltete Richtlinie an die Rolle angehängt. Ab dem 3. Oktober 2022 ist die`AWSDataPipelineRole` verwaltete Richtlinie veraltet und die Pipeline-Rolle muss für eine Pipeline angegeben werden, wenn Sie die Konsole verwenden.
> Wir empfehlen Ihnen, die vorhandenen Pipelines zu überprüfen und festzustellen, ob die der Pipeline zugeordnet`DataPipelineDefaultRole``AWSDataPipelineRole` ist und ob sie dieser Rolle zugeordnet ist. Wenn ja, überprüfen Sie den Zugriff, den diese Richtlinie gewährt, um sicherzustellen, dass er Ihren Sicherheitsanforderungen entspricht. Fügen Sie die dieser Rolle zugeordneten Richtlinien und Richtlinienerklärungen nach Bedarf hinzu, aktualisieren Sie sie oder ersetzen Sie sie. Alternativ können Sie eine Pipeline aktualisieren, um eine Rolle zu verwenden, die Sie mit unterschiedlichen Berechtigungsrichtlinien erstellen.

Beispiel für Berechtigungsrichtlinien fürAWS Data Pipeline Rollen

Jeder Rolle sind eine oder mehrere Berechtigungsrichtlinien zugeordnet, die festlegen, auf welcheAWS Ressourcen die Rolle zugreifen kann, und welche Aktionen die Rolle ausführen kann. Dieses Thema enthält ein Beispiel für eine Berechtigungsrichtlinie für die Pipeline-Rolle. Es enthält auch den Inhalt von`AmazonEC2RoleforDataPipelineRole`, der die verwaltete Richtlinie für die standardmäßige EC2-Instanzrolle ist`DataPipelineDefaultResourceRole`.

Beispiel für eine Berechtigungsrichtlinie für -Rolle

Die folgende Beispielrichtlinie ist so konzipiert, dass sie grundlegende Funktionen zulässt, die den Betrieb einer Pipeline mit Amazon EC2- und Amazon EMR-RessourcenAWS Data Pipeline erfordern. Es bietet auch Berechtigungen für den Zugriff auf andereAWS Ressourcen wie Amazon Simple Storage Service und Amazon Simple Notification Service, die viele Pipelines benötigen. Wenn die in einer Pipeline definierten Objekte nicht die Ressourcen einesAWS Dienstes benötigen, wird dringend empfohlen, die Berechtigungen für den Zugriff auf diesen Dienst zu entfernen. Wenn Ihre Pipeline beispielsweise keineSnsAlarm Aktion definiertDynamoDB DataNode oder verwendet, empfehlen wir, dass Sie die Erlaubungsanweisungen für diese Aktionen entfernen.

- Ersetzen Sie *111122223333* durch Ihre AWS-Konto-ID.

- Ersetzen Sie*NameOfDataPipelineRole* durch den Namen der Pipeline-Rolle (die Rolle, an die diese Richtlinie angehängt ist).
- Ersetzen Sie*NameOfDataPipelineResourceRole* durch den Namen der EC2-Instanzrolle.
- *us-west-1*Ersetzen Sie es durch die entsprechende Region für Ihre Anwendung.

```
{
    "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"
            ]
        }
    ]
}
```

Verwaltete Standardrichtlinie für die EC2-Instanzrolle

Der Inhalt von`AmazonEC2RoleforDataPipelineRole` wird nachstehend dargestellt.
Dies ist die verwaltete Richtlinie, die der Standardressourcenrolle fürAWS Data Pipeline, zugeordnet ist`DataPipelineDefaultResourceRole`. Wenn Sie eine Ressourcenrolle für Ihre Pipeline definieren, empfehlen wir, mit dieser Berechtigungsrichtlinie zu beginnen und dann die Berechtigungen fürAWS Serviceaktionen zu entfernen, die nicht erforderlich sind.

Version 3 der Richtlinie wird angezeigt, die zum Zeitpunkt der Erstellung dieses Artikels die neueste Version ist. Sehen Sie sich die neueste Version der Richtlinie mithilfe der IAM-Konsole an.

```
{
  "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": ["*"]
    }]
}
```

IAM-Rollen für Rollenberechtigungen erstellenAWS Data Pipeline und bearbeiten

Gehen Sie wie folgt vor, um Rollen für dieAWS Data Pipeline Verwendung der IAM-Konsole zu erstellen. Der Prozess besteht aus zwei Schritten. Erstellen Sie zunächst eine Berechtigungsrichtlinie, die Sie der Rolle hinzufügen können. Als Nächstes erstellen Sie die Rolle und fügen Sie die Richtlinie an. Nachdem Sie eine Rolle erstellt haben, können Sie die Berechtigungen der Rolle ändern, indem Sie Berechtigungsrichtlinien anhängen und trennen.

> **Note**
>
> Wenn Sie Rollen für dieAWS Data Pipeline Verwendung der Konsole wie unten beschrieben erstellen, erstellt IAM die entsprechenden Vertrauensrichtlinien, die für die Rolle erforderlich sind, und fügt sie hinzu.

Um eine Berechtigungsrichtlinie zur Verwendung mit einer Rolle für zu erstellenAWS Data Pipeline

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen).
3. Wählen Sie den Tab JSON.
4. Wenn Sie eine Pipeline-Rolle erstellen, kopieren Sie den Inhalt des RichtlinienbeispielsBeispiel für eine Berechtigungsrichtlinie für -Rolle, fügen Sie ihn ein und bearbeiten Sie es entsprechend

Ihren Sicherheitsanforderungen. Wenn Sie alternativ eine benutzerdefinierte EC2-Instanzrolle erstellen, tun Sie dasselbe für das Beispiel in Verwaltete Standardrichtlinie für die EC2-Instanzrolle.

5. Wählen Sie Review policy (Richtlinie prüfen).
6. Geben Sie einen Namen für die Richtlinie ein, z. B.,`MyDataPipelineRolePolicy` und eine optionale Beschreibung, und wählen Sie dann Richtlinie erstellen aus.
7. Beachten Sie den Namen der Richtlinie. Sie benötigen ihn beim Erstellen Ihrer Rolle.

So erstellen Sie eine IAM-Rolle für AWS Data Pipeline

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen) aus.
3. Wählen Sie unter Wählen Sie einen Anwendungsfall die Option Data Pipeline aus.
4. Führen Sie unter Select your use case (Wählen Sie Ihren Anwendungsfall aus) einen der folgenden Schritte aus:
   - Wählen Sie`Data Pipeline`, ob Sie eine Pipeline-Rolle erstellen möchten.
   - Wählen Sie`EC2 Role for Data Pipeline`, ob Sie eine Ressourcenrolle erstellen möchten.
5. Wählen Sie Next: Permissions (Weiter: Berechtigungen) aus.
6. Wenn die Standardrichtlinie für aufgeführtAWS Data Pipeline ist, fahren Sie mit den folgenden Schritten fort, um die Rolle zu erstellen, und bearbeiten Sie sie dann gemäß den Anweisungen im nächsten Verfahren. Andernfalls geben Sie den Namen der Richtlinie ein, die Sie im obigen Verfahren erstellt haben, und wählen Sie sie dann aus der Liste aus.
7. Wählen Sie Weiter: Schlagworte, geben Sie alle Tags ein, die Sie der Rolle hinzufügen möchten, und wählen Sie dann Weiter: Überprüfen.
8. Geben Sie beispielsweise einen Namen für die Rolle`MyDataPipelineRole` und eine optionale Beschreibung ein, und wählen Sie dann Rolle erstellen.

Um eine Berechtigungsrichtlinie für eine IAM-Rolle anzuhängen oder zu trennen fürAWS Data Pipeline

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

2. Wählen Sie im Navigationsbereich Roles aus
3. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie bearbeiten möchten, z. B. DataPipelineDefaultRoleoder, und wählen MyDataPipelineRoleSie dann den Rollennamen aus der Liste aus.
4. Gehen Sie auf der Registerkarte Berechtigungen wie folgt vor:

   - Um eine Berechtigungsrichtlinie zu trennen, klicken Sie unter Berechtigungsrichtlinien auf die Schaltfläche Entfernen ganz rechts neben dem Richtlinieneintrag. Wählen Sie Trennen, wenn Sie zur Bestätigung aufgefordert werden.
   - Um eine Richtlinie anzuhängen, die Sie zuvor erstellt haben, wählen Sie Richtlinien anhängen. Geben Sie im Suchfeld den Namen der Richtlinie ein, die Sie ändern möchten, wählen Sie sie aus der Liste aus und wählen Sie dann Attach policy aus.

Rollen für eine bestehende Pipeline ändern

Wenn Sie einer Pipeline eine andere Pipeline-Rolle oder Ressourcenrolle zuweisen möchten, können Sie den Architect-Editor in derAWS Data Pipeline Konsole verwenden.

So bearbeiten Sie die einer Pipeline zugewiesenen Rollen mithilfe der Konsole

1. Öffnen Sie dieAWS Data Pipeline Konsole unter https://console.aws.amazon.com/datapipeline/.
2. Wählen Sie die Pipeline aus der Liste aus, und wählen Sie dann Aktionen, Bearbeiten.
3. Wählen Sie im rechten Bereich des Architekten-Editors die Option Andere aus.
4. Wählen Sie in den Listen Ressourcenrolle und Rolle die Rollen aus,AWS Data Pipeline denen Sie zuweisen möchten, und klicken Sie dann auf Speichern.

Protokollierung und Überwachung in AWS Data Pipeline

AWS Data Pipelineist in integriertAWS CloudTrail, einem Service, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einemAWS -Service durchgeführten Aktionen bereitstelltAWS Data Pipeline. CloudTrail erfasst alle API-Aufrufe fürAWS Data Pipeline als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS Data Pipeline-Konsole und Code-Aufrufe der AWS Data Pipeline-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket, einschließlich Ereignisse für aktivierenAWS Data Pipeline. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail -Konsole trotzdem in Event history (Ereignisverlauf) anzeigen. Mit den

von CloudTrail gesammelten Informationen können Sie die an gestellte AnfrageAWS Data Pipeline, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und weitere Angaben bestimmen.

Weitere Informationen CloudTrail finden Sie im AWS CloudTrailBenutzerhandbuch.

AWS Data PipelineInformationen in CloudTrail

CloudTrail wird beim Erstellen IhresAWS -Kontos für Sie aktiviert. Die inAWS Data Pipeline auftretenden Aktivitäten werden als CloudTrail Ereignis zusammen mit anderenAWS -Serviceereignissen in Event history (Ereignisverlauf) aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail -Ereignisverlauf.

Zur kontinuierlichen Aufzeichnung von Ereignissen in Ihrem AWS-Konto, einschließlich Ereignissen für AWS Data Pipeline, erstellen Sie einen Trail. Ein Trail ermöglicht es CloudTrail Ihnen, Protokolldateien in einem Amazon S3 S3-Bucket bereitzustellen. Wenn Sie einen Pfad in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon S3 Bucket bereit. Darüber hinaus können Sie andereAWS -Services konfigurieren, um die in den CloudTrail -Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfigurieren von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

AlleAWS Data Pipeline Aktionen werden im Kapitel AWS Data Pipeline API Reference Actions protokolliert CloudTrail und sind dort dokumentiert. Aufrufe der CreatePipelineAktion generieren beispielsweise Einträge in den CloudTrail -Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder IAM-Rollenanmeldeinformationen ausgeführt wurde.

- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer gesendet wurde.
- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

Grundlagen von AWS Data Pipeline-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail Protokolleinträge sind kein geordnetes Stack-Trace der öffentlichen API-Aufrufe und erscheinen daher in keiner bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail -Protokolleintrag, der die`CreatePipeline` -Operation demonstriert:

```
    {
"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
  ]
}
```

Vorfallreaktion in AWS Data Pipeline

Die Reaktion auf Vorfälle für die AWS Data Pipeline liegt in der Verantwortung von AWS. AWS verfügt über eine formelle, dokumentierte Richtlinie und ein Programm, die/das die Reaktion auf Vorfälle regelt.

Operative AWS-Probleme mit weitreichenden Auswirkungen werden im AWS Service Health Dashboard veröffentlicht. Operative Probleme werden ebenfalls über das Personal Health Dashboard in den einzelnen Konten gepostet.

Compliance-Validierung für AWS Data Pipeline

AWS Data Pipeline ist nicht im Umfang der AWS-Compliance-Programme enthalten. Eine Liste der AWS-Services, die in bestimmten Compliance-Programmen enthalten sind, finden Sie unter AWS Services in Scope nach Compliance-Programm. Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Ausfallsicherheit in AWS Data Pipeline

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und Availability Zones. AWS -Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Sicherheit der Infrastruktur in AWS Data Pipeline

Als Managed ServiceAWS Data Pipelineist geschützt durch dieAWSglobale Verfahren zur Netzwerksicherheit, die imAmazon Web Services: Übersicht über SicherheitsprozesseWhitepaper.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf AWS Data Pipeline zuzugreifen. Kunden müssen Transport Layer Security (TLS) 1.0 oder neuer unterstützen. Wir empfehlen TLS 1.2 oder höher. Clients müssen außerdem Cipher Suites mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Konfiguration und Schwachstellenanalyse in AWS Data Pipeline

Konfiguration und IT-Steuerelemente unterliegen der übergreifenden Verantwortlichkeit von AWS und Ihnen, unserem Kunden. Weitere Informationen finden Sie unter AWSModell der übergreifenden Verantwortlichkeit.

Tutorials

Die folgenden Tutorials führen Sie step-by-step durch den Prozess der Erstellung und Verwendung von Pipelines mitAWS Data Pipeline.

Tutorials

- Verarbeiten Sie Daten mithilfe von Amazon EMR mit Hadoop Streaming
- Kopieren Sie CSV-Daten zwischen Amazon S3-Buckets mithilfe von AWS Data Pipeline
- Exportieren Sie MySQL-Daten nach Amazon S3 mit AWS Data Pipeline
- Kopieren Sie Daten nach Amazon Redshift mit AWS Data Pipeline

Verarbeiten Sie Daten mithilfe von Amazon EMR mit Hadoop Streaming

Sie können es verwendenAWS Data Pipeline, um Ihre Amazon EMR-Cluster zu verwalten. Mit können AWS Data Pipeline Sie Vorbedingungen angeben, die erfüllt sein müssen, bevor der Cluster gestartet wird (z. B. sicherstellen, dass die heutigen Daten auf Amazon S3 hochgeladen wurden), einen Zeitplan für die wiederholte Ausführung des Clusters und die zu verwendende Clusterkonfiguration angeben. Das folgende Tutorial führt Sie durch den Start eines einfachen Clusters.

In diesem Tutorial erstellen Sie eine Pipeline für einen einfachen Amazon EMR-Cluster, um einen bereits vorhandenen Hadoop Streaming-Job von Amazon EMR auszuführen und eine Amazon SNS-Benachrichtigung zu senden, nachdem die Aufgabe erfolgreich abgeschlossen wurde. Sie verwenden die Amazon EMR-Cluster-Ressource, die von AWS Data Pipeline für diese Aufgabe bereitgestellt wird. Die Beispielanwendung wird aufgerufen WordCount und kann auch manuell von der Amazon EMR-Konsole aus ausgeführt werden. Beachten Sie, dass Cluster, die in AWS Data Pipeline Ihrem Namen erstellt wurden, in der Amazon EMR-Konsole angezeigt werden und Ihrem AWS-Konto in Rechnung gestellt werden.

Pipeline-Objekte

Die Pipeline verwendet die folgenden Objekte:

EmrActivity

Definiert die Arbeit, die in der Pipeline ausgeführt werden soll (einen bereits vorhandenen Hadoop Streaming-Job ausführen, der von Amazon EMR bereitgestellt wird).

EmrCluster

Die Ressource, die AWS Data Pipelineum Ausführen dieser Aktivität verwendet.

Ein Cluster ist ein Satz von Amazon EC2-Instances. AWS Data Pipelinestartet den Cluster und beendet ihn dann, nachdem die Aufgabe abgeschlossen ist.

Plan

Startdatum, Uhrzeit und Dauer dieser Aktivität. Sie können optional das Enddatum und die Endzeit angeben.

SnsAlarm

Sendet eine Amazon SNS-Benachrichtigung zu dem von Ihnen angegebenen Thema, nachdem die Aufgabe erfolgreich abgeschlossen wurde.

Inhalt

Bevor Sie beginnen

Stellen Sie sicher, dass Sie die folgenden Schritte ausgeführt haben.

- Führen Sie die Aufgaben unter au Einrichten für AWS Data Pipeline.
- (Optional) Richten Sie eine VPC für den Cluster und eine Sicherheitsgruppe für die VPC ein.
- Erstellen Sie ein Thema zum Senden einer E-Mail-Benachrichtigung und notieren Sie sich das Thema Amazon-Ressourcenname (ARN). Weitere Informationen dazu erhalten Sie unter Erstellen eines Themas im Amazon Simple Notification Service Handbuch Erste Schritte.

Einen Cluster über die Befehlszeile starten

Wenn Sie regelmäßig einen Amazon EMR-Cluster zur Analyse von Weblogs oder zur Analyse wissenschaftlicher Daten ausführen, können Sie ihn AWS Data Pipeline zur Verwaltung Ihrer Amazon

EMR-Cluster verwenden. Mit können Sie Vorbedingungen angebenAWS Data Pipeline, die erfüllt sein müssen, bevor der Cluster gestartet wird (z. B. sicherstellen, dass die heutigen Daten auf Amazon S3 hochgeladen wurden). Dieses Tutorial führt Sie durch den Start eines Clusters, der als Modell für eine einfache Amazon EMR-basierte Pipeline oder als Teil einer komplexeren Pipeline dienen kann.

Voraussetzungen

Bevor Sie die Befehlszeile zum ersten Mal verwenden können, müssen Sie die folgenden Schritte ausführen:

1. Installieren und konfigurieren Sie eine Befehlszeilenschnittstelle (CLI). Weitere Informationen finden Sie unter Zugriff auf AWS Data Pipeline.
2. Stellen Sie sicher, dass die IAM-Rollen benannt DataPipelineDefaultResourceRolesind DataPipelineDefaultRoleund existieren. Die AWS Data Pipeline Konsole erstellt diese Rollen automatisch für Sie. Wenn Sie die AWS Data Pipeline Konsole nicht mindestens einmal verwendet haben, müssen Sie diese Rollen manuell erstellen. Weitere Informationen finden Sie unter IAM-Rollen für AWS Data Pipeline.

Aufgaben

- Erstellen der Pipeline-Definitionsdatei
- Hochladen und Aktivieren der Pipeline-Definition
- Überwachen der Pipeline-Runs

Erstellen der Pipeline-Definitionsdatei

Der folgende Code ist die Pipeline-Definitionsdatei für einen einfachen Amazon EMR-Cluster, der einen vorhandenen Hadoop-Streaming-Job ausführt, der von Amazon EMR bereitgestellt wird. Diese Beispielanwendung heißtWordCount, und Sie können sie auch mit der Amazon EMR-Konsole ausführen.

Kopieren Sie diesen Code in eine Textdatei, und speichern Sie sie unter `MyEmrPipelineDefinition.json`. Sie sollten den Amazon S3-Bucket-Speicherort durch den Namen eines Amazon S3-Buckets ersetzen, den Sie besitzen. Sie sollten auch das Start- und das Enddatum ersetzen. Um Cluster sofort zu starten, legen Sie `startDateTime` auf einen Tag früher in der Vergangenheit und `endDateTime` auf einen Tag später in der Zukunft fest. AWS Data Pipeline startet dann die „überfälligen“ Cluster sofort, um den als Problem wahrgenommenen Rückstand an

Aufgaben aufzuholen. So müssen Sie nicht eine Stunde warten, bis AWS Data Pipeline den ersten Cluster startet.

```
{
  "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"
    }
  ]
}
```

Diese Pipeline hat drei Objekte:

- `Hourly`, was den Zeitplan für die Arbeit repräsentiert. Sie können einen Zeitplan als eines der Felder für eine Aktivität festlegen. Wenn Sie das tun, wird die Aktivität gemäß diesem Zeitplan ausgeführt, in diesem Fall stündlich.

- `MyCluster`, das die Gruppe von Amazon EC2-Instances darstellt, die für die Ausführung des Clusters verwendet werden. Sie können die Größe und die Anzahl an EC2-Instances festlegen, die als Cluster ausgeführt werden sollen. Wenn Sie die Anzahl an Instances nicht festlegen, startet der Cluster mit zwei Instances, einem Master-Knoten und einem Aufgabenknoten. Sie können ein Subnetz angeben, in dem der Cluster gestartet werden soll. Sie können dem Cluster zusätzliche Konfigurationen hinzufügen, z. B. Bootstrap-Aktionen, um zusätzliche Software auf das von Amazon EMR bereitgestellte AMI zu laden.
- `MyEmrActivity`, was die Berechnung darstellt, die mit dem Cluster verarbeitet werden soll. Amazon EMR unterstützt verschiedene Arten von Clustern, darunter Streaming, Cascading und Scripted Hive. Das `runsOn` Feld bezieht sich auf`MyCluster`, wobei es als Spezifikation für die Grundlagen des Clusters verwendet wird.

Hochladen und Aktivieren der Pipeline-Definition

Sie müssen Ihre Pipeline-Definition hochladen und Ihre Pipeline aktivieren. Ersetzen Sie in den folgenden Beispielbefehlen *`pipeline_name`* durch ein Label für Ihre Pipeline und *`pipeline_file durch den vollständig qualifizierten Pfad für die Pipeline-Definitionsdatei`*`.json`

AWS CLI

Verwenden Sie den folgenden Befehl create-pipeline, um Ihre Pipeline-Definition zu erstellen und Ihre Pipeline zu aktivieren. Notieren Sie sich die ID Ihrer Pipeline, da Sie diesen Wert bei den meisten CLI-Befehlen verwenden werden.

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

Verwenden Sie den folgenden put-pipeline-definitionBefehl, um Ihre Pipeline-Definition hochzuladen.

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

Wenn Ihre Pipeline erfolgreich validiert wurde, ist das `validationErrors` Feld leer. Sie sollten alle Warnungen überprüfen.

Verwenden Sie den folgenden Befehl activate-pipeline, um Ihre Pipeline zu aktivieren.

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

Mit dem folgenden Befehl list-pipelines können Sie überprüfen, ob Ihre Pipeline in der Pipelineliste erscheint.

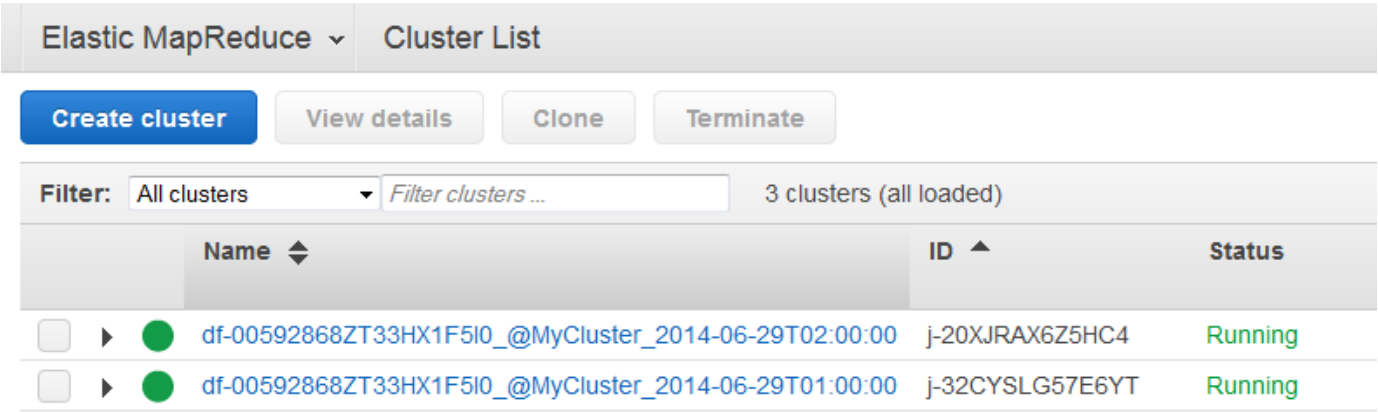
```
aws datapipeline list-pipelines
```

Überwachen der Pipeline-Runs

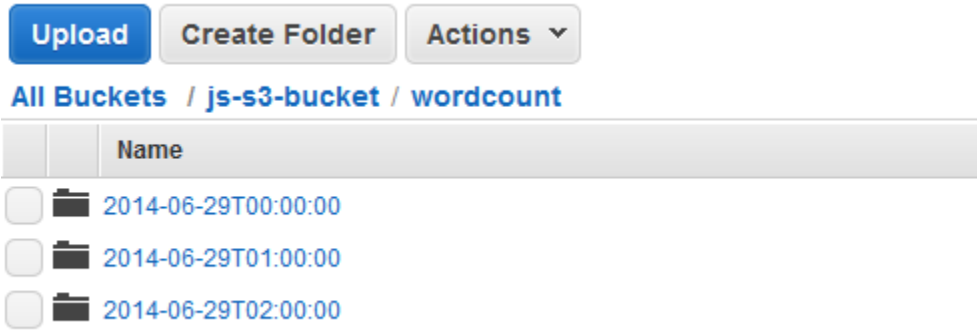
Sie können Cluster anzeigen, die AWS Data Pipeline mithilfe der Amazon EMR-Konsole gestartet wurden, und Sie können den Ausgabeordner mithilfe der Amazon S3-Konsole anzeigen.

So überprüfen Sie den Fortschritt der von AWS Data Pipeline gestarteten Cluster

1. Öffnen Sie die Amazon EMR-Konsole.
2. <launch-time>Die Cluster, die von erzeugt wurden, AWS Data Pipeline haben einen Namen, der wie folgt formatiert ist: <pipeline-identifier>_@ *< > emr-cluster-name* _.



3. Nachdem einer der Läufe abgeschlossen ist, öffnen Sie die Amazon S3-Konsole und überprüfen Sie, ob der Ausgabeordner mit Zeitstempel vorhanden ist und die erwarteten Ergebnisse des Clusters enthält.



Kopieren Sie CSV-Daten zwischen Amazon S3-Buckets mithilfe von AWS Data Pipeline

Nachdem Sie Was ist AWS Data Pipeline? gelesen und entschieden haben, dass Sie AWS Data Pipeline verwenden möchten, um die Bewegung und Transformation Ihrer Daten zu automatisieren, ist es an der Zeit, mit der Erstellung von Datenpipelines zu beginnen. Um Ihnen die Funktionsweise von AWS Data Pipeline näher zu bringen, hier ein Beispiel für eine einfache Aufgabe.

Dieses Tutorial führt Sie durch den Prozess der Erstellung einer Datenpipeline, um Daten von einem Amazon S3-Bucket in einen anderen zu kopieren und dann eine Amazon SNS-Benachrichtigung zu senden, nachdem die Kopieraktivität erfolgreich abgeschlossen wurde. Sie verwenden eine EC2-Instance, die von AWS Data Pipeline für diese Kopieraktivität verwaltet wird.

Pipeline-Objekte

Die Pipeline verwendet die folgenden Objekte:

CopyActivity

Die Aktivität, die für AWS Data Pipeline diese Pipeline ausgeführt wird (Kopieren von CSV-Daten von einem Amazon S3-Bucket in einen anderen).

> **Important**
>
> Es gibt Einschränkungen bei der Verwendung des CSV-Dateiformats mit `CopyActivity` und `S3DataNode`. Weitere Informationen finden Sie unter CopyActivity.

Plan

Das Startdatum, die Uhrzeit und die Wiederholung für diese Aktivität. Sie können optional das Enddatum und die Endzeit angeben.

Ec2Resource

Die Ressource (eine EC2-Instance), die AWS Data Pipeline verwendet, um diese Aktivität auszuführen.

S3 DataNode

Die Eingabe- und Ausgabeknoten (Amazon S3-Buckets) für diese Pipeline.

SnsAlarm

Die Aktion AWS Data Pipeline muss ausgeführt werden, wenn die angegebenen Bedingungen erfüllt sind (Amazon SNS-Benachrichtigungen zu einem Thema senden, nachdem die Aufgabe erfolgreich abgeschlossen wurde).

Inhalt

Bevor Sie beginnen

Stellen Sie sicher, dass Sie die folgenden Schritte ausgeführt haben.

- Führen Sie die Aufgaben unter au Einrichten für AWS Data Pipeline.
- (Optional) Richten Sie eine VPC für die Instance und eine Sicherheitsgruppe für die VPC ein.
- Erstellen Sie einen Amazon S3-Bucket als Datenquelle.

  Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service.
- Laden Sie Ihre Daten in Ihren Amazon S3-Bucket hoch.

  Anleitungen finden Sie unter Hinzufügen eines Objekts zu einem Bucket im Benutzerhandbuch zu Amazon Simple Storage Service.
- Einen weiteren Amazon S3-Bucket als Datenziel erstellen

- Erstellen Sie ein Thema zum Senden einer E-Mail-Benachrichtigung und notieren Sie sich das Thema Amazon-Ressourcenname (ARN). Weitere Informationen dazu erhalten Sie unter Erstellen eines Themas im Amazon Simple Notification Service Handbuch Erste Schritte.
- (Optional) Dieses Tutorial verwendet die Standard-IAM-Rollenrichtlinien, die von AWS Data Pipeline erstellt wurden. Wenn Sie lieber Ihre eigenen IAM-Rollenrichtlinien und Vertrauensbeziehungen erstellen und konfigurieren möchten, folgen Sie den Anweisungen unterIAM-Rollen für AWS Data Pipeline.

CSV-Daten mithilfe der Befehlszeile kopieren

Sie können Pipelines erstellen und verwenden, um Daten von einem Amazon S3-Bucket in einen anderen zu kopieren.

Voraussetzungen

Bevor Sie beginnen, müssen Sie die folgenden Schritte ausführen:

1. Installieren und konfigurieren Sie eine Befehlszeilenschnittstelle (CLI). Weitere Informationen finden Sie unter Zugriff auf AWS Data Pipeline.
2. Stellen Sie sicher, dass die IAM-Rollen benannt DataPipelineDefaultResourceRolesind DataPipelineDefaultRoleund existieren. Die AWS Data Pipeline Konsole erstellt diese Rollen automatisch für Sie. Wenn Sie die AWS Data Pipeline Konsole nicht mindestens einmal verwendet haben, müssen Sie diese Rollen manuell erstellen. Weitere Informationen finden Sie unter IAM-Rollen für AWS Data Pipeline.

Aufgaben

- Definieren Sie eine Pipeline im JSON-Format
- Hochladen und Aktivieren der Pipeline-Definition

Definieren Sie eine Pipeline im JSON-Format

Dieses Beispielszenario zeigt, wie JSON-Pipeline-Definitionen und die AWS Data Pipeline CLI verwendet werden, um das Kopieren von Daten zwischen zwei Amazon S3-Buckets in einem bestimmten Zeitintervall zu planen. Dies ist die vollständige Pipeline-Definition-JSON-Datei, gefolgt von einer Erläuterung für jeden ihrer Abschnitte.

> **Note**
>
> Wir empfehlen, dass Sie einen Texteditor verwenden, mit dem Sie die Syntax von JSON-formatierten Dateien überprüfen und die Datei mit der Dateierweiterung .json benennen können.

In diesem Beispiel überspringen wir aus Gründen der Übersichtlichkeit die optionalen Felder und zeigen nur erforderliche Felder an. Die vollständige Pipeline-JSON-Datei für dieses Beispiel lautet:

```
{
  "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"
      }
    }
  ]
}
```

Plan

Die Pipeline definiert einen Zeitplan mit einem Start- und Enddatum sowie einem Zeitraum, um zu bestimmen, wie häufig die Aktivität in dieser Pipeline ausgeführt wird.

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

Amazon S3-Datenknoten

Als Nächstes definiert die DataNode S3-Eingabe-Pipeline-Komponente einen Speicherort für die Eingabedateien, in diesem Fall einen Amazon S3-Bucket-Speicherort. Die DataNode S3-Eingabekomponente wird durch die folgenden Felder definiert:

```
{
```

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

Id

Der benutzerdefinierte Name für den Eingabestandort (eine Bezeichnung nur für Ihre Referenz).

Typ

Der Pipeline-Komponententyp, der „S3DataNode“ ist, um dem Ort zu entsprechen, an dem sich die Daten befinden, in einem Amazon S3-Bucket.

Plan

Ein Verweis auf die Zeitplankomponente, die wir in den vorherigen Zeilen der JSON-Datei mit der Bezeichnung „MySchedule“ erstellt haben.

Pfad

Der Pfad zu den Daten, die dem Datenknoten zugeordnet sind. Die Syntax für einen Datenknoten wird von seinem Typ bestimmt. Die Syntax für einen Amazon S3-Pfad folgt beispielsweise einer anderen Syntax, die für eine Datenbanktabelle geeignet ist.

Als Nächstes definiert die DataNode S3-Ausgabekomponente den Ausgabezielort für die Daten. Sie folgt demselben Format wie die DataNode S3-Eingabekomponente, mit Ausnahme des Namens der Komponente und eines anderen Pfads zur Angabe der Zieldatei.

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

Ressource

Dies ist eine Definition der Rechenressource, die die Kopieroperation ausführt. In diesem Beispiel sollte AWS Data Pipeline automatisch eine EC2-Instance erstellen, um die Kopieraufgabe auszuführen und die Ressource zu beenden, nachdem die Aufgabe abgeschlossen wurde. Die hier definierten Felder steuern die Erstellung und Funktion der EC2 Instance, die die Arbeit erledigt. Die EC2Resource ist durch folgende Felder definiert:

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

Id

Der benutzerdefinierte Name für den Pipeline-Zeitplan, der nur für Ihre Referenz gilt.

Typ

Die Art der Rechenressource zur Ausführung der Arbeit; in diesem Fall eine EC2 Instance. Es sind andere Ressourcentypen verfügbar, z. B. ein EmrCluster Typ.

Plan

Der Zeitplan für die Erstellung dieser Rechenressource.

instanceType

Die Größe der zu erstellenden EC2 Instance. Stellen Sie sicher, dass Sie die geeignete Größe der EC2-Instance festgelegt haben, die der Last der Arbeit am besten entspricht, die Sie mit AWS Data Pipeline durchführen möchten. In diesem Fall setzen wir eine m1.medium EC2 Instance. Weitere Informationen zu den verschiedenen Instance-Typen und zur Verwendung der einzelnen Instance-Typen finden Sie im Thema Amazon EC2-Instance-Typen unter http://aws.amazon.com/ec2/instance-types/.

Rolle

Die IAM-Rolle des Kontos, das auf Ressourcen zugreift, z. B. auf einen Amazon S3-Bucket zugreift, um Daten abzurufen.

resourceRole

Die IAM-Rolle des Kontos, das Ressourcen erstellt, z. B. für Sie eine EC2 Instance erstellt und konfiguriert. Rolle und ResourceRole können dieselbe Rolle sein, bieten jedoch getrennt voneinander eine größere Granularität in Ihrer Sicherheitskonfiguration.

Aktivität

Der letzte Abschnitt in der JSON-Datei ist die Definition der Aktivität, die die auszuführende Arbeit darstellt. In diesem Beispiel werden `CopyActivity` Daten aus einer CSV-Datei in einem http://aws.amazon.com/ec2/instance-types/ -Bucket in einen anderen kopiert. Die `CopyActivity`-Komponente ist durch folgende Felder definiert:

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

Id

Der benutzerdefinierte Name für die Aktivität, der nur für Ihre Referenz eine Bezeichnung ist.

Typ

Die Art der auszuführenden Aktivität, z. MyCopyActivity B.

runsOn

Die Datenverarbeitungsressource, die die Arbeit ausführt, die diese Aktivität definiert. In diesem Beispiel stellen wir einen Verweis auf die zuvor definierte EC2 Instance bereit. Durch die Verwendung des Felds `runsOn` wird AWS Data Pipeline veranlasst, die EC2-Instance für Sie zu erstellen. Das Feld `runsOn` zeigt an, dass die Ressource in der AWS-Infrastruktur vorhanden ist, während der Wert `workerGroup` angibt, dass Sie Ihre eigenen lokalen Ressourcen zur Ausführung der Arbeit verwenden möchten.

Eingabe

Der Speicherort der zu kopierenden Daten.

Ausgabe

Die Zielortdaten.

Plan

Der Zeitplan für die Ausführung dieser Aktivität.

Hochladen und Aktivieren der Pipeline-Definition

Sie müssen Ihre Pipeline-Definition hochladen und Ihre Pipeline aktivieren. Ersetzen Sie in den folgenden Beispielbefehlen *`pipeline_name`* durch ein Label für Ihre Pipeline und *`pipeline_file durch den vollständig qualifizierten Pfad für die Pipeline-Definitionsdatei`*`.json`

AWS CLI

Verwenden Sie den folgenden Befehl create-pipeline, um Ihre Pipeline-Definition zu erstellen und Ihre Pipeline zu aktivieren. Notieren Sie sich die ID Ihrer Pipeline, da Sie diesen Wert bei den meisten CLI-Befehlen verwenden werden.

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

Verwenden Sie den folgenden put-pipeline-definitionBefehl, um Ihre Pipeline-Definition hochzuladen.

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

Wenn Ihre Pipeline erfolgreich validiert wurde, ist das `validationErrors` Feld leer. Sie sollten alle Warnungen überprüfen.

Verwenden Sie den folgenden Befehl activate-pipeline, um Ihre Pipeline zu aktivieren.

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

Mit dem folgenden Befehl list-pipelines können Sie überprüfen, ob Ihre Pipeline in der Pipelineliste erscheint.

```
aws datapipeline list-pipelines
```

Exportieren Sie MySQL-Daten nach Amazon S3 mit AWS Data Pipeline

Dieses Tutorial führt Sie durch den Prozess der Erstellung einer Datenpipeline, um Daten (Zeilen) aus einer Tabelle in der MySQL-Datenbank in eine CSV-Datei (durch Kommas getrennte Werte) in einem Amazon S3-Bucket zu kopieren und dann eine Amazon SNS-Benachrichtigung zu senden, nachdem die Kopieraktivität erfolgreich abgeschlossen wurde. Für diese Kopieraktivität verwenden Sie eine von AWS Data Pipeline bereitgestellte EC2-Instance.

Pipeline-Objekte

Die Pipeline verwendet die folgenden Objekte:

- CopyActivity
- Ec2Resource
- MySqlDataNode
- S3 DataNode
- SnsAlarm

Inhalt

Bevor Sie beginnen

Stellen Sie sicher, dass Sie die folgenden Schritte ausgeführt haben.

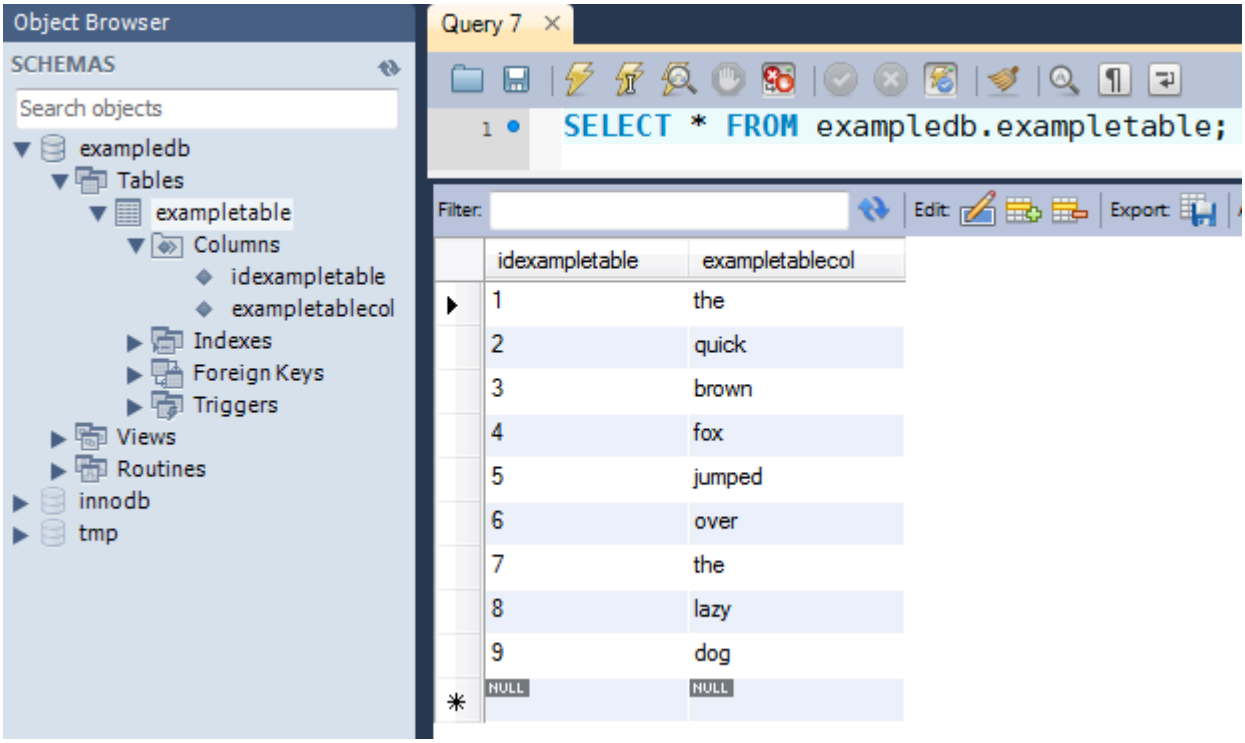
- Führen Sie die Aufgaben unter au Einrichten für AWS Data Pipeline.
- (Optional) Richten Sie eine VPC für die Instance und eine Sicherheitsgruppe für die VPC ein.
- Erstellen Sie einen Amazon S3-Bucket als Datenausgabe.

  Weitere Informationen finden Sie unter Erstellen eines Buckets im Amazon Simple Storage Service-Benutzerhandbuch.
- Erstellen und starten Sie eine MySQL-Datenbank-Instance als Datenquelle.

  Weitere Informationen finden Sie unter Starten einer DB-Instance im Amazon RDS-Handbuch „Erste Schritte“. Nachdem Sie eine Amazon RDS-Instance eingerichtet haben, finden Sie in der MySQL-Dokumentation weitere Informationen unter Erstellen einer Tabelle.

  > **Note**
  >
  > Notieren Sie sich den Benutzernamen und das Passwort, das Sie beim Erstellen der MySQL-Instance verwendet haben. Nachdem Sie Ihre MySQL-Datenbank-Instance gestartet haben, notieren Sie sich den Endpunkt der Instance. Sie benötigen diese Informationen später wieder.
- Stellen Sie eine Verbindung mit der MySQL-Datenbank-Instance her, erstellen Sie eine Tabelle und fügen Sie die Testdatenwerte zur neu erstellten Tabelle hinzu.

  Zur Verdeutlichung und Unterstützung haben wir dieses Tutorial erstellt, mit einer MySQL-Tabelle mit der folgenden Konfiguration und Beispieldaten. Der folgende Screenshot stammt von MySQL Workbench 5.2 CE:



Weitere Informationen finden Sie unter Erstellen einer Tabelle in der MySQL-Dokumentation und auf der MySQL Workbench-Produktseite.

- Erstellen Sie ein Thema zum Senden einer E-Mail-Benachrichtigung und notieren Sie sich das Thema Amazon-Ressourcenname (ARN). Weitere Informationen finden Sie unter Thema erstellen im Amazon Simple Notification Service Getting Started Guide.
- (Optional) Dieses Tutorial verwendet die Standard-IAM-Rollenrichtlinien, die von AWS Data Pipeline erstellt wurden. Wenn Sie lieber Ihre IAM-Rollenrichtlinie und Vertrauensbeziehungen erstellen und konfigurieren möchten, folgen Sie den Anweisungen unterIAM-Rollen für AWS Data Pipeline.

MySQL-Daten über die Befehlszeile kopieren

Sie können eine Pipeline erstellen, um Daten aus einer MySQL-Tabelle in eine Datei in einem Amazon S3-Bucket zu kopieren.

Voraussetzungen

Bevor Sie beginnen, müssen Sie die folgenden Schritte ausführen:

1. Installieren und konfigurieren Sie eine Befehlszeilenschnittstelle (CLI). Weitere Informationen finden Sie unter Zugriff auf AWS Data Pipeline.

2. Stellen Sie sicher, dass die IAM-Rollen benannt DataPipelineDefaultResourceRolesind DataPipelineDefaultRoleund existieren. Die AWS Data Pipeline Konsole erstellt diese Rollen automatisch für Sie. Wenn Sie die AWS Data Pipeline Konsole nicht mindestens einmal verwendet haben, müssen Sie diese Rollen manuell erstellen. Weitere Informationen finden Sie unter IAM-Rollen für AWS Data Pipeline.
3. Richten Sie einen Amazon S3-Bucket und eine Amazon RDS-Instance ein. Weitere Informationen finden Sie unter Bevor Sie beginnen.

Aufgaben

- Definieren Sie eine Pipeline im JSON-Format
- Hochladen und Aktivieren der Pipeline-Definition

Definieren Sie eine Pipeline im JSON-Format

Dieses Beispielszenario zeigt, wie JSON-Pipeline-Definitionen und die AWS Data Pipeline CLI verwendet werden, um Daten (Zeilen) aus einer Tabelle in einer MySQL-Datenbank in eine CSV-Datei (durch Kommas getrennte Werte) in einem Amazon S3-Bucket in einem bestimmten Zeitintervall zu kopieren.

Dies ist die vollständige Pipeline-Definition-JSON-Datei, gefolgt von einer Erläuterung für jeden ihrer Abschnitte.

> **Note**
>
> Wir empfehlen, dass Sie einen Texteditor verwenden, mit dem Sie die Syntax von JSON-formatierten Dateien überprüfen und die Datei mit der Dateierweiterung .json benennen können.

```
{
  "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"
    }
  ]
}
```

MySQL-Datenknoten

Die MySqlDataNode Eingabe-Pipeline-Komponente definiert einen Speicherort für die Eingabedaten, in diesem Fall eine Amazon RDS-Instance. Die MySqlDataNode Eingabekomponente wird durch die folgenden Felder definiert:

```
{
  "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

Den benutzerdefinierten Namen, der nur als Referenz dient.

Benutzername

Der Benutzername des Datenbank-Kontos, das über ausreichend Berechtigungen verfügt, um Daten aus der Datenbanktabelle abzurufen. Ersetzen Sie *my-username* durch den Namen Ihres Benutzers.

Plan

Ein Verweis auf die Zeitplankomponente, die wir in den vorhergehenden Zeilen der JSON-Datei erstellt haben.

Name

Den benutzerdefinierten Namen, der nur als Referenz dient.

*Passwort

Das Passwort für das Datenbankkonto mit dem Sternchen-Präfix, das darauf hinweist, dass AWS Data Pipeline den Passwortwert verschlüsseln muss. Ersetzen Sie *my-password* durch

das richtige Passwort für Ihren Benutzer. Dem Passwortfeld ist das Sternchen-Sonderzeichen vorangestellt. Weitere Informationen finden Sie unter Sonderzeichen.

Tabelle

Der Name der Datenbanktabelle, die die zu kopierenden Daten enthält. Ersetzen Sie *table-name* durch den Namen Ihrer Datenbanktabelle.

connectionString

Die JDBC-Verbindungszeichenfolge für das CopyActivity Objekt, das eine Verbindung zur Datenbank herstellen soll.

selectQuery

Eine gültige SQL-SELECT-Abfrage, die festlegt, welche Daten aus der Datenbanktabelle kopiert werden sollen. Beachten Sie, dass `#{table}` ein Ausdruck ist, der den Tabellennamen wiederverwendet, der in den vorhergehenden Zeilen der JSON-Datei durch die Variable "table" angegeben wird.

Typ

Der SqlDataNode Typ, bei dem es sich in diesem Beispiel um eine Amazon RDS-Instance handelt, die MySQL verwendet.

> **Note**
>
> Der MySqlDataNode-Typ ist veraltet. Sie können es zwar weiterhin verwendenMySqlDataNode, wir empfehlen jedoch, es zu verwendenSqlDataNode.

Amazon S3-Datenknoten

Als Nächstes definiert die S3Output-Pipeline-Komponente einen Speicherort für die Ausgabedatei, in diesem Fall eine CSV-Datei in einem Amazon S3-Bucket-Speicherort. Die DataNode S3-Ausgabekomponente wird durch die folgenden Felder definiert:

```
{
  "id": "S3DataNodeId114",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "filePath": "s3://example-bucket/rds-output/output.csv",
```

```
  "name": "My S3 Data",
  "type": "S3DataNode"
},
```

Id

Die benutzerdefinierte ID, die nur als Referenz dient.

Plan

Ein Verweis auf die Zeitplankomponente, die wir in den vorhergehenden Zeilen der JSON-Datei erstellt haben.

filePath

Den Pfad zu den Daten, die mit dem Datenknoten verknüpft sind, der in diesem Beispiel eine CSV-Ausgabedatei ist.

Name

Den benutzerdefinierten Namen, der nur als Referenz dient.

Typ

Der Pipeline-Objekttyp, der S3 ist, DataNode um dem Ort zu entsprechen, an dem sich die Daten befinden, in einem Amazon S3-Bucket.

Ressource

Dies ist eine Definition der Rechenressource, die die Kopieroperation ausführt. In diesem Beispiel sollte AWS Data Pipeline automatisch eine EC2-Instance erstellen, um die Kopieraufgabe auszuführen und die Ressource zu beenden, nachdem die Aufgabe abgeschlossen wurde. Die hier definierten Felder steuern die Erstellung und Funktion der EC2 Instance, die die Arbeit erledigt. Die EC2Resource ist durch folgende Felder definiert:

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

```
},
```

Id

Die benutzerdefinierte ID, die nur als Referenz dient.

Plan

Der Zeitplan für die Erstellung dieser Rechenressource.

Name

Den benutzerdefinierten Namen, der nur als Referenz dient.

Rolle

Die IAM-Rolle des Kontos, das auf Ressourcen zugreift, z. B. auf einen Amazon S3-Bucket zugreift, um Daten abzurufen.

Typ

Die Art der Rechenressource zur Ausführung der Arbeit; in diesem Fall eine EC2 Instance. Es sind andere Ressourcentypen verfügbar, z. B. ein EmrCluster Typ.

resourceRole

Die IAM-Rolle des Kontos, das Ressourcen erstellt, z. B. für Sie eine EC2 Instance erstellt und konfiguriert. Rolle und ResourceRole können dieselbe Rolle sein, bieten jedoch getrennt voneinander eine größere Granularität in Ihrer Sicherheitskonfiguration.

Aktivität

Der letzte Abschnitt in der JSON-Datei ist die Definition der Aktivität, die die auszuführende Arbeit darstellt. In diesem Fall verwenden wir eine CopyActivity Komponente, um Daten aus einer Datei in einem Amazon S3-Bucket in eine andere Datei zu kopieren. Die CopyActivity-Komponente ist durch folgende Felder definiert:

```
{
  "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

Die benutzerdefinierte ID, die nur als Referenz dient

Eingabe

Den Speicherort der zu kopierenden MySQL-Daten

Plan

Den Zeitplan für die Ausführung dieser Aktivität

Name

Den benutzerdefinierten Namen, der nur als Referenz dient

runsOn

Die Datenverarbeitungsressource, die die Arbeit ausführt, die diese Aktivität definiert. In diesem Beispiel stellen wir einen Verweis auf die zuvor definierte EC2 Instance bereit. Durch die Verwendung des Felds `runsOn` wird AWS Data Pipeline veranlasst, die EC2-Instance für Sie zu erstellen. Das Feld `runsOn` zeigt an, dass die Ressource in der AWS-Infrastruktur vorhanden ist, während der Wert workerGroup angibt, dass Sie Ihre eigenen lokalen Ressourcen zur Ausführung der Arbeit verwenden möchten.

onSuccess

Den SnsAlarm, der versendet werden soll, wenn die Aktivität erfolgreich abgeschlossen wurde.

onFail

Den SnsAlarm, der versendet werden soll, wenn die Aktivität fehlschlägt.

Ausgabe

Der Amazon S3-Speicherort der CSV-Ausgabedatei

Typ

Den Typ der Aktivität, die durchgeführt werden soll.

Hochladen und Aktivieren der Pipeline-Definition

Sie müssen Ihre Pipeline-Definition hochladen und Ihre Pipeline aktivieren. Ersetzen Sie in den folgenden Beispielbefehlen *pipeline_name* durch ein Label für Ihre Pipeline und *pipeline_file durch den vollständig qualifizierten Pfad für die Pipeline-Definitionsdatei*.`.json`

AWS CLI

Verwenden Sie den folgenden Befehl create-pipeline, um Ihre Pipeline-Definition zu erstellen und Ihre Pipeline zu aktivieren. Notieren Sie sich die ID Ihrer Pipeline, da Sie diesen Wert bei den meisten CLI-Befehlen verwenden werden.

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

Verwenden Sie den folgenden put-pipeline-definitionBefehl, um Ihre Pipeline-Definition hochzuladen.

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

Wenn Ihre Pipeline erfolgreich validiert wurde, ist das `validationErrors` Feld leer. Sie sollten alle Warnungen überprüfen.

Verwenden Sie den folgenden Befehl activate-pipeline, um Ihre Pipeline zu aktivieren.

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

Mit dem folgenden Befehl list-pipelines können Sie überprüfen, ob Ihre Pipeline in der Pipelineliste erscheint.

```
aws datapipeline list-pipelines
```

Kopieren Sie Daten nach Amazon Redshift mit AWS Data Pipeline

Dieses Tutorial führt Sie durch den Prozess der Erstellung einer Pipeline, die regelmäßig Daten von Amazon S3 nach Amazon Redshift verschiebt, indem Sie entweder die Vorlage „In Redshift kopieren“ in der AWS Data Pipeline Konsole oder eine Pipeline-Definitionsdatei mit der AWS Data Pipeline CLI verwenden.

Amazon S3 ist ein Webservice, mit dem Sie Daten in der Cloud speichern können. Weitere Informationen finden Sie im Benutzerhandbuch für Amazon Simple Storage Service.

Amazon Redshift ist ein Data Warehouse-Service in der Cloud. Weitere Informationen finden Sie im Amazon Redshift Management Guide.

Für dieses Tutorial gibt es mehrere Voraussetzungen. Nachdem Sie die folgenden Schritte ausgeführt haben, können Sie das Tutorial über die Konsole oder die Befehlszeile fortsetzen.

Inhalt

Bevor Sie beginnen: Konfigurieren Sie COPY-Optionen und laden Sie Daten

Bevor Sie Daten innerhalb von Amazon Redshift kopieren, stellen Sie sicherAWS Data Pipeline, dass Sie:

- Laden Sie Daten aus Amazon S3.
- Richten Sie die `COPY` Aktivität in Amazon Redshift ein.

Wenn Sie diese Optionen aktiviert und erfolgreich Daten geladen haben, übertragen Sie diese Optionen in AWS Data Pipeline, um den Kopiervorgang darin auszuführen.

COPYOptionen finden Sie unter COPY im Amazon Redshift Database Developer Guide.

Schritte zum Laden von Daten aus Amazon S3 finden Sie unter Daten aus Amazon S3 laden im Amazon Redshift Database Developer Guide.

Beispielsweise erstellt der folgende SQL-Befehl in Amazon Redshift eine neue Tabelle mit dem Namen LISTING und kopiert Beispieldaten aus einem öffentlich verfügbaren Bucket in Amazon S3.

Ersetzen Sie den <iam-role-arn> und die Region durch Ihre eigenen Werte.

Einzelheiten zu diesem Beispiel finden Sie unter Laden von Beispieldaten aus Amazon S3 im Amazon Redshift Getting Started Guide.

```
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';
```

Pipeline einrichten, Sicherheitsgruppe erstellen und Amazon Redshift-Cluster erstellen

So richten Sie Ihr System für das Tutorial ein

1. Führen Sie die Aufgaben unter au Einrichten für AWS Data Pipeline.
2. Erstellen einer Sicherheitsgruppe.
    a. Öffnen Sie die Amazon EC2-Konsole.
    b. Klicken Sie im Navigationsbereich auf Security Groups.
    c. Klicken Sie auf Create Security Group.
    d. Geben Sie einen Namen und eine Beschreibung für die Sicherheitsgruppe an.
    e. [EC2-Classic] Wählen Sie für No VPCVPC aus.

f. [EC2-VPC] Wählen Sie für VPC die ID Ihres VPC aus.

g. Klicken Sie auf Create.

3. [EC2-classic] Erstellen Sie eine Amazon Redshift-Cluster-Sicherheitsgruppe und geben Sie die Amazon EC2-Sicherheitsgruppe an.

   a. Öffnen Sie die Amazon Redshift-Konsole.

   b. Klicken Sie im Navigationsbereich auf Security Groups.

   c. Klicken Sie auf Create Cluster Security Group.

   d. Geben Sie im Dialogfeld Create Cluster Security Group einen Namen und eine Beschreibung für die Cluster-Sicherheitsgruppe an.

   e. Klicken Sie auf den Namen der neuen Cluster-Sicherheitsgruppe.

   f. Klicken Sie auf Add Connection Type.

   g. Wählen Sie im Dialogfeld Add Connection Type die Option EC2 Security Group unter Connection Type aus, wählen Sie die Sicherheitsgruppe aus, die Sie von EC2 Security Group Name erstellt haben, und klicken Sie dann auf Authorize.

4. [EC2-VPC] Erstellen Sie eine Amazon Redshift-Cluster-Sicherheitsgruppe und geben Sie die VPC-Sicherheitsgruppe an.

   a. Öffnen Sie die Amazon EC2-Konsole.

   b. Klicken Sie im Navigationsbereich auf Security Groups.

   c. Klicken Sie auf Create Security Group.

   d. Geben Sie im Dialogfeld Create Security Group einen Namen und eine Beschreibung für die Sicherheitsgruppe an und wählen Sie für VPC die ID Ihres VPC aus.

   e. Klicken Sie auf Add Rule. Geben Sie den Typ, das Protokoll und den Portbereich an und geben Sie die ID der Sicherheitsgruppe in Source ein. Wählen Sie die Sicherheitsgruppe aus, die Sie im zweiten Schritt erstellt haben.

   f. Klicken Sie auf Create.

5. Nachfolgend finden Sie eine kurze Zusammenfassung der Schritte.

   Wenn Sie über einen bestehenden Amazon Redshift-Cluster verfügen, notieren Sie sich die Cluster-ID.

   Um einen neuen Cluster zu erstellen und Beispieldaten zu laden, folgen Sie den Schritten unter Erste Schritte mit Amazon Redshift. Weitere Informationen zum Erstellen von Clustern finden Sie unter Creating a Cluster im Amazon Redshift Management Guide.

a. Öffnen Sie die Amazon Redshift-Konsole.

b. Klicken Sie auf Launch Cluster.

c. Geben Sie die erforderlichen Details für Ihren Cluster an und klicken Sie dann auf Continue.

d. Geben Sie die Knotenkonfiguration an und klicken Sie dann auf Continue.

e. Wählen Sie auf der Seite für zusätzliche Konfigurationsdaten die von Ihnen erstellte Cluster-Sicherheitsgruppe aus und klicken Sie dann auf Continue.

f. Überprüfen Sie die Spezifikationen für Ihren Cluster und klicken Sie dann auf Launch Cluster.

Daten über die Befehlszeile nach Amazon Redshift kopieren

Dieses Tutorial zeigt, wie Sie Daten von Amazon S3 nach Amazon Redshift kopieren. Sie erstellen eine neue Tabelle in Amazon Redshift und verwenden sie dann, AWS Data Pipeline um Daten aus einem öffentlichen Amazon S3-Bucket, der Beispieleingabedaten im CSV-Format enthält, in diese Tabelle zu übertragen. Die Protokolle werden in einem Amazon S3-Bucket gespeichert, dessen Eigentümer Sie sind.

Amazon S3 ist ein Webservice, mit dem Sie Daten in der Cloud speichern können. Weitere Informationen finden Sie im Benutzerhandbuch für Amazon Simple Storage Service. Amazon Redshift ist ein Data Warehouse-Service in der Cloud. Weitere Informationen finden Sie im Amazon Redshift Management Guide.

Voraussetzungen

Bevor Sie beginnen, müssen Sie die folgenden Schritte ausführen:

1. Installieren und konfigurieren Sie eine Befehlszeilenschnittstelle (CLI). Weitere Informationen finden Sie unter Zugriff auf AWS Data Pipeline.
2. Stellen Sie sicher, dass die IAM-Rollen benannt DataPipelineDefaultResourceRolesind DataPipelineDefaultRoleund existieren. Die AWS Data Pipeline Konsole erstellt diese Rollen automatisch für Sie. Wenn Sie die AWS Data Pipeline Konsole nicht mindestens einmal verwendet haben, müssen Sie diese Rollen manuell erstellen. Weitere Informationen finden Sie unter IAM-Rollen für AWS Data Pipeline.
3. Richten Sie den `COPY` Befehl in Amazon Redshift ein, da dieselben Optionen funktionieren müssen, wenn Sie das Kopieren innerhalb von AWS Data Pipeline Amazon Redshift

durchführen. Weitere Informationen finden Sie unter Bevor Sie beginnen: Konfigurieren Sie COPY-Optionen und laden Sie Daten.

4. Richten Sie eine Amazon Redshift-Datenbank ein. Weitere Informationen finden Sie unter Pipeline einrichten, Sicherheitsgruppe erstellen und Amazon Redshift-Cluster erstellen.

Aufgaben

- Definieren Sie eine Pipeline im JSON-Format
- Hochladen und Aktivieren der Pipeline-Definition

Definieren Sie eine Pipeline im JSON-Format

Dieses Beispielszenario zeigt, wie Daten aus einem Amazon S3-Bucket nach Amazon Redshift kopiert werden.

Dies ist die vollständige Pipeline-Definition-JSON-Datei, gefolgt von einer Erläuterung für jeden ihrer Abschnitte. Wir empfehlen, dass Sie einen Texteditor verwenden, mit dem Sie die Syntax von JSON-formatierten Dateien überprüfen können, und die Datei mit der Dateierweiterung `.json` benennen.

```
{
  "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"
      }
    }
  ]
}
```

Weitere Informationen zu diesen Objekten finden Sie in der folgenden Dokumentation.

Objekte

- Datenknoten
- Ressource
- Aktivität

Datenknoten

Bei diesem Beispiel werden ein Eingabedatenknoten, ein Ausgabedatenknoten und eine Datenbank verwendet.

Eingabedatenknoten

Die `S3DataNode` Eingabe-Pipeline-Komponente definiert den Speicherort der Eingabedaten in Amazon S3 und das Datenformat der Eingabedaten. Weitere Informationen finden Sie unter S3 DataNode.

Diese Eingabekomponente wird durch folgende Felder definiert:

```
{
  "id": "S3DataNodeId1",
  "schedule": {
    "ref": "ScheduleId1"
  },
  "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
  "name": "DefaultS3DataNode1",
  "dataFormat": {
    "ref": "CSVId1"
  },
  "type": "S3DataNode"
},
```

`id`

Die benutzerdefinierte ID, die nur als Referenz dient.

`schedule`

Einen Verweis auf die Zeitplankomponente.

`filePath`

Den Pfad zu den Daten, die mit dem Datenknoten verknüpft sind, der in diesem Beispiel eine CSV-Eingabedatei ist.

`name`

Den benutzerdefinierten Namen, der nur als Referenz dient.

`dataFormat`

Einen Verweis auf das Format der Daten für die Aktivitätsverarbeitung.

Ausgabedatenknoten

Die `RedshiftDataNode` Output-Pipeline-Komponente definiert einen Speicherort für die Ausgabedaten, in diesem Fall eine Tabelle in einer Amazon Redshift-Datenbank. Weitere

Informationen finden Sie unter RedshiftDataNode. Diese Ausgabekomponente wird durch folgende Felder definiert:

```
{
  "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`

Die benutzerdefinierte ID, die nur als Referenz dient.

`schedule`

Einen Verweis auf die Zeitplankomponente.

`tableName`

Der Name der Amazon Redshift-Tabelle.

`name`

Den benutzerdefinierten Namen, der nur als Referenz dient.

`createTableSql`

Einen SQL-Ausdruck, der die Tabelle in der Datenbank erstellt.

`database`

Ein Verweis auf die Amazon Redshift-Datenbank.

Datenbank

Die `RedshiftDatabase`-Komponente wird durch folgende Felder definiert: Weitere Informationen finden Sie unter RedshiftDatabase.

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

`id`

Die benutzerdefinierte ID, die nur als Referenz dient.

`databaseName`

Den Namen der logischen Datenbank.

`username`

Den Benutzernamen für die Verbindung zur Datenbank.

`name`

Den benutzerdefinierten Namen, der nur als Referenz dient.

`password`

Das Passwort für die Verbindung zur Datenbank.

`clusterId`

Die ID des Redshift-Clusters.

Ressource

Dies ist eine Definition der Rechenressource, die die Kopieroperation ausführt. In diesem Beispiel sollte AWS Data Pipeline automatisch eine EC2-Instance erstellen, um die Kopieraufgabe auszuführen und die Instance zu beenden, nachdem die Aufgabe abgeschlossen wurde. Die hier definierten Felder steuern die Erstellung und Funktion der Instance, die die Arbeit erledigt. Weitere Informationen finden Sie unter Ec2Resource.

Die `Ec2Resource` wird durch folgende Felder definiert:

```
{
  "id": "Ec2ResourceId1",
  "schedule": {
    "ref": "ScheduleId1"
  },
  "securityGroups": "MySecurityGroup",
  "name": "DefaultEc2Resource1",
  "role": "DataPipelineDefaultRole",
  "logUri": "s3://myLogs",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "type": "Ec2Resource"
},
```

`id`

Die benutzerdefinierte ID, die nur als Referenz dient.

`schedule`

Der Zeitplan für die Erstellung dieser Rechenressource.

`securityGroups`

Die Sicherheitsgruppe, die für die Instances im Ressourcenpool verwendet werden soll.

`name`

Den benutzerdefinierten Namen, der nur als Referenz dient.

`role`

Die IAM-Rolle des Kontos, das auf Ressourcen zugreift, z. B. auf einen Amazon S3-Bucket zugreift, um Daten abzurufen.

`logUri`

Der Amazon S3-Zielpfad zum Sichern von Task Runner-Protokollen aus dem`Ec2Resource`.

`resourceRole`

Die IAM-Rolle des Kontos, das Ressourcen erstellt, z. B. für Sie eine EC2 Instance erstellt und konfiguriert. Rolle und ResourceRole können dieselbe Rolle sein, bieten jedoch getrennt voneinander eine größere Granularität in Ihrer Sicherheitskonfiguration.

Aktivität

Der letzte Abschnitt in der JSON-Datei ist die Definition der Aktivität, die die auszuführende Arbeit darstellt. In diesem Fall verwenden wir eine `RedshiftCopyActivity` Komponente, um Daten von Amazon S3 nach Amazon Redshift zu kopieren. Weitere Informationen finden Sie unter RedshiftCopyActivity.

Die `RedshiftCopyActivity`-Komponente ist durch folgende Felder definiert:

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

`id`

Die benutzerdefinierte ID, die nur als Referenz dient.

`input`

Ein Verweis auf die Amazon S3-Quelldatei.

`schedule`

Der Zeitplan für die Ausführung dieser Aktivität.

`insertMode`

Der Einfügetyp (`KEEP_EXISTING`, `OVERWRITE_EXISTING` oder `TRUNCATE`).

`name`

Den benutzerdefinierten Namen, der nur als Referenz dient.

`runsOn`

Die Datenverarbeitungsressource, die die Arbeit ausführt, die diese Aktivität definiert.

`output`

Ein Verweis auf die Amazon Redshift-Zieltabelle.

Hochladen und Aktivieren der Pipeline-Definition

Sie müssen Ihre Pipeline-Definition hochladen und Ihre Pipeline aktivieren. Ersetzen Sie in den folgenden Beispielbefehlen *`pipeline_name`* durch ein Label für Ihre Pipeline und *`pipeline_file durch den vollständig qualifizierten Pfad für die Pipeline-Definitionsdatei`*`.json`

AWS CLI

Verwenden Sie den folgenden Befehl create-pipeline, um Ihre Pipeline-Definition zu erstellen und Ihre Pipeline zu aktivieren. Notieren Sie sich die ID Ihrer Pipeline, da Sie diesen Wert bei den meisten CLI-Befehlen verwenden werden.

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

Verwenden Sie den folgenden put-pipeline-definitionBefehl, um Ihre Pipeline-Definition hochzuladen.

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

Wenn Ihre Pipeline erfolgreich validiert wurde, ist das `validationErrors` Feld leer. Sie sollten alle Warnungen überprüfen.

Verwenden Sie den folgenden Befehl activate-pipeline, um Ihre Pipeline zu aktivieren.

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

Mit dem folgenden Befehl list-pipelines können Sie überprüfen, ob Ihre Pipeline in der Pipelineliste erscheint.

```
aws datapipeline list-pipelines
```

Pipeline-Ausdrücke und -Funktionen

Dieser Abschnitt enthält Informationen zur Syntax der in Pipelines verwendeten Ausdrücke und Funktionen sowie zu den zugehörigen Datentypen.

Einfache Datentypen

Die folgenden Datentypen sind als Feldwerte zulässig.

Typen

- DateTime
- Numerischer Wert
- Objektverweise
- Intervall
- Zeichenfolge

DateTime

AWS Data Pipeline unterstützt Datums- und Uhrzeitangaben nur als UTC/GMT-Zeit im Format "JJJJ-MM-TTTHH:MM:SS". Im folgenden Beispiel wird dem Feld `startDateTime` eines `Schedule`-Objekts der Wert `1/15/2012, 11:59 p.m.` in der UTC/GMT-Zeitzone zugewiesen.

```
"startDateTime" : "2012-01-15T23:59:00"
```

Numerischer Wert

AWS Data Pipeline unterstützt sowohl Ganzzahlen als auch Gleitkommazahlen.

Objektverweise

Dies ist ein Objekt in der Pipeline-Definition. Dabei kann es sich um das aktuelle Objekt, um den Namen eines woanders in der Pipeline definierten Objekts oder um ein Objekt mit einem Feld handeln, in dem mit dem Schlüsselwort `node` auf das aktuelle Objekt verwiesen wird. Mehr über `node` erfahren Sie unter Verweisen auf Felder und Objekte. Weitere Informationen zu den Pipeline-Objekttypen finden Sie unter Pipeline-Objektreferenz.

Intervall

Gibt an, wie oft ein geplantes Ereignis ausgeführt werden soll. Die Angabe erfolgt im Format "N [`years|months|weeks|days|hours|minutes`]", wobei N eine positive Ganzzahl ist.

Der Mindestzeitraum beträgt 15 Minuten und der maximale Zeitraum beträgt 3 Jahre.

Im folgenden Beispiel wird das Feld `period` des Objekts `Schedule` auf 3 Stunden eingestellt. Dadurch wird ein geplantes Ereignis alle drei Stunden ausgeführt.

```
"period" : "3 hours"
```

Zeichenfolge

Standard-Zeichenfolgenwerte. Zeichenfolgen müssen in Anführungszeichen (') eingeschlossen werden. Der umgekehrte Schrägstrich (\) kann als Escape-Zeichen in einer Zeichenfolge verwendet werden. Mehrzeilige Zeichenfolgen werden nicht unterstützt.

Die folgenden Beispiele sind gültige Zeichenfolgenwerte für das Feld `id`.

```
"id" : "My Data Object"

"id" : "My \"Data\" Object"
```

Zeichenfolgen können auch Ausdrücke enthalten, die zu Zeichenfolgenwerten ausgewertet werden. Diese Ausdrücke müssen in der Zeichenfolge zwischen den Trennzeichen "#{" und "}" stehen. Im folgenden Beispiel wird mit einem Ausdruck der Name des aktuellen Objekts in einen Pfad eingefügt.

```
"filePath" : "s3://myBucket/#{name}.csv"
```

Weitere Informationen zur Arbeit mit Ausdrücken finden Sie unter Verweisen auf Felder und Objekte und Ausdrucksauswertung.

Ausdrücke

Ausdrücke ermöglichen die Nutzung eines Werts in mehreren zusammengehörigen Objekten. Ausdrücke werden vom AWS Data Pipeline-Web-Service zur Laufzeit verarbeitet. Dadurch wird sichergestellt, dass alle Ausdrücke durch deren Werte ersetzt werden.

Ausdrücke müssen in die "#{" und "}" Trennzeichen eingeschlossen werden. Ein Ausdruck kann in jedem Pipeline-Definitionsobjekt verwendet werden, in dem Zeichenfolgen zulässig sind. Wenn ein Slot ein Verweis ist bzw. den Typ ID, NAME, TYPE oder SPHERE hat, wird sein Wert nicht ausgewertet und er wird unverändert übernommen.

Der folgende Ausdruck ruft eine der AWS Data Pipeline-Funktionen auf. Weitere Informationen finden Sie unter Ausdrucksauswertung .

```
#{format(myDateTime,'YYYY-MM-dd hh:mm:ss')}
```

Verweisen auf Felder und Objekte

Ausdrücke können Felder des aktuellen Objekts, in dem sie sich befinden, oder Felder eines anderen Objekts verwenden, das durch einen Verweis verknüpft ist.

Ein Slot-Format besteht aus einem Zeitpunkt der Erstellung, gefolgt von dem Zeitpunkt der Objekterstellung, z. B. `@S3BackupLocation_2018-01-31T11:05:33`.

Sie können auch auf die genaue Slot-ID verweisen, die in der Pipeline-Definition angegeben ist, wie zum Beispiel die Slot-ID des Amazon S3-Sicherungsspeicherorts. Um auf die Slot-ID zu verweisen, verwenden Sie `#{parent.@id}`.

Im folgenden Beispiel verweist das Feld `filePath` auf das Feld `id` desselben Objekts, um einen Dateinamen zu bilden. Der Wert von `filePath` ergibt sich als „`s3://mybucket/ExampleDataNode.csv`“.

```
{
  "id" : "ExampleDataNode",
  "type" : "S3DataNode",
  "schedule" : {"ref" : "ExampleSchedule"},
  "filePath" : "s3://mybucket/#{parent.@id}.csv",
  "precondition" : {"ref" : "ExampleCondition"},
  "onFail" : {"ref" : "FailureNotify"}
}
```

Um auf das Feld eines anderen, durch einen Verweis verknüpften Objekts zuzugreifen, muss das Schlüsselwort `node` angegeben werden. Dieses Schlüsselwort ist nur für Alarm- und Vorbedingungsobjekte verfügbar.

Zurück zum obigen Beispiel: Ein Ausdruck in einem `SnsAlarm`-Objekt kann auf den Datums- und Uhrzeitbereich in einem `Schedule`-Objekt zugreifen, da das `S3DataNode`-Objekt auf beide verweist.

Insbesondere kann im `FailureNotify`-Objekt das Feld `message` die Laufzeitfelder `@scheduledStartTime` und `@scheduledEndTime` von `ExampleSchedule` verwenden, da im `ExampleDataNode`-Objekt das Feld `onFail` auf `FailureNotify` und das Feld `schedule` auf `ExampleSchedule` verweist.

```
{
    "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**
>
> Sie können Pipelines erstellen, die Abhängigkeiten enthalten, wie z. B. Pipeline-Aufgaben, die von der Verarbeitung anderer Systeme oder Aufgaben abhängen. Wenn Ihre Pipeline bestimmte Ressourcen erfordert, fügen Sie diese Abhängigkeiten mithilfe von Vorbedingungen hinzu, die Sie dann den gewünschten Datenknoten und Aufgaben zuordnen. Die Pipeline lässt sich dann einfacher debuggen und ist weniger fehleranfällig. Verwenden Sie die Abhängigkeiten möglichst in einer einzigen Pipeline, da die Pipeline-übergreifende Fehlersuche schwierig ist.

Verschachtelte Ausdrücke

Sie können in AWS Data Pipeline Werte verschachteln, um komplexere Ausdrücke zu erstellen. Um beispielsweise eine Zeitberechnung durchzuführen (30 Minuten von `scheduledStartTime` subtrahieren) und das Ergebnis für die Verwendung in einer Pipeline-Definition zu formatieren, könnten Sie folgenden Ausdruck in einer Aktivität benutzen

```
#{format(minusMinutes(@scheduledStartTime,30),'YYYY-MM-dd hh:mm:ss')}
```

und das Präfix node angeben, wenn der Ausdruck Teil eines SnsAlarm- oder Precondition-Objekts ist:

```
#{format(minusMinutes(node.@scheduledStartTime,30),'YYYY-MM-dd hh:mm:ss')}
```

Listen

In Ausdrücken können auch Listen verwendet werden. Gehen wir von dieser Listendefinition aus: `"myList":["one","two"]`aus. Wenn Sie diese Liste im Ausdruck verwenden`#{'this is ' + myList}`, wird es auswerten`["this is one", "this is two"]`aus. Wenn zwei Listen vorhanden sind, werden diese von Data Pipeline bei der Auswertung "abgeflacht" (flattened). Ist beispielsweise `myList1` als `[1,2]` definiert und `myList2` als `[3,4]`, dann wird der Ausdruck `[#{myList1}, #{myList2}]` zu `[1,2,3,4]` ausgewertet.

Knotenausdruck

AWS Data Pipeline verwendet den Ausdruck `#{node.*}` in `SnsAlarm` oder `PreCondition` als Rückverweis auf das übergeordnete Objekt einer Pipeline-Komponente. Da der Verweis auf `SnsAlarm` und `PreCondition` in einer Aktivität oder Ressource ohne Rückverweis aus diesen Objekten erfolgt, bietet `node` die Möglichkeit, das verweisende Objekt zu referenzieren. In der folgenden Pipeline-Definition wird beispielsweise gezeigt, wie in einer Fehlerbenachrichtigung mit `node` auf deren übergeordnetes Objekt (`ShellCommandActivity`) verwiesen und wie dessen geplante Start- und Endzeit in die `SnsAlarm`-Nachricht eingefügt werden. Dem scheduledStartTime-Verweis im ShellCommandActivity-Objekt muss das Präfix `node` nicht hinzugefügt werden, da scheduledStartTime auf sich selbst verweist.

> **Note**
>
> Die mit dem Zeichen "@" beginnenden Felder sind Laufzeitfelder.

```
{
  "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"
},
```

AWS Data Pipeline unterstützt transitive Verweise bei benutzerdefinierten Felder, jedoch nicht bei Laufzeitfeldern. Ein transitiver Verweis ist ein Verweis zwischen zwei Pipeline-Komponenten, die von einer anderen als "Vermittler" dienenden Pipeline-Komponente abhängen. Das folgende Beispiel zeigt einen Verweis auf ein transitives benutzerdefiniertes Feld und einen Verweis auf ein nichttransitives Laufzeitfeld. Beide Verweise sind zulässig. Weitere Informationen finden Sie unter Benutzerdefinierte Felder .

```
{
  "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"
}
```

Ausdrucksauswertung

AWS Data Pipeline stellt eine Reihe von Funktionen bereit, mit denen der Wert eines Feldes berechnet werden kann. Im folgenden Beispiel wird die Funktion `makeDate` verwendet, um dem Feld `startDateTime` eines `Schedule`-Objekts die GMT/UTC-Zeit `"2011-05-24T0:00:00"` zuzuweisen.

```
"startDateTime" : "makeDate(2011,5,24)"
```

Mathematische Funktionen

Die folgenden Funktionen sind für die Arbeit mit numerischen Werten verfügbar.

Funktion	Beschreibung
+	Addition. Beispiel: `#{1 + 2}` Ergebnis: `3`
-	Subtraktion. Beispiel: `#{1 - 2}` Ergebnis: `-1`
*	Multiplikation. Beispiel: `#{1 * 2}` Ergebnis: `2`
/	Division. Wenn Sie zwei Ganzzahlen dividieren, werden die Nachkommastellen abgeschnitten. Beispiel: `#{1 / 2}`, Ergebnis:`0` Beispiel: `#{1.0 / 2}`, Ergebnis:`.5`
^	Exponent. Beispiel: `#{2 ^ 2}` Ergebnis: `4.0`

Funktionen für Zeichenfolgen

Die folgenden Funktionen sind für die Arbeit mit Zeichenfolgen verfügbar.

Funktion	Beschreibung
+	Verkettung. Werte mit einem anderen Typ werden zuerst in Zeichenfolgen konvertiert. Beispiel: `#{"hel" + "lo"}` Ergebnis: `"hello"`

Datums- und Zeitfunktionen

Die folgenden Funktionen sind für die Arbeit mit DateTime-Werten verfügbar. In den Beispielen hat `myDateTime` den Wert `May 24, 2011 @ 5:10 pm GMT`.

> **Note**
>
> Das Datums-/Uhrzeitformat für AWS Data Pipeline ist Joda-Time. Dabei handelt es sich um einen Ersatz für die Datums- und Zeitklassen von Java. Weitere Informationen finden Sie unter Joda Time - Class DateTimeFormat.

Funktion	Beschreibung
`int day(DateTime myDateTime)`	Gibt den Tag des angegeben en DateTime-Wertes als Ganzzahl zurück. Beispiel: `#{day(myD ateTime)}` Ergebnis: `24`
`int dayOfYear(DateTime myDateTime)`	Gibt den Tag des Jahres des angegebenen DateTime-Wertes als Ganzzahl zurück.

Funktion	Beschreibung
	Beispiel: `#{dayOfYear(myDateTime)}` Ergebnis: `144`
`DateTime firstOfMonth(DateTime myDateTime)`	Erstellt ein DateTime-Objekt für den Monatsbeginn im angegebenen DateTime-Wert. Beispiel: `#{firstOfMonth(myDateTime)}` Ergebnis: `"2011-05-01T17:10:00z"`
`String format(DateTime myDateTime,String format)`	Konvertiert das angegebene DateTime-Objekt entsprechend der übergebenen Formatzeichenfolge und erstellt aus dem Ergebnis ein Zeichenfolgenobjekt. Beispiel: `#{format(myDateTime,'YYYY-MM-dd HH:mm:ss z')}` Ergebnis: `"2011-05-24T17:10:00 UTC"`
`int hour(DateTime myDateTime)`	Gibt die Stunde des angegebenen DateTime-Wertes als Ganzzahl zurück. Beispiel: `#{hour(myDateTime)}` Ergebnis: `17`

Funktion	Beschreibung
`DateTime makeDate(int year,int month,int day)`	Erstellt ein DateTime-Objekt (UTC) mit dem angegebenen Jahr, Monat und Tag um Mitternacht. Beispiel: `#{makeDate(2011,5,24)}` Ergebnis: `"2011-05-24T0:00:00z"`
`DateTime makeDateTime(int year,int month,int day,int hour,int minute)`	Erstellt ein DateTime-Objekt (UTC) mit den angegebenen Werten für Jahr, Monat, Tag, Stunde und Minute. Beispiel: `#{makeDateTime(2011,5,24,14,21)}` Ergebnis: `"2011-05-24T14:21:00z"`
`DateTime midnight(DateTime myDateTime)`	Erstellt ein DateTime-Objekt für die aktuelle Mitternacht relativ zum angegebenen DateTime-Wert. Hat beispielsweise `MyDateTime` den Wert `2011-05-25T17:10:00z`, lautet das Ergebnis wie folgt. Beispiel: `#{midnight(myDateTime)}` Ergebnis: `"2011-05-25T0:00:00z"`

Funktion	Beschreibung
`DateTime minusDays(DateTime myDateTime,int daysToSub)`	Subtrahiert die angegebene Anzahl von Tagen vom übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{minusDays(myDateTime,1)}` Ergebnis: `"2011-05-23T17:10:00z"`
`DateTime minusHours(DateTime myDateTime,int hoursToSub)`	Subtrahiert die angegebene Anzahl von Stunden vom übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{minusHours(myDateTime,1)}` Ergebnis: `"2011-05-24T16:10:00z"`
`DateTime minusMinutes(DateTime myDateTime,int minutesToSub)`	Subtrahiert die angegebene Anzahl von Minuten vom übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{minusMinutes(myDateTime,1)}` Ergebnis: `"2011-05-24T17:09:00z"`

Funktion	Beschreibung
`DateTime minusMonths(DateTime myDateTime,int monthsToSub)`	Subtrahiert die angegebene Anzahl von Monaten vom übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{minusMonths(myDateTime,1)}` Ergebnis: `"2011-04-24T17:10:00z"`
`DateTime minusWeeks(DateTime myDateTime,int weeksToSub)`	Subtrahiert die angegebene Anzahl von Wochen vom übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{minusWeeks(myDateTime,1)}` Ergebnis: `"2011-05-17T17:10:00z"`
`DateTime minusYears(DateTime myDateTime,int yearsToSub)`	Subtrahiert die angegebene Anzahl von Jahren vom übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{minusYears(myDateTime,1)}` Ergebnis: `"2010-05-24T17:10:00z"`

Funktion	Beschreibung
`int minute(DateTime myDateTime)`	Gibt die Minute des angegebenen DateTime-Wertes als Ganzzahl zurück. Beispiel: `#{minute(myDateTime)}` Ergebnis: `10`
`int month(DateTime myDateTime)`	Gibt den Monat des angegebenen DateTime-Wertes als Ganzzahl zurück. Beispiel: `#{month(myDateTime)}` Ergebnis: `5`
`DateTime plusDays(DateTime myDateTime,int daysToAdd)`	Addiert die angegebene Anzahl von Tagen zum übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{plusDays(myDateTime,1)}` Ergebnis: `"2011-05-25T17:10:00z"`

Funktion	Beschreibung
`DateTime plusHours(DateTime myDateTime,int hoursToAdd)`	Addiert die angegebene Anzahl von Stunden zum übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{plusHours(myDateTime,1)}` Ergebnis: `"2011-05-24T18:10:00z"`
`DateTime plusMinutes(DateTime myDateTime,int minutesToAdd)`	Addiert die angegebene Anzahl von Minuten zum übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{plusMinutes(myDateTime,1)}` Ergebnis: `"2011-05-24 17:11:00z"`
`DateTime plusMonths(DateTime myDateTime,int monthsToAdd)`	Addiert die angegebene Anzahl von Monaten zum übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{plusMonths(myDateTime,1)}` Ergebnis: `"2011-06-24T17:10:00z"`

Funktion	Beschreibung
`DateTime plusWeeks(DateTime myDateTime,int weeksToAdd)`	Addiert die angegebene Anzahl von Wochen zum übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{plusWeeks(myDateTime,1)}` Ergebnis: `"2011-05-31T17:10:00z"`
`DateTime plusYears(DateTime myDateTime,int yearsToAdd)`	Addiert die angegebene Anzahl von Jahren zum übergebenen DateTime-Wert und erstellt aus dem Ergebnis ein DateTime-Objekt. Beispiel: `#{plusYears(myDateTime,1)}` Ergebnis: `"2012-05-24T17:10:00z"`
`DateTime sunday(DateTime myDateTime)`	Erstellt ein DateTime-Objekt für den vorherigen Sonntag relativ zum angegebenen DateTime-Wert. Wenn der angegebene DateTime-Wert ein Sonntag ist, wird dieser zurückgegeben. Beispiel: `#{sunday(myDateTime)}` Ergebnis: `"2011-05-22 17:10:00 UTC"`

Funktion	Beschreibung
`int year(DateTime myDateTime)`	Gibt das Jahr des angegeben en DateTime-Wertes als Ganzzahl zurück. Beispiel: `#{year(my DateTime)}` Ergebnis: `2011`
`DateTime yesterday(DateTime myDateTime)`	Erstellt ein DateTime-Objekt für den vorherigen Tag relativ zum angegebenen DateTime-Wert. Das Ergebnis ist identisch mit minusDays(1). Beispiel: `#{yesterd ay(myDateTime)}` Ergebnis: `"2011-05-23T17:10:00z"`

Sonderzeichen

In AWS Data Pipeline werden die in der folgenden Tabelle beschriebenen Sonderzeichen verwendet, die in Pipeline-Definitionen eine spezielle Bedeutung haben.

Sonderzeichen	Beschreibung	Beispiele
@	Laufzeitfeld. Wenn dieses Zeichen dem Namen eines Feldes vorangestellt wird, ist dieses nur während der Ausführung der Pipeline verfügbar.	@actualStartTime @failureReason @resourceStatus

Sonderzeichen	Beschreibung	Beispiele
#	Ausdruck. Ausdrücke werden in die Trennzeichen "#{" und "}" eingeschlossen. Der Inhalt der geschweiften Klammern wird dann von AWS Data Pipeline ausgewertet. Weitere Informationen finden Sie unter Ausdrücke .	#{format(myDateTime,'JJJJ-MM-TT hh:mm:ss')} s3://mybucket/#{id}.csv
*	Verschlüsseltes Feld. Wenn dieses Zeichen dem Namen eines Feldes vorangestellt wird, verschlüsselt AWS Data Pipeline dessen Inhalt bei der Übertragung zwischen der Konsole oder CLI und dem AWS Data Pipeline-Service.	*Passwort

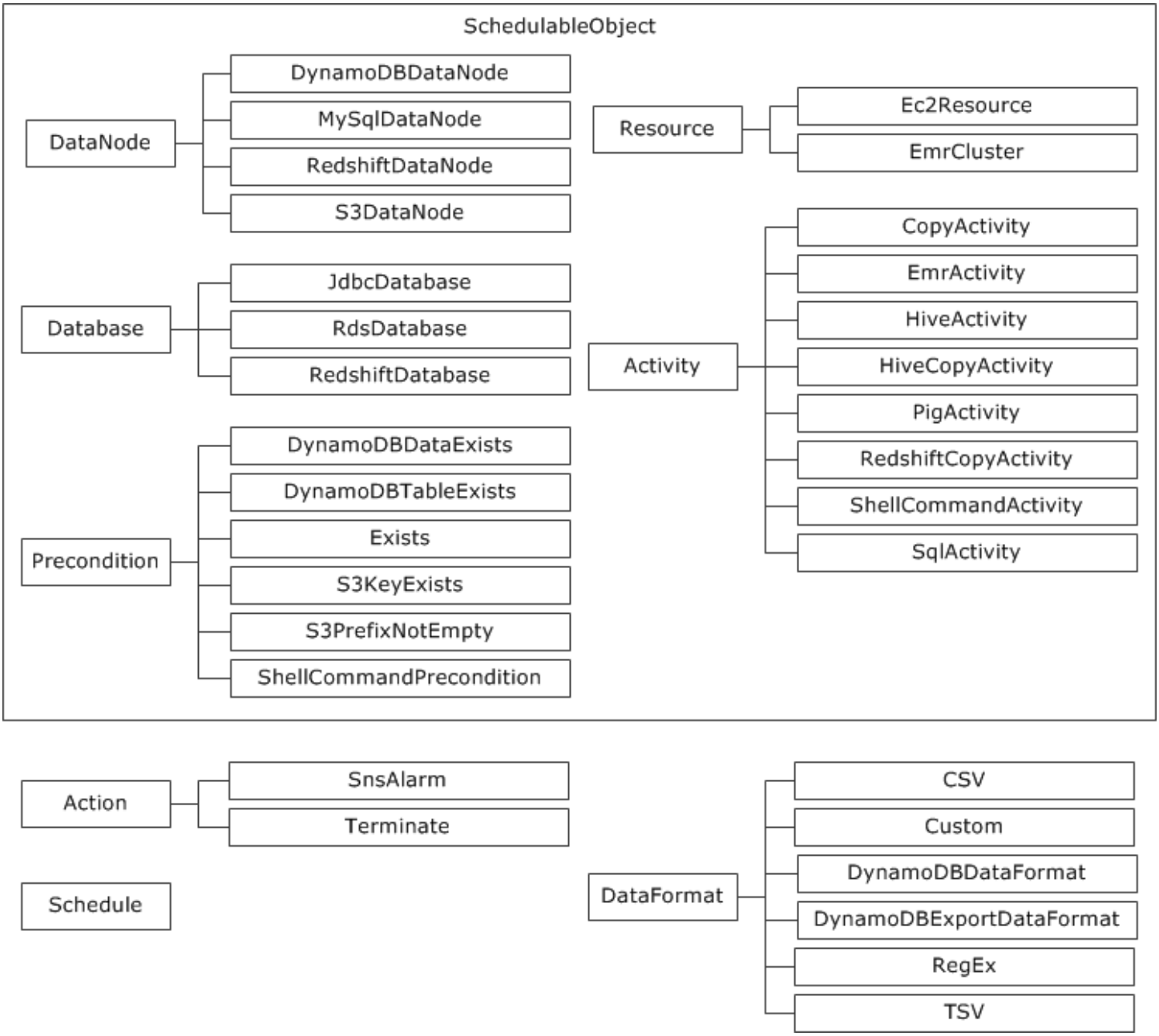
Pipeline-Objektreferenz

Sie können die folgenden Pipeline-Objekte und -Komponenten in Ihrer Pipeline-Definitionsdatei verwenden.

Inhalt

> **Note**
>
> Eine Beispielanwendung, die das AWS Data Pipeline Java-SDK verwendet, finden Sie unter Data Pipeline DynamoDB Export Java Sample on. GitHub

Folgendes ist die Objekthierarchie für AWS Data Pipeline.



Datenknoten

Nachfolgend sind die AWS Data Pipeline-Datenknotenobjekte aufgelistet:

Objekte

- DynamoDB DataNode
- MySqlDataNode
- RedshiftDataNode

- S3 DataNode
- SqlDataNode

DynamoDB DataNode

Definiert mithilfe von DynamoDB einen Datenknoten, der als Eingabe für ein OR-Objekt angegeben wird. `HiveActivity` `EMRActivity`

> **Note**
>
> Das `DynamoDBDataNode`-Objekt unterstützt die Vorbedingung `Exists` nicht.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Dieses Objekt verweist auf zwei andere Objekte, die Sie in derselben Pipeline-Definitionsdatei definieren. `CopyPeriod` ist ein `Schedule`-Objekt und `Ready` ist ein Vorbedingungsobjekt.

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

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
tableName	Die DynamoDB-Tabelle.	String

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer	Referenzobjekt, zum Beispiel „schedule“:

Objektaufruf-Felder	Beschreibung	Slot-Typ
	müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispiels weise „schedule“: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Benutzer ein übergeordnetes Objekt mit Zeitpläne nreferenz erstellen. Weitere Informationen zu optionalen Zeitplankonfigurationen finden Sie unter Zeitplan.	{"ref“:“ myScheduleId „}

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Ak tivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlo ssen. Wenn dieses Feld aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dataFormat	DataFormat für die von diesem Datenknoten beschriebenen Daten. Derzeit unterstützt für HiveActivity und HiveCopyActivity.	Referenzobjekt, „DataFormat“: {"ref“ DataForma tId :"MyDynamoDB „}

Optionale Felder	Beschreibung	Slot-Typ
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt	Referenzobjekt, z. B. „dependSon“: {"ref“:“ „} myActivityId
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String

Optionale Felder	Beschreibung	Slot-Typ
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ „} myPreconditionId
readThroughputPercent	Legt die Rate der Lesevorgänge so fest, dass Ihre von DynamoDB bereitgestellte Durchsatzrate im für Ihre Tabelle zugewiesenen Bereich liegt. Der Wert ist zweistellig und liegt zwischen 0,1 und 1,0 (einschließlich).	Double
Region	Der Code für die Region, in der die DynamoDB-Tabelle vorhanden ist. Beispiel: us-east-1. Dies wird verwendet, HiveActivity wenn es Staging für DynamoDB-Tabellen in Hive durchführt.	Aufzählung
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.	String
writeThroughputPer cent	Legt die Rate der Schreibvorgänge so fest, dass Ihre von DynamoDB bereitgestellte Durchsatzrate im für Ihre Tabelle zugewiese nen Bereich liegt. Der Wert ist zweistellig und liegt zwischen .1 und 1.0 (einschließlich).	Double

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Zuständigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ healthStatusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

MySqlDataNode

Legt ein Datenknoten mit MySQL fest.

> **Note**
>
> Der `MySqlDataNode`-Typ ist veraltet. Stattdessen empfehlen wir, SqlDataNode zu verwenden.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Dieses Objekt verweist auf zwei andere Objekte, die Sie in derselben Pipeline-Definitionsdatei definieren. `CopyPeriod` ist ein `Schedule`-Objekt und `Ready` ist ein Vorbedingungsobjekt.

```
{
```

```
  "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" }
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
Tabelle	Der Name der Tabelle in der MySQL-Dat enbank.	String

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispiels weise „schedule“: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne	Referenzobjekt, z. B. „schedule“: {"ref“:“ mySchedule Id „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	innerhalb des Hauptplans), können Benutzer ein übergeordnetes Objekt mit Zeitpläne nreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
createTableSql	Ein SQL-Tabellenerstellungsausdruck, der die Tabelle erstellt.	String
Datenbank	Name der Datenbank.	Referenzobjekt, z. B. „Datenbank“: {"ref“:“ myDatabaseId „}
dependsOn	Gibt eine Abhängigkeit von einem anderen ausführbaren Objekt an.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
insertQuery	Eine SQL-Anweisung zum Einfügen von Daten in die Tabelle.	String
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. ondemand	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ „} myPrecond itionId

Optionale Felder	Beschreibung	Slot-Typ
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihenstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführen. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
schemaName	Der Name des Schemas für die Tabelle.	String
selectQuery	Eine SQL-Anweisung zum Abrufen von Daten aus der Tabelle.	String
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String

Laufzeitfelder	Beschreibung	Slot-Typ
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ healthStatusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- S3 DataNode

RedshiftDataNode

Definiert einen Datenknoten mithilfe von Amazon Redshift. `RedshiftDataNode`stellt die Eigenschaften der Daten in einer Datenbank dar, z. B. einer Datentabelle, die von Ihrer Pipeline verwendet wird.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "id" : "MyRedshiftDataNode",
  "type" : "RedshiftDataNode",
  "database": { "ref": "MyRedshiftDatabase" },
  "tableName": "adEvents",
  "schedule": { "ref": "Hour" }
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
Datenbank	Die Datenbank, in der die Tabelle gespeichert ist.	Referenzobjekt, z. B. „database“: {"ref“:“ myRedshiftDatabase Id "}
tableName	Der Name der Amazon Redshift-Tabelle. Die Tabelle wird erstellt, falls sie noch nicht existiert und Sie sie angegeben haben createTableSql.	String

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü	Referenzobjekt, z. B. „schedule“: {"ref“:“ mySchedule Id „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispiels weise „schedule“: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Benutzer ein übergeordnetes Objekt mit Zeitpläne nreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Ak tivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlo ssen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
createTableSql	Ein SQL-Ausdruck, der die Tabelle in der Datenbank erstellt. Wir empfehlen, dass Sie das Schema angeben, in dem die Tabelle erstellt werden soll, zum Beispiel: CREATE TABLE mySchema.myTable (bestColumn	String

Optionale Felder	Beschreibung	Slot-Typ
	varchar (25) primary key distkey, integer sortKey). numberOfWins AWS Data Pipelinef ührt das Skript in dem createTableSql Feld aus, wenn die durch TableName angegeben e Tabelle nicht in dem durch das Feld SchemaName angegebenen Schema existiert . Wenn Sie beispielsweise SchemaName als mySchema angeben, mySchema jedoch nicht in das createTableSql Feld aufnehmen, wird die Tabelle im falschen Schema erstellt (standard mäßig würde sie in PUBLIC erstellt). Dies passiert, da AWS Data Pipeline die CREATE-TABLE-Anweisungen nicht parst.	
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt	Referenzobjekt, z. B. „dependSon“: {"ref“:“ „} myActivityId
failureAndRerunMod us	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Die maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl

Optionale Felder	Beschreibung	Slot-Typ
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ „} myPrecond itionId
primaryKeys	Wenn Sie für eine Zieltabelle in `RedShiftC opyActivity` keine primaryKeys festlegen , können Sie eine Liste der Spalten angeben, die primaryKeys nutzen, die als mergeKey fungieren. Wenn Sie jedoch einen vorhandenen PrimaryKey in einer Amazon Redshift Redshift-Tabelle definiert haben, überschreibt diese Einstellung den vorhandenen Schlüssel.	String

Optionale Felder	Beschreibung	Slot-Typ
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
schemaName	In diesem optionalen Feld wird der Name des Schemas für die Amazon Redshift-Tabelle festgelegt. Wenn kein Name festgelegt wird, ist der Schemaname ÖFFENTLICH, was das Standardschema bei Amazon Redshift ist. Weitere Informationen finden Sie im Amazon Redshift Database Developer Guide.	String
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ healthStatusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

S3 DataNode

Definiert einen Datenknoten mit Amazon S3. Standardmäßig DataNode verwendet der S3 serverseitige Verschlüsselung. Wenn Sie dies deaktivieren möchten, setzen Sie s3 EncryptionType auf NONE.

> **Note**
>
> Wenn Sie einen `S3DataNode` als Eingabe für `CopyActivity` nutzen, werden nur die Datenformate CSV und TSV unterstützt.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Dieses Objekt verweist auf ein anderes Objekt, das Sie in derselben Pipeline-Definitionsdatei definieren. `CopyPeriod` ist ein `Schedule`-Objekt.

```
{
  "id" : "OutputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://myBucket/#{@scheduledStartTime}.csv"
}
```

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, z. B. indem sie „schedule “: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanv	Referenzobjekt, z. B. „schedule“: {"ref“:“ mySchedule Id „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	erweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Benutzer ein übergeord netes Objekt mit Zeitplänenreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
Kompression	Die vom S3 beschriebene Art der Komprimierung für die DatenDataNode. „none“ ist keine Komprimierung und „gzip“ wird mit dem Gzip-Algorithmus komprimiert. Dieses Feld wird nur für die Verwendung mit Amazon Redshift und wenn Sie S3 DataNode mit CopyActivity verwenden, unterstützt.	Aufzählung
dataFormat	DataFormat für die in diesem S3 DataNode beschriebenen Daten.	Referenzobjekt, z. B. „dataFormat“: {"ref“:“ myDataFormat Id "}

Optionale Felder	Beschreibung	Slot-Typ
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
directoryPath	Amazon S3 S3-Verzeichnispfad als URI: s3:// my-bucket/my-key-for-directory. Sie müssen entweder einen Dateipfad (filePath) oder einen Wert für directoryPath angeben.	String
failureAndRerunMod us	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
filePath	Der Pfad zum Objekt in Amazon S3 als URI, zum Beispiel: s3://my-bucket/my-key-for-f ile. Sie müssen entweder einen Dateipfad (filePath) oder einen Wert für directoryPath angeben. Diese repräsentieren einen Ordner und einen Dateinamen. Mit dem directoryPath-Wert können Sie mehrere Dateien in einem Verzeichnis unterbringen.	String
lateAfterTimeout	Die verstrichene Zeit nach dem Start der Pipeline, innerhalb derer das Objekt abgeschlo ssen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
manifestFilePath	Der Amazon S3 S3-Pfad zu einer Manifestd atei in dem von Amazon Redshift unterstüt zten Format. AWS Data Pipelineverwendet die Manifestdatei, um die angegebenen Amazon S3 S3-Dateien in die Tabelle zu kopieren. Dieses Feld ist nur gültig, wenn a RedShiftC opyActivity auf den S3 verweistDataNode.	String

Optionale Felder	Beschreibung	Slot-Typ
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ „} myPrecond itionId
reportProgressTime out	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
s3 EncryptionType	Überschreibt den Amazon S3-Verschlüsselung styp. Die Werte sind SERVER_SIDE_ENCRYP TION oder NONE. Die serverseitige Verschlüs selung ist standardmäßig aktiviert.	Aufzählung
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String

Laufzeitfelder	Beschreibung	Slot-Typ
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ healthStatusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref":“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Weitere Informationen finden Sie unter:

- MySqlDataNode

SqlDataNode

Legt ein Datenknoten mit SQL fest.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Dieses Objekt verweist auf zwei andere Objekte, die Sie in derselben Pipeline-Definitionsdatei definieren. `CopyPeriod` ist ein `Schedule`-Objekt und `Ready` ist ein Vorbedingungsobjekt.

```
{
  "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" }
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
Tabelle	Der Name der Tabelle in der SQL-Datenbank.	String

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispiels weise „schedule“: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-	Referenzobjekt, z. B. „schedule“: {"ref“:“ mySchedule Id „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Benutzer ein übergeordnetes Objekt mit Zeitpläne nreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
createTableSql	Ein SQL-Tabellenerstellungsausdruck, der die Tabelle erstellt.	String
Datenbank	Name der Datenbank.	Referenzobjekt, z. B. „Datenbank“: {"ref“:“ myDatabaseId „}
dependsOn	Gibt die Abhängigkeit von einem anderen ausführbaren Objekt an.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}

Optionale Felder	Beschreibung	Slot-Typ
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
insertQuery	Eine SQL-Anweisung zum Einfügen von Daten in die Tabelle.	String
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. ondemand	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String

Optionale Felder	Beschreibung	Slot-Typ
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ „} myPrecond itionId
reportProgressTime out	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung
schemaName	Der Name des Schemas für die Tabelle.	String
selectQuery	Eine SQL-Anweisung zum Abrufen von Daten aus der Tabelle.	String
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ healthStatusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Weitere Informationen finden Sie unter:

- S3 DataNode

Aktivitäten

Nachfolgend sind die AWS Data Pipeline-Aktivitätsobjekte aufgelistet:

Objekte

- CopyActivity
- EmrActivity
- HadoopActivity
- HiveActivity
- HiveCopyActivity
- PigActivity

- RedshiftCopyActivity
- ShellCommandActivity
- SqlActivity

CopyActivity

Kopiert Daten von einem Ort an einen anderen. `CopyActivity`unterstützt S3 DataNode und SqlDataNode als Eingabe und Ausgabe, und der Kopiervorgang wird normalerweise ausgeführt record-by-record. `CopyActivity`Stellt jedoch eine leistungsstarke Kopie von Amazon S3 zu Amazon S3 bereit, wenn alle folgenden Bedingungen erfüllt sind:

- Die Eingabe und Ausgabe sind S3 DataNodes
- Das Feld `dataFormat` ist für Ein- und Ausgabe dasselbe.

Wenn Sie die komprimierten Daten als Eingabe verwenden und dies nicht über as Feld `compression` auf den S3-Datenknoten angeben, kann `CopyActivity` möglicherweise fehlschlagen. In diesem Fall erkennt `CopyActivity` das Ende Datensatzzeichens nicht ordnungsgemäß und der Vorgang schlägt fehl. `CopyActivity`Unterstützt außerdem das Kopieren von einem Verzeichnis in ein anderes Verzeichnis und das Kopieren einer Datei in ein Verzeichnis. Das record-by-record Kopieren erfolgt jedoch, wenn ein Verzeichnis in eine Datei kopiert wird. Schließlich unterstützt `CopyActivity` es nicht das Kopieren mehrteiliger Amazon S3 S3-Dateien.

Bei `CopyActivity` gibt es bestimmte Einschränkungen der CSV-Unterstützung. Wenn Sie ein S3 DataNode als Eingabe für verwenden`CopyActivity`, können Sie nur eine Unix/Linux-Variante des CSV-Datendateiformats für die Amazon S3 S3-Eingabe- und Ausgabefelder verwenden. Die Unix-/Linux-Variante setzt Folgendes voraus:

- Das Trennzeichen muss ein Komma (,) sein.
- Die Datensätze werden nicht in Anführungszeichen gesetzt.
- Das Standard-Escape-Zeichen ist ASCII-Wert 92 (Backslash).
- Das Datensatzende-Identifier ist ASCII-Wert 10 (oder "\n").

Windows-basierte Systeme verwenden in der Regel eine andere end-of-record Zeichenfolge: Zeilenumbruch und Zeilenvorschub zusammen (ASCII-Wert 13 und ASCII-Wert 10). Sie müssen diesen Unterschied mit einem zusätzlichen Mechanismus ausgleichen, z. B. einem Skript zum

Ändern der Eingabedaten vor dem Kopieren, um sicherzustellen, dass `CopyActivity` das Datensatzende korrekt erkennt. Andernfalls schlägt `CopyActivity` wiederholt fehl.

Wenn Sie mit `CopyActivity` einen Exportvorgang von einem PostgreSQL-RDS-Objekt in das TSV-Datenformat durchführen, ist das Standard-NULL-Zeichen \n.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Dieses Objekt verweist auf drei andere Objekte, die Sie in derselben Pipeline-Definitionsdatei definieren. `CopyPeriod` ist ein `Schedule`-Objekt und `InputData` und `OutputData` sind Datenknotenobjekte.

```
{
  "id" : "S3ToS3Copy",
  "type" : "CopyActivity",
  "schedule" : { "ref" : "CopyPeriod" },
  "input" : { "ref" : "InputData" },
  "output" : { "ref" : "OutputData" },
  "runsOn" : { "ref" : "MyEc2Resource" }
}
```

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispiels weise „schedule“: {"ref“: "„} angeben. DefaultSc hedule In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne	Referenzobjekt, z. B. „schedule“: {"ref“:“ mySchedule Id „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	innerhalb des Hauptplans), können Benutzer ein übergeordnetes Objekt mit Zeitpläne nreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
input	Die Eingangsdatenquelle.	Referenzobjekt, z. B. „input“: {"ref“:“ myDataNode Id "}
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. ondemand	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}

Optionale Felder	Beschreibung	Slot-Typ
output	Die Eingangsdatenquelle.	Referenzobjekt, z. B. „output“: {"ref“:“ myDataNode Id "}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ „} myPrecond itionId
reportProgressTime out	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Zuständigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromI nstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@ healthSta tusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
@lastDeactivatedTi me	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCom pletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Weitere Informationen finden Sie unter:

- ShellCommandActivity
- EmrActivity
- Exportieren Sie MySQL-Daten nach Amazon S3 mit AWS Data Pipeline

EmrActivity

Führt einen EMR-Cluster.

AWS Data Pipelineverwendet ein anderes Format für Schritte als Amazon EMR; AWS Data Pipeline verwendet beispielsweise kommagetrennte Argumente nach dem JAR-Namen im `EmrActivity` Schrittfeld. Das folgende Beispiel zeigt einen für Amazon EMR formatierten Schritt, gefolgt von seinem AWS Data Pipeline Äquivalent:

```
s3://example-bucket/MyWork.jar arg1 arg2 arg3
```

```
"s3://example-bucket/MyWork.jar,arg1,arg2,arg3"
```

Beispiele

Es folgt ein Beispiel für diesen Objekttyp. In diesem Beispiel werden ältere Versionen von Amazon EMR verwendet. Überprüfen Sie die Richtigkeit dieses Beispiels anhand der Version des Amazon EMR-Clusters, die Sie verwenden.

Dieses Objekt verweist auf drei andere Objekte, die Sie in derselben Pipeline-Definitionsdatei definieren. `MyEmrCluster` ist ein `EmrCluster`-Objekt und `MyS3Input` und `MyS3Output` sind `S3DataNode`-Objekte.

> **Note**
>
> In diesem Beispiel können Sie das Feld `step` mit der gewünschten Cluster-Zeichenfolge ersetzen. Hierbei kann es sich u. a. um ein Pig-Skript, ein Hadoop-Streaming-Cluster oder Ihre eigene benutzerdefinierte JAR-Datei mit ihren Parametern handeln.

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**
>
> Um in einem Schritt Argumente an eine Anwendung zu übergeben, müssen Sie die Region im Pfad des Skripts angeben, wie im folgenden Beispiel gezeigt: Darüber hinaus müssen Sie für die zu übergebenden Argumente möglicherweise ein Escape-Zeichen verwenden. Wenn Sie beispielsweise mit `script-runner.jar` ein Shell-Skript ausführen und Argumente an das Skript übergeben möchten, müssen Sie für die Kommas, die als Trennzeichen dienen, Escape-Zeichen verwenden. Der folgende Schritt-Slot veranschaulicht die entsprechende Vorgehensweise:
>
> ```
> "step" : "s3://eu-west-1.elasticmapreduce/libs/script-runner/script-
> runner.jar,s3://datapipeline/echo.sh,a\\\\,b\\\\,c"
> ```

Dieser Schritt verwendet `script-runner.jar`, um das Shell-Skript `echo.sh` auszuführen, und übergibt `a`, `b` und `c` als einzelne Argumente an das Skript. Die erste Escape-Zeichen wird vom resultierenden Argument entfernt, weshalb Sie möglicherweise erneut ein Escape-Zeichen verwenden müssen. Wenn Sie beispielsweise `File\.gz` als Argument in JSON verwendet haben, können Sie als Escape-Zeichen `File\\\\.gz` verwenden. Da das erste Escape-Zeichen jedoch verworfen wird, müssen Sie `File\\\\\\\\.gz` verwenden.

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Sie müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausführungsreihenfolge für dieses Objekt festzulegen. Sie können diese Anforderung erfüllen, indem Sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispielsweise `"schedule": {"ref": "DefaultSchedule"}` angeben. In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Sie ein übergeordnetes Objekt mit Zeitplänenreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	Referenzobjekt, zum Beispiel „schedule“: {"ref“:“ myScheduleId „}

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
runsOn	Der Amazon EMR-Cluster, auf dem dieser Job ausgeführt wird.	Referenzobjekt, zum Beispiel „runsOn“: {"ref“:“ myEmrClusterId "}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen `runs0n`-Wert angeben und `workerGroup` vorhanden ist, wird ignoriert.`workerGroup`	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt.	Referenzobjekt, zum Beispiel „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
input	Der Speicherort der Eingabedaten.	Referenzobjekt, zum Beispiel „input“:

Optionale Felder	Beschreibung	Slot-Typ
		{"ref“:“ myDataNode Id "}
lateAfterTimeout	Die verstrichene Zeit nach dem Start der Pipeline, innerhalb derer das Objekt abgeschlo ssen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Die maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, zum Beispiel „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, zum Beispiel "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, zum Beispiel „onSuccess“: {"ref“:“ myActionId „}
output	Der Speicherort der Ausgabedaten.	Referenzobjekt, zum Beispiel „output“: {"ref“:“ myDataNode Id "}

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Das übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die Amazon S3 S3-URI, z. B. 's3://Buc ketName/Prefix/ 'zum Hochladen von Protokoll en für die Pipeline.	String
postStepCommand	Shell-Skripts, die nach Abschluss aller Schritte ausgeführt werden. Wenn Sie mehrere Skripts angeben möchten (maximal 255), fügen Sie die entsprechende Anzahl von `postStepC ommand` -Feldern hinzu.	String
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, zum Beispiel „precondi tion“: {"ref“:“ „} myPreconditionId
preStepCommand	Shell-Skripts, die vor allen Schritten ausgeführ t werden. Wenn Sie mehrere Skripts angeben möchten (maximal 255), fügen Sie die entsprechende Anzahl von `preStepCo mmand` -Feldern hinzu.	String
reportProgressTime out	Das Timeout für aufeinanderfolgende Aufrufe von `reportProgress` durch Remote-Ar beit. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
resizeClusterBeforeWird ausgeführt	Ändern Sie die Größe des Clusters, bevor Sie diese Aktivität ausführen, um DynamoDB-Tabellen aufzunehmen, die als Eingaben oder Ausgaben angegeben sind. **Note** Wenn Ihre `EmrActivity` einen `DynamoDBDataNode` als Eingabe- oder Ausgabedatenknoten verwendet und Sie `resizeClusterBeforeRunning` auf `TRUE` festlegen, beginnt AWS Data Pipeline mit der Verwendung von `m3.xlarge` -Instance-Typen. Dadurch wird Ihre Auswahl an Instance-Typen mit `m3.xlarge` überschrieben, wodurch Ihre monatlichen Kosten ansteigen könnten.	Boolesch
resizeClusterMaxInstanzen	Ein Limit für die maximale Anzahl von Instances, die vom Resize-Algorithmus angefordert werden können.	Ganzzahl
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definitio n am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Werte sind `cron`, `ondemand` und `timeseries` . Die `timeseries` -Planung bedeutet, dass Instances am Ende jedes Intervalls geplant sind. Die `cron`-Planung bedeutet, dass Instances am Anfang jedes Intervalls geplant sind. Ein `ondemand`-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführen. Sie müssen die Pipeline nicht klonen oder neu erstellen, um sie erneut auszuführen. Wenn Sie einen `ondemand`-Zeitplan verwenden, muss er im Standardo bjekt angegeben werden und der einzige für die Objekte in der Pipeline angegeben e `scheduleType` sein. Um `ondemand`-Pipelines zu verwenden, rufen Sie einfach den `ActivatePipeline` -Vorgang für jeden nachfolgenden Lauf auf.	Aufzählung
Schritt	Einzelne oder mehrere vom Cluster auszuführ ende Schritte. Wenn Sie mehrere Schritte angeben möchten (maximal 255), fügen Sie die entsprechende Anzahl von step-Feld ern hinzu. Verwenden Sie durch Komma getrennte Argumente nach dem JAR-Namen, z. B. "`s3://example-bucket/MyWork. jar,arg1,arg2,arg3` ".	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, zum Beispiel "cascadeF ailedOn„: {" ref“:“ myRunnableObject Id "}
emrStepLog	Amazon EMR-Schrittprotokolle sind nur bei EMR-Aktivitätsversuchen verfügbar	String
errorId	Die `errorId`, wenn dieses Objekt fehlgesch lagen ist.	String
errorMessage	Die `errorMessage` , wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ Zeit healthStatusUpdated	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für das Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für das Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Laufzeitfelder	Beschreibung	Slot-Typ
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, zum Beispiel „WaitingO n“: {"ref“:“ myRunnabl eObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- ShellCommandActivity
- CopyActivity
- EmrCluster

HadoopActivity

Führt einen MapReduce Job auf einem Cluster aus. Bei dem Cluster kann es sich um einen EMR-Cluster handeln, der von AWS Data Pipeline oder einer anderen Ressource verwaltet wird, wenn Sie ihn verwenden TaskRunner. Verwenden Sie diese Option, HadoopActivity wenn Sie parallel arbeiten möchten. Auf diese Weise können Sie die Planungsressourcen des YARN-Frameworks oder des MapReduce Resource Negotiators in Hadoop 1 verwenden. Wenn Sie die Arbeit sequenziell mit der Amazon EMR Step-Aktion ausführen möchten, können Sie dies trotzdem verwenden. EmrActivity

Beispiele

HadoopActivity unter Verwendung eines EMR-Clusters, verwaltet von AWS Data Pipeline

Das folgende HadoopActivity Objekt verwendet eine EmrCluster Ressource, um ein Programm auszuführen:

```
 {
   "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”
 }
```

Hier ist das entsprechende *MyEmrCluster*, das die Warteschlangen FairScheduler und in YARN für Hadoop 2-basierte AMIs konfiguriert:

```
{
  "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”]
}
```

Dies ist der, den EmrCluster Sie zur Konfiguration FairScheduler in Hadoop 1 verwenden:

```
{
      "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"
          }
```

Die folgenden Konfigurationen CapacityScheduler für Hadoop EmrCluster 2-basierte AMIs:

```
{
      "id": "MyEmrCluster",
      "type": "EmrCluster",
      "hadoopSchedulerType": "PARALLEL_CAPACITY_SCHEDULING",
      "amiVersion": "3.7.0",
      "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
\\\\,high,-z,yarn.scheduler.capacity.root.high.capacity=40,-
z,yarn.scheduler.capacity.root.low.capacity=60"
    }
```

HadoopActivity mit einem vorhandenen EMR-Cluster

In diesem Beispiel verwenden Sie workergroups und a, TaskRunner um ein Programm auf einem vorhandenen EMR-Cluster auszuführen. Die folgende Pipeline-Definition dient dazu: HadoopActivity

- Führen Sie ein MapReduce Programm nur auf *myWorkerGroup*Ressourcen aus. Weitere Informationen zu Worker-Gruppen finden Sie unter Ausführen von Arbeiten an vorhandenen Ressourcen mithilfe von Task Runner.
- Führen Sie eine preActivityTask Config und eine postActivityTask Config aus

```
{
```

```
  "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"
      }
    }
  ]
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
jarUri	Speicherort einer JAR in Amazon S3 oder im lokalen Dateisystem des Clusters, mit dem ausgeführt werden soll HadoopActivity.	String

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg	Referenzobjekt, z. B. „schedule“: {"ref“:“ mySchedule Id „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, z. B. indem sie „schedule “: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanv erweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Benutzer ein übergeord netes Objekt mit Zeitplänenreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Erforderliche Gruppe (mindestens eine der folgenden ist erforderl ich)	Beschreibung	Slot-Typ
runsOn	EMR-Cluster, auf dem dieser Auftrag ausgeführ t wird.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myEmrCluster Id "}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.	String

Optionale Felder	Beschreibung	Slot-Typ
argument	Argumente, die an die JAR-Dateien übergeben werden.	String
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
hadoopQueue	Der Name der Hadoop-Scheduler-Warteschlange, an die die Aktivität übergeben wird.	String
input	Speicherort der Eingabedaten.	Referenzobjekt, z. B. „input“: {"ref“:“ myDataNode Id "}
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
mainClass	Die Hauptklasse der JAR, mit der Sie die Ausführung ausführen HadoopActivity.	String

Optionale Felder	Beschreibung	Slot-Typ
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
output	Speicherort der Ausgabedaten.	Referenzobjekt, z. B. „output“: {"ref“:“ myDataNode Id "}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
postActivityTaskConfig	Post-Activity-Konfigurationsskript, das ausgeführt werden soll. Dieses besteht aus einer URI des Shell-Skripts in Amazon S3 und einer Liste von Argumenten.	Referenzobjekt, z. B. "postActivityTaskConfig“: {"ref“:“ myShellScriptConfigId „}

Optionale Felder	Beschreibung	Slot-Typ
preActivityTaskConfig	Pre-Activity-Konfigurationsskript, das ausgeführt werden soll. Dieses besteht aus einer URI des Shell-Skripts in Amazon S3 und einer Liste von Argumenten.	Referenzobjekt, z. B. "preActivityTaskConfig“: {"ref“:“ myShellScriptConfigId „}
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „Vorbedingung“: {"ref“:“ myPreconditionId „}
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromI nstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@ healthSta tusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
@lastDeactivatedTi me	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCom pletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- ShellCommandActivity
- CopyActivity
- EmrCluster

HiveActivity

Führt eine Hive-Abfrage auf einem EMR-Cluster aus. `HiveActivity`erleichtert die Einrichtung einer Amazon EMR-Aktivität und erstellt automatisch Hive-Tabellen auf der Grundlage von Eingabedaten, die entweder von Amazon S3 oder Amazon RDS stammen. Sie müssen lediglich die HiveQL angeben, die auf den Quellendaten ausgeführt werden soll. AWS Data Pipeline erstellt automatisch Hive-Tabellen mit `${input1}`, `${input2}` und so weiter, basierend auf den Eingabefeldern im `HiveActivity`-Objekt.

Für Amazon S3 S3-Eingaben wird das `dataFormat` Feld verwendet, um die Hive-Spaltennamen zu erstellen.

Bei MySQL-Eingaben (Amazon RDS) werden die Spaltennamen für die SQL-Abfrage verwendet, um die Hive-Spaltennamen zu erstellen.

> **Note**
>
> Diese Aktivität verwendet den CSV-Serde von Hive.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Dieses Objekt verweist auf drei andere Objekte, die Sie in derselben Pipeline-Definitionsdatei definieren. `MySchedule` ist ein `Schedule`-Objekt und `MyS3Input` und `MyS3Output` sind Datenknotenobjekte.

```
{
  "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" }
}
```

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Sie müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausführungsreihenfolge für dieses Objekt festzulegen. Sie können diese Anforderung erfüllen, indem Sie explizit einen Zeitplan für das Objekt festlegen, indem Sie beispielsweise „schedule“: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Sie ein übergeordnetes Objekt mit Zeitplänenreferenz erstellen. Weitere Informationen zu optionalen Beispiel-	Referenzobjekt, z. B. „schedule“: {"ref“:“ myScheduleId „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
hiveScript	Das auszuführende Hive-Skript.	String
scriptUri	Der Speicherort des auszuführenden Hive-Skripts (z. B. s3://scriptLocation).	String

Erforderliche Gruppe	Beschreibung	Slot-Typ
runsOn	Der EMR-Cluster, auf dem diese `HiveActivity` ausgeführt wird	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myEmrCluster Id "}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen `runsOn`-Wert angeben und `workerGroup` vorhanden ist, wird ignoriert.`workerGroup`	String
input	Die Eingangsdatenquelle.	Referenzobjekt, z. B. „input“: {"ref“:“ myDataNode Id "}
output	Die Eingangsdatenquelle.	Referenzobjekt, z. B. „output“: {"ref“:“ myDataNode Id "}

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
hadoopQueue	Der Name der Hadoop-Scheduler-Warteschlange, in der der Auftrag übermittelt wird.	String
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Die maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}

Optionale Felder	Beschreibung	Slot-Typ
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die S3-URI (z. B. 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
postActivityTaskConfig	Post-Activity-Konfigurationsskript, das ausgeführt werden soll. Dieses besteht aus einer URI des Shell-Skripts in Amazon S3 und einer Liste von Argumenten.	Referenzobjekt, z. B. "postActivityTaskConfig“: {"ref“:“ myShellScriptConfigId „}
preActivityTaskConfig	Pre-Activity-Konfigurationsskript, das ausgeführt werden soll. Dieses besteht aus einer URI des Shell-Skripts in Amazon S3 und einer Liste von Argumenten.	Referenzobjekt, z. B. "preActivityTaskConfig“: {"ref“:“ myShellScriptConfigId „}
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „Vorbedingung“: {"ref“:“ myPreconditionId „}

Optionale Felder	Beschreibung	Slot-Typ
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in `reportProgress` . Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegeben en Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
resizeClusterBeforeWird ausgeführt	Ändern Sie die Größe des Clusters, bevor Sie diese Aktivität ausführen, um DynamoDB-Datenknoten aufzunehmen, die als Eingaben oder Ausgaben angegeben sind. **Note** Wenn Ihre Aktivität einen `DynamoDBDataNode` als Eingabe- oder Ausgabedatenknoten verwendet und Sie `resizeClusterBeforeRunning` auf `TRUE` festlegen , beginnt AWS Data Pipeline, `m3.xlarge` -Instance-Typen zu verwenden. Dadurch wird Ihre Auswahl an Instance-Typen mit `m3.xlarge` überschrieben, wodurch Ihre monatlichen Kosten ansteigen könnten.	Boolesch
resizeClusterMaxInstanzen	Ein Limit für die maximale Anzahl von Instances, die vom Resize-Algorithmus angefordert werden können.	Ganzzahl
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
scriptVariable	Gibt Skriptvariablen an, die Amazon EMR bei der Ausführung eines Skripts an Hive weitergibt. Im folgenden Beispiel etwa würden Skriptvariablen eine SAMPLE- und FILTER_DATE-Variable an Hive übergeben: `SAMPLE=s3://elasticmapreduce/samples/hive-ads` und `FILTER_DATE=#{format(@scheduledStartTime,'YYYY-MM-dd')}%` . Dieses Feld akzeptiert mehrere Werte und funktioniert sowohl mit `script`- als auch mit `scriptUri` -Feldern. Darüber hinaus funktioniert `scriptVariable` unabhängig davon, ob "stage" auf `true` oder `false` festgelegt ist. Dieses Feld ist besonders nützlich, um mithilfe von AWS Data Pipeline-Ausdrücken und -Funktionen dynamische Werte an Hive zu senden.	String
stage	Legt fest, ob vor oder nach dem Ausführen des Skripts Staging aktiviert wird. Ist mit Hive 11 unzulässig. Verwenden Sie daher eine Amazon EMR-AMI in der Version 3.2.0 oder höher.	Boolesch

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ Id "} myRunnableObject
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	Amazon EMR-Schrittprotokolle sind nur bei EMR-Aktivitätsversuchen verfügbar.	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromI nstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@ Zeit healthSta tusUpdated	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
@lastDeactivatedTi me	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCom pletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für ein Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für ein Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „WaitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- ShellCommandActivity
- EmrActivity

HiveCopyActivity

Führt eine Hive-Abfrage auf einem EMR-Cluster aus. `HiveCopyActivity`erleichtert das Kopieren von Daten zwischen DynamoDB-Tabellen. `HiveCopyActivity`akzeptiert eine HiveQL-Anweisung zum Filtern von Eingabedaten aus DynamoDB auf Spalten- und Zeilenebene.

Beispiel

Das folgende Beispiel zeigt, wie Sie mit `HiveCopyActivity` und `DynamoDBExportDataFormat` Daten von einem `DynamoDBDataNode` auf einen anderen kopieren können, während gleichzeitig Daten basierend auf einem Zeitstempel gefiltert werden.

```
{
  "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"
    }
  ]
}
```

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispiels weise „schedule“: {"ref“: "„} angeben. DefaultSc hedule In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Benutzer ein übergeordnetes Objekt mit Zeitpläne nreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	Referenzobjekt, z. B. „schedule“: {"ref“:“ mySchedule Id „}

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
runsOn	Geben Sie den Cluster an, auf dem ausgeführt werden soll.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen `runs0n`-Wert angeben und `workerGroup` vorhanden ist, wird ignoriert.`workerGroup`	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Der zuletzt gemeldete Status von der Remote-Aktivität.	String
attemptTimeout	Das Timeout für die Fertigstellung der Remote-Arbeit. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dependsOn	Gibt die Abhängigkeit von einem anderen ausführbaren Objekt an.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
filterSql	Ein Hive-SQL-Anweisungsfragment, das eine Teilmenge der zu kopierenden DynamoDB- oder Amazon S3 S3-Daten filtert. Der Filter	String

Optionale Felder	Beschreibung	Slot-Typ
	darf nur Prädikate enthalten und nicht mit einer `WHERE`-Klausel beginnen, da AWS Data Pipeline diese automatisch hinzufügt.	
input	Die Eingangsdatenquelle. Dies muss ein `S3DataNode` oder `DynamoDBDataNode` sein. Wenn Sie `DynamoDBNode` verwenden , geben Sie ein `DynamoDBExportDataFormat` an.	Referenzobjekt, z. B. „input“: {"ref“:“ Id "} myDataNode
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Die maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}

Optionale Felder	Beschreibung	Slot-Typ
output	Die Eingangsdatenquelle. Wenn die Eingabe `S3DataNode` ist, muss diese auf `DynamoDBDataNode` festgelegt sein. Andernfalls kann dies `S3DataNode` oder `DynamoDBDataNode` sein. Wenn Sie `DynamoDBNode` verwenden, geben Sie ein `DynamoDBExportDataFormat` an.	Referenzobjekt, z. B. „output“: {"ref“:“ myDataNode Id "}
übergeordneter	Das übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die Amazon S3 S3-URI, z. B. `'s3://BucketName/Key/'` für das Hochladen von Protokollen für die Pipeline.	String
postActivityTaskConfig	Das Post-Activity-Konfigurationsskript, das ausgeführt werden soll. Dieses besteht aus einer URI des Shell-Skripts in Amazon S3 und einer Liste von Argumenten.	Referenzobjekt, z. B. "postActivityTaskConfig“: {"ref“:“ myShellScriptConfigId „}
preActivityTaskConfig	Das Pre-Activity-Konfigurationsskript, das ausgeführt werden soll. Dieses besteht aus einer URI des Shell-Skripts in Amazon S3 und einer Liste von Argumenten.	Referenzobjekt, z. B. "preActivityTaskConfig“: {"ref“:“ myShellScriptConfigId „}
precondition	Definiert optional eine Vorbedingung. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „Vorbedingung“: {"ref“:“ myPreconditionId „}

Optionale Felder	Beschreibung	Slot-Typ
reportProgressTimeout	Das Timeout für aufeinanderfolgende Aufrufe von `reportProgress` durch Remote-Arbeit. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
resizeClusterBeforeWird ausgeführt	Ändern Sie die Größe des Clusters, bevor Sie diese Aktivität ausführen, um DynamoDB-Datenknoten aufzunehmen, die als Eingaben oder Ausgaben angegeben sind. **Note** Wenn Ihre Aktivität einen `DynamoDBDataNode` als Eingabe- oder Ausgabedatenknoten verwendet und Sie `resizeClusterBeforeRunning` auf `TRUE` festlegen, beginnt AWS Data Pipeline, `m3.xlarge` -Instance-Typen zu verwenden. Dadurch wird Ihre Auswahl an Instance-Typen mit `m3.xlarge` überschrieben, wodurch Ihre monatlichen Kosten ansteigen könnten.	Boolesch
resizeClusterMaxInstanzen	Ein Limit für die maximale Anzahl von Instances, die vom Resize-Algorithmus angefordert werden können.	Ganzzahl
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	Amazon EMR-Schrittprotokolle sind nur bei EMR-Aktivitätsversuchen verfügbar.	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromI nstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@ Zeit healthSta tusUpdated	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
@lastDeactivatedTi me	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCom pletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Sphäre eines Objekts bezeichnet seine Position im Lebenszyklus: Komponent enobjekte ergeben Instance-Objekte, die ein Versuchsobjekt ausführen.	String

Weitere Informationen finden Sie unter:

- ShellCommandActivity
- EmrActivity

PigActivity

PigActivity bietet native Unterstützung für Pig-Skripte, AWS Data Pipeline ohne dass die Verwendung von `ShellCommandActivity` oder erforderlich ist`EmrActivity`. PigActivity Unterstützt außerdem Daten-Staging. Wenn das Stage-Feld auf „true“ festgelegt wurde, arrangiert AWS Data Pipeline die Eingabedaten ohne zusätzlichen Code des Benutzers als Schema in Pig.

Beispiel

Im folgenden Pipeline-Beispiel wird gezeigt, wie `PigActivity` verwendet wird. Die Beispiel-Pipeline führt die folgenden Schritte aus:

- MyPigActivity1 lädt Daten aus Amazon S3 und führt ein Pig-Skript aus, das einige Datenspalten auswählt und sie auf Amazon S3 hochlädt.
- MyPigActivity2 lädt die erste Ausgabe, wählt einige Spalten und drei Datenzeilen aus und lädt sie als zweite Ausgabe auf Amazon S3 hoch.
- MyPigActivity3 lädt die zweiten Ausgabedaten, fügt zwei Datenzeilen und nur die Spalte mit dem Namen „Fifth“ in Amazon RDS ein.

- MyPigActivity4 lädt Amazon RDS-Daten, wählt die erste Datenzeile aus und lädt sie auf Amazon S3 hoch.

```
{
  "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"
    }
  ]
}
```

Der Inhalt von `pigTestScript.q` ist wie folgt:

```
B = LIMIT ${input1} $three; ${output1} = FOREACH B GENERATE $column1, $column2, Third,
 Fourth, Fifth, Sixth, Seventh, Eighth;
```

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Benutzer müssen einen Zeitplanverweis auf ein anderes	Referenzobjekt, zum Beispiel „schedule“:

Objektaufruf-Felder	Beschreibung	Slot-Typ
	Objekt angeben, um die Abhängigkeitsausfü hrungsreihenfolge für dieses Objekt festzuleg en. Benutzer können diese Anforderung erfüllen, indem sie explizit einen Zeitplan für das Objekt festlegen, z. B. indem sie „schedule “: {"ref“: "DefaultSchedule„} angeben. In den meisten Fällen ist es besser, den Zeitplanv erweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitpläne innerhalb des Hauptplans), können Benutzer ein übergeord netes Objekt mit Zeitplänenreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	{"ref“:“ myScheduleId „}

Erforderliche Gruppe (mindestens eine der folgenden ist erforderl ich)	Beschreibung	Slot-Typ
script	Das auszuführende Pig-Skript.	String
scriptUri	Der Speicherort des auszuführenden Pig-Skrip ts (z. B. s3://scriptLocation).	String

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
runsOn	EMR-Cluster, auf dem das PigActivity läuft.	Referenzobjekt, zum Beispiel „runsOn“: {"ref“:“ myEmrClusterId "}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen `runs0n`-Wert angeben und `workerGroup` vorhanden ist, wird ignoriert.`workerGroup`	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Der zuletzt gemeldete Status von der Remote-Aktivität.	String
attemptTimeout	Das Timeout für die Fertigstellung der Remote-Arbeit. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dependsOn	Gibt die Abhängigkeit von einem anderen ausführbaren Objekt an.	Referenzobjekt, zum Beispiel „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
input	Die Eingangsdatenquelle.	Referenzobjekt, zum Beispiel „input“:

Optionale Felder	Beschreibung	Slot-Typ
		{"ref“:“ myDataNode Id "}
lateAfterTimeout	Die verstrichene Zeit nach dem Start der Pipeline, innerhalb derer das Objekt abgeschlo ssen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Die maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, zum Beispiel „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, zum Beispiel "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, zum Beispiel „onSuccess“: {"ref“:“ myActionId „}
output	Die Eingangsdatenquelle.	Referenzobjekt, zum Beispiel „output“: {"ref“:“ myDataNode Id "}

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“: {"ref“:“ myBaseObject Id "}
pipelineLogUri	Die Amazon S3 S3-URI (z. B. 's3://Buc ketName/Key/ ') zum Hochladen von Protokoll en für die Pipeline.	String
postActivityTaskCo nfig	Post-Activity-Konfigurationsskript, das ausgeführt werden soll. Dies besteht aus einer URI des Shell-Skripts in Amazon S33 und einer Liste von Argumenten.	Referenzobjekt, zum Beispiel "postActi vityTaskConfig“: {"ref“:“ myShellScript ConfigId „}
preActivityTaskConfig	Pre-Activity-Konfigurationsskript, das ausgeführ t werden soll. Dieses besteht aus einer URI des Shell-Skripts in Amazon S3 und einer Liste von Argumenten.	Referenzobjekt, zum Beispiel "preActiv ityTaskConfig“: {"ref“:“ myShellScript ConfigId „}
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, zum Beispiel „precondition“: {"ref“:“ myPreconditionId „}
reportProgressTime out	Das Timeout für aufeinanderfolgende Aufrufe von `reportProgress` durch Remote-Ar beit. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
resizeClusterBeforeWird ausgeführt	Ändern Sie die Größe des Clusters, bevor Sie diese Aktivität ausführen, um DynamoDB-Datenknoten aufzunehmen, die als Eingaben oder Ausgaben angegeben sind. **Note** Wenn Ihre Aktivität einen `DynamoDBDataNode` als Eingabe- oder Ausgabedatenknoten verwendet und Sie `resizeClusterBeforeRunning` auf `TRUE` festlegen, beginnt AWS Data Pipeline, `m3.xlarge` -Instance-Typen zu verwenden. Dadurch wird Ihre Auswahl an Instance-Typen mit `m3.xlarge` überschrieben, wodurch Ihre monatlichen Kosten ansteigen könnten.	Boolesch
resizeClusterMaxInstanzen	Ein Limit für die maximale Anzahl von Instances, die vom Resize-Algorithmus angefordert werden können.	Ganzzahl
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Zeitreihe nstilplanung bedeutet, dass Instances am Ende jedes Intervalls geplant werden und Cron-Stil-Planung bedeutet, dass Instances zu Beginn jedes Intervalls geplant werden. Ein On-Demand-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene scheduleType sein. Um On-Demand-Pipelines zu verwenden, rufen Sie den ActivatePipeline Vorgang einfach für jeden nachfolgenden Lauf auf. Die Werte sind: cron, ondemand und timeseries.	Aufzählung
scriptVariable	Die Argumente, die an das Pig-Skript übergeben werden sollen. Sie können scriptVar iable mit script oder scriptUri verwenden.	String
stage	Legt fest, ob Staging aktiviert ist, und gewährt Ihrem Pig-Skript den Zugriff auf Staging-Daten-Tabellen, z. B. ${INPUT1} und ${OUTPUT1}.	Boolesch

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, zum Beispiel „ActiveIn

Laufzeitfelder	Beschreibung	Slot-Typ
		stances“: {"ref“:“ myRunnableObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, zum Beispiel "cascadeF ailedOn„: {" ref“:“ myRunnableObject Id "}
emrStepLog	Amazon EMR-Schrittprotokolle sind nur bei EMR-Aktivitätsversuchen verfügbar.	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ Zeit healthStatusUpdated	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für das Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für das Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Laufzeitfelder	Beschreibung	Slot-Typ
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, zum Beispiel „WaitingO n“: {"ref“:“ myRunnabl eObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- ShellCommandActivity
- EmrActivity

RedshiftCopyActivity

Kopiert Daten von DynamoDB oder Amazon S3 nach Amazon Redshift. Sie können Daten in eine neue Tabelle laden oder Daten in einer vorhandenen Tabelle einfach zusammenführen.

Hier finden Sie eine Übersicht über einen Anwendungsfall, in dem `RedshiftCopyActivity` verwendet wird:

1. Verwenden Sie zunächstAWS Data Pipeline, um Ihre Daten in Amazon S3 bereitzustellen.
2. Wird verwendet`RedshiftCopyActivity`, um die Daten von Amazon RDS und Amazon EMR nach Amazon Redshift zu verschieben.

Auf diese Weise können Sie Ihre Daten in Amazon Redshift laden, wo Sie sie analysieren können.

3. Wird verwendetSqlActivity, um SQL-Abfragen für die Daten durchzuführen, die Sie in Amazon Redshift geladen haben.

Darüber hinaus unterstützt `RedshiftCopyActivity` Ihre Arbeit mit einem `S3DataNode`, weil es eine Manifestdatei unterstützt. Weitere Informationen finden Sie unter S3 DataNode.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

Um die Formatkonvertierung sicherzustellen, verwendet dieses Beispiel EMPTYASNULL und IGNOREBLANKLINES, spezielle Konvertierungsparameter in `commandOptions`. Weitere Informationen finden Sie unter Datenkonvertierungsparameter im Amazon Redshift Database Developer Guide.

```
{
  "id" : "S3ToRedshiftCopyActivity",
  "type" : "RedshiftCopyActivity",
  "input" : { "ref": "MyS3DataNode" },
  "output" : { "ref": "MyRedshiftDataNode" },
  "insertMode" : "KEEP_EXISTING",
  "schedule" : { "ref": "Hour" },
  "runsOn" : { "ref": "MyEc2Resource" },
  "commandOptions": ["EMPTYASNULL", "IGNOREBLANKLINES"]
}
```

Die folgende Pipeline-Beispieldefinition zeigt eine Aktivität, die den Einfügemodus `APPEND` nutzt:

```
{
  "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"
      }
    }
  ]
}
```

Der APPEND-Vorgang fügt Elemente zu einer Tabelle hinzu, unabhängig von Primär- oder Sortierschlüsseln. Bei der folgenden Tabelle können Sie beispielsweise einen Datensatz mit demselben ID- und Benutzer-Wert anfügen.

```
ID(PK)     USER
1          aaa
```

```
2           bbb
```

Sie können einen Datensatz mit demselben ID- und Benutzer-Wert anfügen:

```
ID(PK)     USER
1          aaa
2          bbb
1          aaa
```

> **Note**
>
> Wenn ein APPEND-Vorgang unterbrochen und wieder aufgenommen wird, ist es möglich, dass die entstandene Wiederausführungs-Pipeline von Anfang an Anfügungen vornimmt. Dies kann zu weiteren Duplizierungen führen. Sie sollten dieses Verhalten kennen, besonders, wenn Sie Logik verwenden, die die Anzahl an Zeilen zählt.

Ein Tutorial finden Sie unter Kopieren Sie Daten nach Amazon Redshift mit AWS Data Pipeline.

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
insertMode	Legt fest, wie AWS Data Pipeline die bereits in der Zieltabelle enthaltenen Daten verarbeitet, die sich mit Zeilen in den zu ladenden Daten überschneiden. Gültige Werte sind: KEEP_EXISTING , OVERWRITE_EXISTING , TRUNCATE und APPEND. KEEP_EXISTING fügt der Tabelle neue Zeilen hinzu und lässt die vorhandenen Zeilen unverändert. KEEP_EXISTING und OVERWRITE_EXISTING verwenden den Primärsch	Aufzählung

Pflichtfelder	Beschreibung	Slot-Typ
	lüssel, Sortier- und Verteilschlüssel, um zu identifizieren, welche eingehende Zeilen mit vorhandenen Zeilen übereinstimmen. Weitere Informationen finden Sie unter Aktualisieren und Einfügen neuer Daten im Amazon Redshift Database Developer Guide. TRUNCATE löscht alle Daten in der Zieltabelle, bevor die neuen Daten hinzugefügt werden. APPEND fügt alle Datensätze am Ende der Redshift-Tabelle an. APPEND setzt keinen Primär-, Verteilungs- oder Sortierschlüssel voraus. Es können also Zeilen hinzugefügt werden, bei denen es sich um potenzielle Duplikate handelt.	

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Sie müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigk eitsausführungsreihenfolge für dieses Objekt festzulegen. In den meisten Fällen empfehlen wir, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Sie können beispielsweise einen Zeitplan explizit für das Objekt festlegen , indem Sie `"schedule": {"ref": "DefaultSchedule"}` angeben.	Referenzobjekt, wie z. B.: `"schedule":{"ref": "mySchedu leId"}`

Objektaufruf-Felder	Beschreibung	Slot-Typ
	Wenn der Hauptplan in Ihrer Pipeline verschachtelte Zeitpläne enthält, erstellen Sie ein übergeordnetes Objekt mit Zeitplanreferenz. Weitere Informationen zu optionalen Zeitplank onfigurationen finden Sie unter Zeitplan.	

Erforderliche Gruppe (mindestens eine der folgenden ist erforderl ich)	Beschreibung	Slot-Typ
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen `runsOn`-Wert angeben und `workerGroup` vorhanden ist, wird workerGroup ignoriert.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Ak tivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlo ssen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
commandOptions	Verwendet Parameter, die während des COPY Vorgangs an den Amazon Redshift Redshift-Datenknoten übergeben werden. Informationen zu Parametern finden Sie unter COPY im Amazon Redshift Database Developer Guide. Wenn COPY die Tabelle lädt, versucht der Befehl implizit, die Zeichenfolgen in den Quelldaten in den Datentyp der Zielspalte zu konvertieren. Zusätzlich zu den Standard-Datenkonvertierungen, die automatisch stattfinden, wenn Fehler erhalten oder andere Konvertierungen benötigen, können Sie zusätzliche Umrechnungsparameter angeben. Weitere Informationen finden Sie unter Datenkonvertierungsparameter im Amazon Redshift Database Developer Guide. Wenn dem Eingabe- oder Ausgabedatenknoten ein Datenformat zugeordnet ist, werden die angegebenen Parameter ignoriert. Da beim Kopieren die Daten zunächst mit dem Befehl COPY in eine Staging-Tabelle eingefügt und danach mit dem Befehl INSERT von der Staging- in die Zieltabelle kopiert werden, können einige COPY-Parameter nicht verwendet werden (z. B. die Fähigkeit des COPY-Befehls, der das automatische Komprimieren der Tabelle aktiviert). Wenn die Tabelle komprimiert werden soll, fügen Sie der Anweisung CREATE TABLE Angaben zur Spaltencodierung hinzu. In einigen Fällen, in denen Daten aus dem Amazon Redshift-Cluster entladen und Dateien in Amazon S3 erstellt werden müssen, ist	String

Optionale Felder	Beschreibung	Slot-Typ
	das außerdem auf den `UNLOAD` Betrieb von Amazon Redshift `RedshiftCopyActivity` angewiesen. Zur Verbesserung der Leistung beim Kopieren und Entladen geben Sie den `PARALLEL OFF`-Parameter aus dem `UNLOAD` Befehl an. Informationen zu Parametern finden Sie unter UNLOAD im Amazon Redshift Database Developer Guide.	
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt.	Referenzobjekt: `"dependsOn": {"ref":"myActivityId"}`
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
input	Der Eingabedatenknoten. Die Datenquelle kann Amazon S3, DynamoDB oder Amazon Redshift sein.	Referenzobjekt: `"input":{ "ref":"myDataNodeId"}`
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl

Optionale Felder	Beschreibung	Slot-Typ
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt: `"onFail": {"ref":"myActionId"}`
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt: `"onLateAction": {"ref":"myActionId"}`
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt: `"onSuccess": {"ref":"myActionId"}`
output	Der Ausgabedatenknoten. Der Ausgabespeicherort kann Amazon S3 oder Amazon Redshift sein.	Referenzobjekt: `"output": {"ref":"myDataNodeId"}`
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt: `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	Die S3-URI (z. B. 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt: `"precondition": {"ref":"myPreconditionId"}`

Optionale Felder	Beschreibung	Slot-Typ
Warteschlange	Entspricht der `query_group` Einstellung in Amazon Redshift, mit der Sie gleichzeitige Aktivitäten anhand ihrer Platzierung in Warteschlangen zuweisen und priorisieren können. In Amazon Redshift sind bis zu 15 gleichzeitige Verbindungen möglich. Weitere Informationen finden Sie unter Zuweisen von Abfragen zu Warteschlangen im Amazon RDS Database Developer Guide.	String
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in `reportProgress` . Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dieser Option können Sie angeben, ob der Plan für die Objekte in Ihrer Pipeline vorgesehen ist. Werte sind `cron`, `ondemand` und `timeseries` . Die `timeseries` Planung bedeutet, dass Instances am Ende jedes Intervalls geplant sind. Die `Cron` Planung bedeutet, dass Instances am Anfang jedes Intervalls geplant sind. Ein `ondemand`-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Um `ondemand`-Pipelines zu verwenden, rufen Sie einfach den `ActivatePipeline` - Vorgang für jeden nachfolgenden Lauf auf. Wenn Sie einen `ondemand`-Zeitplan verwenden , müssen Sie ihn im Standardobjekt angeben, und er muss der einzige für die Objekte in der Pipeline angegebene `scheduleType` sein.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
transformSql	Der zum Transformieren der Eingabedaten verwendete `SQL SELECT`-Ausdruck. Führen Sie den Ausdruck `transformSql` in der Tabelle mit dem Namen `staging` aus. Wenn Sie Daten aus DynamoDB oder Amazon S3 kopieren, AWS Data Pipeline erstellt eine Tabelle namens „Staging“ und lädt zunächst Daten hinein. Die Daten dieser Tabelle werden zum Aktualisieren der Zieltabelle verwendet. Das Ausgabe-Schema von `transform Sql` muss mit dem Schema der endgültigen Zieltabelle übereinstimmen. Wenn Sie die Option `transformSql` angeben, wird von der angegebenen SQL-Anweisung eine zweite Staging-Tabelle erstellt. Die Daten dieser zweiten Staging-Tabelle werden anschließend in die endgültige Zieltabel le übernommen.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt: `"activeIn stances": {"ref":"m yRunnable ObjectId"}`
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt: `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktivitätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ Zeit healthStatusUpdated	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt: `"waitingOn": {"ref":"myRunnableObjectId"}`

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Sphäre eines Objekts. Gibt seine Position im Lebenszyklus an. Beispielsweise ergeben Komponentenobjekte Instance-Objekte, die Versuchsobjekte ausführen.	String

ShellCommandActivity

Führt einen Befehl oder ein Skript aus. Mit `ShellCommandActivity` können Sie Zeitreihen oder Cron-ähnliche geplante Aufgaben ausführen.

Wenn das `stage` Feld auf true gesetzt ist und mit einem verwendet wird`S3DataNode`, `ShellCommandActivity` unterstützt es das Konzept der Datenbereitstellung, was bedeutet, dass Sie Daten von Amazon S3 an einen Staging-Speicherort wie Amazon EC2 oder Ihre lokale Umgebung verschieben können, die Daten mithilfe von Skripten und dem `ShellCommandActivity` bearbeiten und sie zurück zu Amazon S3 verschieben können.

Wenn in diesem Fall Ihr Shell-Befehl mit einem Eingabe-`S3DataNode` verbunden ist, werden Ihre Shell-Skripts mit `${INPUT1_STAGING_DIR}`, `${INPUT2_STAGING_DIR}` und anderen Feldern ausgeführt, die auf die `ShellCommandActivity`-Eingabefelder verweisen.

In ähnlicher Weise kann die Ausgabe des Shell-Befehls in einem Ausgabeverzeichnis bereitgestellt werden, um automatisch an Amazon S3 weitergeleitet zu werden, auf das mit, verwiesen wird `${OUTPUT1_STAGING_DIR}${OUTPUT2_STAGING_DIR}`, usw.

Diese Ausdrücke können als Befehlszeilenargumente zum Shell-Befehl weitergeleitet werden, sodass Sie sie für Datentransformationslogik verwenden können.

`ShellCommandActivity` gibt Linux-ähnliche Fehlercodes und Zeichenfolgen aus. Wenn `ShellCommandActivity` fehlschlägt, ist der angezeigte `error` ein Wert ungleich Null.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

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

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines `schedule`-Intervalls aufgerufen. Um die Abhängigkeitsausführungsreihenfolge für dieses Objekt festzulegen, geben Sie eine `schedule`-Referenz auf ein anderes Objekt an. Um diese Anforderung zu erfüllen, setzen Sie explizit einen `schedule` auf das Objekt, z. B. mit `"schedule": {"ref": "DefaultSchedule"}` . In den meisten Fällen ist es besser, die `schedule`-Referenz auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Besteht die Pipeline aus einem Baum mit Zeitplänen (Zeitpläne innerhalb des Hauptplans), erstellen Sie ein übergeordnetes Objekt, das eine Zeitplanr eferenz besitzt. Um die Last zu verteilen, erstellt AWS Data Pipeline physische Objekte etwas vor dem Zeitplan, führt sie jedoch gemäß dem Zeitplan aus.	Referenzobjekt, z. B. „schedule“: {"ref“:“ „} myScheduleId

Objektaufruf-Felder	Beschreibung	Slot-Typ
	Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
command	Den auszuführenden Befehl. Verwenden Sie $, um auf Positionsparameter zu verweisen , und geben Sie mit `scriptArgument` die Parameter für den Befehl an. Dieser Wert und alle zugehörigen Parameter müssen in der Umgebung funktionieren, in der Sie den Task-Runner ausführen.	String
scriptUri	Ein Amazon S3-URI-Pfad für eine Datei, die heruntergeladen und als Shell-Befehl ausgeführt werden soll. Geben Sie nur ein Feld `scriptUri` oder `command` an. `scriptUri` kann keine Parameter verwenden. Verwenden Sie stattdessen `command`.	String

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
runsOn	Die Rechenressource zur Ausführung der Aktivität oder des Befehls, z. B. eine Amazon EC2 EC2-Instance oder ein Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
workerGroup	Wird für Routing-Aufgaben verwendet. Wenn Sie einen `runs0n`-Wert angeben und `workerGroup` vorhanden ist, wird ignoriert `.workerGroup`	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Der zuletzt gemeldete Status von der Remote-Aktivität.	String
attemptTimeout	Das Timeout für die Fertigstellung der Remote-Arbeit. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dependsOn	Gibt eine Abhängigkeit von einem anderen ausführbaren Objekt an.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
input	Der Speicherort der Eingabedaten.	Referenzobjekt, z. B. „input“: {"ref“:“ myDataNode Id "}
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. ondemand	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Die maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
output	Der Speicherort der Ausgabedaten.	Referenzobjekt, z. B. „output“: {"ref“:“ myDataNode Id "}
übergeordneter	Das übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Optionale Felder	Beschreibung	Slot-Typ
pipelineLogUri	Die Amazon S3 S3-URI, z. B. `'s3://BucketName/Key/'` für das Hochladen von Protokollen für die Pipeline.	String
precondition	Definiert optional eine Vorbedingung. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ myPreconditionId „}
reportProgressTimeout	Das Timeout für aufeinanderfolgende Aufrufe von `reportProgress` durch Remote-Aktivitäten. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Gestattet Ihnen, anzugeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Die Werte sind: `cron`, `ondemand` und `timeseries` . `timeseries` bedeutet, dass Instances am Ende jedes Intervalls geplant sind. `Cron` bedeutet, dass Instances am Anfang jedes Intervalls geplant sind. `ondemand` bedeutet, Sie können eine Pipeline jeweils einmal pro Aktivierung auszuführen. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen `ondemand`-Zeitplan verwenden, geben Sie ihn im Standardobjekt als einzigen `scheduleType` für Objekte in der Pipeline an. Um `ondemand`-Pipelines zu verwenden, rufen Sie einfach den `ActivatePipeline` -Vorgang für jeden nachfolgenden Lauf auf.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
scriptArgument	Ein Zeichenfolgenarray im JSON-Format, das dem von dem Befehl angegebenen Befehl übergeben wird. Ist der Befehl beispielsweise `echo $1 $2`, geben Sie `scriptArgument` als `"param1", "param2"` an. Für mehrere Argumente und Parameter übergeben Sie das `scriptArgument` wie folgt: `"scriptArgument":"arg1","scriptArgument":"param1","scriptArgument":"arg2","scriptArgument":"param2"` Das `scriptArgument` ist `command`. Die Verwendung durch einen Fehler `scriptUri` verursacht.	String
stage	Legt fest, ob Staging aktiviert ist, und gewährt Ihren Shell-Befehlen den Zugriff auf Staging-Datenvariablen, z. B. `${INPUT1_STAGING_DIR}` und `${OUTPUT1_STAGING_DIR}` .	Boolesch
stderr	Der -Pfad, zu dem Systemfehlermeldungen vom Befehl umgeleitet werden. Wenn Sie das `runs0n` Feld verwenden, muss es sich um einen Amazon S3 S3-Pfad handeln, da die Ressource, auf der Ihre Aktivität ausgeführt wird, vorübergehend ist. Wenn Sie jedoch das Feld `workerGroup` angeben, ist ein lokaler Dateipfad zulässig.	String

Optionale Felder	Beschreibung	Slot-Typ
stdout	Der Amazon S3 S3-Pfad, der die umgeleite te Ausgabe des Befehls empfängt. Wenn Sie das `runs0n` Feld verwenden, muss es sich um einen Amazon S3 S3-Pfad handeln, da die Ressource, auf der Ihre Aktivität ausgeführt wird, vorübergehend ist. Wenn Sie jedoch das Feld `workerGroup` angeben, ist ein lokaler Dateipfad zulässig.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Die Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeIn stances“: {"ref“:“ Id "} myRunnableObject
@actualEndTime	Der Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Der Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Der `cancellationReason` , wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Die Beschreibung der Zuständigkeitskette, die den Objektausfall verursacht hat.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	Amazon EMR-Schrittprotokolle sind nur bei Amazon EMR-Aktivitätsversuchen verfügbar.	String

Laufzeitfelder	Beschreibung	Slot-Typ
errorId	Die `errorId`, wenn dieses Objekt fehlgesch lagen ist.	String
errorMessage	Die `errorMessage` , wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der das Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle sind bei Versuchen für Amazon EMR-basierte Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromI nstanceId	Die ID des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ Zeit healthSta tusUpdated	Der Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
@lastDeactivatedTi me	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCom pletedRun Zeit	Der Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Der Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@nextRunTime	Der Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Die geplante Endzeit für das Objekt.	DateTime
@scheduledStartTime	Die geplante Startzeit für das Objekt.	DateTime
@Status	Der Status des Objekts.	String
@Version	Die AWS Data Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Die Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Der Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Die Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Position eines Objekts im Lebenszyklus. Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- CopyActivity

- EmrActivity

SqlActivity

Führt eine SQL-Abfrage (Skript) auf einer Datenbank aus.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "id" : "MySqlActivity",
  "type" : "SqlActivity",
  "database" : { "ref": "MyDatabaseID" },
  "script" : "SQLQuery" | "scriptUri" : s3://scriptBucket/query.sql,
  "schedule" : { "ref": "MyScheduleID" },
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
Datenbank	Die Datenbank für die Ausführung des bereitgestellten SQL-Skripts.	Referenzobjekt, z. B. „database“: {"ref“:“ myDatabaseId „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Sie müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausführungsreihenfolge für dieses Objekt festzulegen. Sie können einen Zeitplan explizit für das Objekt festlegen, indem Sie `"schedule": {"ref": "DefaultSchedule"}` angeben.	Referenzobjekt, z. B. „schedule“: {"ref“:“ myScheduleId „}

Objektaufruf-Felder	Beschreibung	Slot-Typ
	In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum über in den Hauptplan verschachtelte Zeitplänen, können Benutzer ein übergeordnetes Objekt mit Zeitplanreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
script	Das auszuführende SQL-Skript. Sie müssen das Skript oder scriptUri angeben. Wenn das Skript in Amazon S3 gespeichert ist, wird das Skript nicht als Ausdruck ausgewertet. Die Angabe mehrerer Werte für scriptArgument ist hilfreich, wenn das Skript in Amazon S3 gespeichert ist.	String
scriptUri	Ein URI, der den Speicherort eines SQL-Skripts angibt, das in dieser Aktivität ausgeführt wird.	String

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
runsOn	Die Rechenressource zum Ausführen der Aktivität oder des Befehls. Beispiel: Amazon EC2 Instance oder Amazon EMR-Cluster.	Referenzobjekt, z. B. „runsOn“: {"ref“:“ myResourceId „}
workerGroup	Die Auftragnehmergruppe. Dies wird für Routing-Aufgaben verwendet. Wenn Sie einen `runs0n`-Wert angeben und `workerGroup` vorhanden ist, wird ignoriert.`workerGroup`	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
dependsOn	Angeben der Abhängigkeit von einem anderen ausführbaren Objekt.	Referenzobjekt, z. B. „dependSon“: {"ref“:“ myActivityId „}
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
input	Speicherort der Eingabedaten.	Referenzobjekt, z. B. „input“: {"ref“:“ myDataNode Id "}

Optionale Felder	Beschreibung	Slot-Typ
lateAfterTimeout	Der Zeitraum seit dem geplanten Start der Pipeline, in dem die Objektausführung starten muss.	Intervall
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt in dem durch 'lateAfterTimeout' angegebenen Zeitraum seit dem geplanten Start der Pipeline noch nicht geplant oder immer noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
output	Speicherort der Ausgabedaten. Dies ist nur nützlich, um innerhalb eines Skripts zu referenzieren (z. B.`#{output.tablename}`) und um die Ausgabetabelle zu erstellen, indem 'createTableSql' im Ausgabedatenknoten gesetzt wird. Die Ausgabe der SQL-Abfra ge wird nicht in den Ausgabedatenknoten geschrieben.	Referenzobjekt, z. B. „output“: {"ref“:“ myDataNode Id "}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Optionale Felder	Beschreibung	Slot-Typ
pipelineLogUri	Die S3-URI (wie 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String
precondition	Legen Sie optional eine Vorbedingung fest. Ein Datenknoten ist solange nicht als "BEREIT" markiert, bis alle Vorbedingungen erfüllt sind.	Referenzobjekt, z. B. „precondition“: {"ref“:“ „} myPreconditionId
Warteschlange	[Nur Amazon Redshift] Entspricht der Einstellung query_group in Amazon Redshift, mit der Sie gleichzeitig auszuführende Aktivitäten anhand ihrer Platzierung in Warteschlangen zuweisen und priorisieren können. In Amazon Redshift sind bis zu 15 gleichzeitige Verbindungen möglich. Weitere Informationen finden Sie unter Zuweisen von Abfragen zu Warteschlangen im Amazon Redshift Datenbankentwicklungshandbuch.	String
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Werte sind `cron`, `ondemand` und `timeseries` . `timeseries` Planung bedeutet, dass Instances am Ende jedes Intervalls geplant sind. `cron` Planung bedeutet, dass Instances am Anfang jedes Intervalls geplant sind. Ein `ondemand`-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Dies bedeutet, dass Sie die Pipeline nicht klonen oder neu erstellen müssen, um sie erneut auszuführen. Wenn Sie einen `ondemand`-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene `scheduleType` sein. Um `ondemand`-Pipelines zu verwenden, rufen Sie einfach den `ActivatePipeline` -Vorgang für jeden nachfolgenden Lauf auf.	Aufzählung
scriptArgument	Eine Liste der Variablen für das Skript. Sie können alternativ Ausdrücke direkt in das Skriptfeld einfügen. Mehrere Werte für scriptArgument sind hilfreich, wenn das Skript in Amazon S3 gespeichert ist. Beispiel: # {format (@scheduledStartTime, „YY-MM-DD HH:MM:SS"}\n# {format (plusPeriod (@, „1 Tag“)scheduledStartTime, „YY-MM-DD HH:MM:SS"}	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „ActiveIn stances“: {"ref“:“ Id "} myRunnableObject
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ healthStatusUpdated Zeit	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
@lastDeactivatedTime	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCompletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „waitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Ressourcen

Nachfolgend sind die AWS Data Pipeline-Ressourcenobjekte aufgelistet:

Objekte

- Ec2Resource
- EmrCluster
- HttpProxy

Ec2Resource

Eine Amazon EC2 EC2-Instance, die die durch eine Pipeline-Aktivität definierte Arbeit ausführt.

AWS Data Pipelineunterstützt jetzt IMDSv2 für die Amazon EC2 EC2-Instance, die eine sitzungsorientierte Methode verwendet, um die Authentifizierung beim Abrufen von Metadateninformationen von Instances besser handhaben zu können. Eine Sitzung beginnt und

beendet eine Reihe von Anfragen, die Software, die auf einer Amazon EC2 EC2-Instance ausgeführt wird, verwendet, um auf die lokal gespeicherten Amazon EC2 EC2-Instance-Metadaten und Anmeldeinformationen zuzugreifen. Die Software startet eine Sitzung mit einer einfachen HTTP-PUT-Anfrage an IMDSv2. IMDSv2 gibt ein geheimes Token an die Software zurück, die auf der Amazon EC2 EC2-Instance ausgeführt wird. Diese verwendet das Token als Passwort, um Anfragen an IMDSv2 nach Metadaten und Anmeldeinformationen zu richten.

> **Note**
>
> Um IMDSv2 für Ihre Amazon EC2 EC2-Instance zu verwenden, müssen Sie die Einstellungen ändern, da das Standard-AMI nicht mit IMDSv2 kompatibel ist. Sie können eine neue AMI-Version angeben, die Sie über den folgenden SSM-Parameter abrufen können:`/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs`.

Informationen zu standardmäßigen Amazon EC2 EC2-Instances, die AWS Data Pipeline erstellt werden, wenn Sie keine Instance angeben, finden Sie unter Amazon EC2-Standardinstanzen nach AWS-Region.

Beispiele

EC2-Classic

> **Important**
>
> Nur AWS Konten, die vor dem 4. Dezember 2013 erstellt wurden, unterstützen die EC2-Classic-Plattform. Wenn Sie über eines dieser Konten verfügen, haben Sie möglicherweise die Möglichkeit, EC2Resource-Objekte für eine Pipeline in einem EC2-Classic-Netzwerk anstelle einer VPC zu erstellen. Wir empfehlen dringend, Ressourcen für alle Ihre Pipelines in VPCs zu erstellen. Wenn Sie bereits über Ressourcen in EC2-Classic verfügen, empfehlen wir Ihnen außerdem, diese auf eine VPC zu migrieren.

Das folgende Beispielobjekt startet eine EC2-Instance in EC2-Classic, wobei einige optionale Felder gesetzt sind.

```
{
  "id" : "MyEC2Resource",
```

```
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroups" : [
    "test-group",
    "default"
  ],
  "keyPair" : "my-key-pair"
}
```

EC2-VPC

Das folgende Beispielobjekt startet eine EC2 Instance in einem nicht standardmäßigen VPC, wobei einige optionale Felder festgelegt sind.

```
{
  "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"
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
resourceRole	Die IAM-Rolle, die die Ressourcen steuert, auf die die Amazon EC2 EC2-Instance zugreifen kann.	String

Pflichtfelder	Beschreibung	Slot-Typ
role	Die IAM-Rolle, mit der die AWS Data Pipeline EC2-Instance erstellt wird.	String

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Sie müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausführungsreihenfolge für dieses Objekt festzulegen. Sie können dafür eine der folgenden Möglichkeiten auswählen: • Um sicherzustellen, dass alle Objekte in der Pipeline den Zeitplan übernehmen, legen Sie einen Zeitplan explizit für das Objekt fest: `"schedule": {"ref": "DefaultSchedule"}` . In den meisten Fällen ist es nützlich, den Zeitplanverweis auf das Standard-Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan übernehmen. • Wenn der Hauptplan in Ihrer Pipeline verschachtelte Zeitpläne enthält, können Sie ein übergeordnetes Objekt mit Zeitplanreferenz erstellen. Weitere Informationen zu optionalen Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	Referenzobjekt, z. B. `"schedule":{"ref":"myScheduleId"}`

Optionale Felder	Beschreibung	Slot-Typ
actionOnResourceFehlschlag	Die Aktion, die nach einem Ressourcenfehler dieser Ressource ausgeführt wird. Gültige Werte sind `"retryall"` und `"retrynone"`.	String
actionOnTaskFehlschlag	Die Aktion, die nach einem Aufgabenfehler dieser Ressource ausgeführt wird. Gültige Werte sind `"continue"` oder `"terminate"`.	String
associatePublicIpAdresse	Gibt an, ob der Instance eine öffentliche IP-Adresse zugewiesen wird. Wenn sich die Instance in Amazon EC2 oder Amazon VPC befindet, ist der Standardwert. `true` Andernfalls ist der Standardwert `false`.	Boolesch
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Fertigstellung der Remote-Arbeit. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
availabilityZone	Die Availability Zone, in der die Amazon EC2 EC2-Instance gestartet werden soll.	String
Deaktivieren Sie IMDS V1	Der Standardwert ist false und aktiviert sowohl IMDSv1 als auch IMDSv2. Wenn Sie ihn auf true setzen, wird IMDSv1 deaktiviert und es werden nur IMDSv2s bereitgestellt	Boolesch
failureAndRerunModus	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung

Optionale Felder	Beschreibung	Slot-Typ
httpProxy	Der Proxy-Host, der von Clients zum Verbinden mit AWS-Services verwendet wird.	Referenzobjekt, z. B. `"httpProxy": {"ref":"myHttpProxyId"}`
imageId	Die ID des für die Instance zu verwendenden AMI. Standardmäßig verwendet AWS Data Pipeline den HVM-AMI-Virtualisierungstyp . Die konkret eingesetzten AMI-IDs sind regionsspezifisch. Sie können das Standard-AMI überschreiben, indem Sie das von Ihnen gewählte HVM-AMI angeben. Weitere Informationen zu AMI-Typen finden Sie unter Linux AMI Virtualization Types und Finding a Linux AMI im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.	String
initTimeout	Die Zeit, die auf den Start der Ressource gewartet wird.	Intervall
instanceCount	Als veraltet gekennzeichnet.	Ganzzahl
instanceType	Der Typ der Amazon EC2 EC2-Instance, die gestartet werden soll.	String
keyPair	Der Name des Schlüsselpaars. Wenn Sie eine Amazon EC2 EC2-Instance starten, ohne ein key pair anzugeben, können Sie sich nicht bei ihr anmelden.	String
lateAfterTimeout	Die verstrichene Zeit nach dem Start der Pipeline, innerhalb derer das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall

Optionale Felder	Beschreibung	Slot-Typ
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Die maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl
minInstanceCount	Als veraltet gekennzeichnet.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. `"onFail": {"ref":"myActionId"}`
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant wurde oder noch ausgeführt wird.	Referenzobjekt, z. B. `"onLateAction": {"ref":"myActionId"}`
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. `"onSuccess": {"ref":"myActionId"}`
übergeordneter	Das übergeordnete Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	Die Amazon S3 S3-URI (z. B. `'s3://BucketName/Key/'`) für das Hochladen von Protokollen für die Pipeline.	String

Optionale Felder	Beschreibung	Slot-Typ
Region	Der Code für die Region, in der die Amazon EC2 EC2-Instance ausgeführt werden soll. Standardmäßig wird die Instance in derselben Region wie die Pipeline ausgeführt. Sie können die Instance in derselben Region als abhängige s Datenset ausführen.	Aufzählung
reportProgressTimeout	Das Timeout für aufeinanderfolgende Aufrufe von `reportProgress` durch Remote-Arbeit. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall
runAsUser	Der Benutzer, der ausgeführt werden soll. TaskRunner	String
runsOn	Dieses Feld ist für dieses Objekt nicht zulässig.	Referenzobjekt, z. B. `"runsOn": {"ref":"myResourceId"}`

Optionale Felder	Beschreibung	Slot-Typ
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang oder am Ende eines Intervalls oder bedarfsabhängig geplant werden sollen. Die Werte sind: • `timeseries` . Instanzen werden am Ende jedes Intervalls geplant. • `cron`. Instanzen werden zu Beginn jedes Intervalls geplant. • `ondemand`. Ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführ en. Sie müssen die Pipeline nicht klonen oder neu erstellen, um sie erneut auszuführ en. Wenn Sie einen On-Demand-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene `scheduleType` sein. Um On-Demand -Pipelines zu verwenden, rufen Sie den `ActivatePipeline` -Vorgang für jeden nachfolgenden Lauf auf.	Aufzählung
securityGroupIds	Die IDs einer oder mehrerer Amazon EC2-Sicherheitsgruppen, die für die Instances im Ressourcenpool verwendet werden sollen.	String
securityGroups	Eine oder mehrere Amazon EC2-Sicherheitsgru ppen, die für die Instances im Ressourcenpool verwendet werden sollen.	String

Optionale Felder	Beschreibung	Slot-Typ
spotBidPrice	Die maximale Datenmenge pro Stunde für Ihre Spot-Instance in Dollar, wobei es sich um einen Dezimalwert zwischen 0 und einschließlich 20,00 handelt.	String
subnetId	Die ID des Amazon EC2-Subnetzes, in dem die Instance gestartet werden soll.	String
terminateAfter	Die Anzahl der Stunden, nach denen die Ressource zu beenden ist.	Intervall
useOnDemandOnLastAttempt	Dieses Feld bestimmt, ob beim letzten Versuch, eine Spot-Instance anzufordern, stattdessen eine On-Demand-Instance angefordert wird. Auf diese Weise wird sichergestellt, dass wenn die vorherigen Versuche fehlgeschlagen sind, der letzte Versuch nicht unterbrochen wird.	Boolesch
workerGroup	Dieses Feld ist für dieses Objekt nicht zulässig.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. `"activeInstances": {"ref":"myRunnableObjectId"}`
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
cancellationReason	Der `cancellationReason` , wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Zuständigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	Schrittprotokolle sind nur bei Amazon EMR-Aktivitätsversuchen verfügbar.	String
errorId	Die Fehler-ID, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die Fehlermeldung, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@failureReason	Der Grund für den Ressourcenfehler.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle sind bei Versuchen für Amazon EMR-Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromInstanceId	Id des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@ Zeit healthSta tusUpdated	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
@lastDeactivatedTi me	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime
@ latestCom pletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Die geplante Endzeit für das Objekt.	DateTime
@scheduledStartTime	Die geplante Startzeit für das Objekt.	DateTime
@Status	Der Status des Objekts.	String
@Version	Die Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. `"waitingOn": {"ref":"myRunna bleObjectId"}`

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Position eines Objekts im Lebenszyklus. Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

EmrCluster

Stellt die Konfiguration eines Amazon EMR-Clusters dar. Dieses Objekt wird von EmrActivity und HadoopActivity zum Starten eines Clusters verwendet.

Inhalt

- Schedulers
- Amazon EMR-Release-Versionen
- Amazon EMR-Berechtigungen
- Syntax
- Beispiele
- Weitere Informationen finden Sie unter:

Schedulers

Scheduler bieten eine Möglichkeit, die Ressourcenzuweisung und Auftragspriorisierung in einem Hadoop-Cluster festzulegen. Administratoren oder Benutzer können einen Scheduler für verschiedene Klassen von Benutzern und Anwendungen auswählen. Ein Scheduler könnte Warteschlangen nutzen, um Ressourcen für Benutzer und Anwendungen zuzuweisen. Sie richten diese Warteschlangen beim Erstellen des Clusters ein. Anschließend können Sie für bestimmte Arbeits- und Benutzertypen eine höhere Priorität festlegen als für andere. Dieses Vorgehen ermöglicht die effiziente Nutzung von Cluster-Ressourcen, wenn mehrere Benutzer Arbeiten zum Cluster übermitteln. Es gibt drei Arten von Schedulern:

- FairScheduler— Versucht, Ressourcen gleichmäßig über einen längeren Zeitraum einzuplanen.

- CapacityScheduler— Verwendet Warteschlangen, damit Clusteradministratoren Benutzer Warteschlangen mit unterschiedlicher Priorität und Ressourcenzuweisung zuweisen können.
- Standard: wird vom Cluster verwendet, was über Ihre Site konfiguriert werden kann.

Amazon EMR-Release-Versionen

Eine Amazon-EMR-Version ist eine Gruppe von Open-Source-Anwendungen aus dem Big-Data-Ökosystem. Jede Version umfasst verschiedene Big-Data-Anwendungen, Komponenten und Funktionen, die Sie bei der Erstellung eines Clusters für die Installation und Konfiguration von Amazon EMR auswählen. Sie geben die Version unter Verwendung der Versionsbezeichnung an. Versionsbezeichnungen haben die Form `emr-x.x.x`. Zum Beispiel `emr-5.30.0`. Amazon EMR-Cluster basieren auf dem Release-Label `emr-4.0.0` und verwenden später die `releaseLabel` Eigenschaft, um das Release-Label eines `EmrCluster` Objekts anzugeben. Frühere Versionen verwenden die Eigenschaft `amiVersion`.

> **Important**
>
> Alle Amazon EMR-Cluster, die mit Version 5.22.0 oder höher erstellt wurden, verwenden Signature Version 4, um Anfragen an Amazon S3 zu authentifizieren. Einige frühere Versionen verwenden Signature Version 2. Die Unterstützung für Signature Version 2 wird eingestellt. Weitere Informationen finden Sie unter Amazon S3 Update — SigV2 Deprecation Period Extended and Modified. Wir empfehlen dringend, eine Amazon EMR-Release-Version zu verwenden, die Signature Version 4 unterstützt. Für frühere Versionen, beginnend mit EMR 4.7.x, wurde die neueste Version der Serie aktualisiert, um Signature Version 4 zu unterstützen. Wenn Sie eine frühere EMR-Version verwenden, empfehlen wir, die neueste Version der Serie zu verwenden. Vermeiden Sie außerdem Versionen vor EMR 4.7.0.

Überlegungen und Einschränkungen

Verwenden Sie die neueste Version von Task Runner

Wenn Sie ein selbstverwaltetes `EmrCluster` Objekt mit einem Release-Label verwenden, verwenden Sie den neuesten Task Runner. Weitere Informationen zu Task-Runner finden Sie unter Arbeiten mit Task Runner. Sie können Eigenschaftswerte für alle Amazon EMR-Konfigurationsklassifizierungen konfigurieren. Weitere Informationen finden Sie unter Configuring Applications im Amazon EMR Release Guide, in und in den the section called “EmrConfiguration” the section called “Eigenschaft” Objektreferenzen.

Support für IMDSv2

Bisher wurde nur AWS Data Pipeline IMDSv1 unterstützt. AWS Data PipelineUnterstützt jetzt IMDSv2 in Amazon EMR 5.23.1, 5.27.1 und 5.32 oder höher sowie Amazon EMR 6.2 oder höher. IMDSv2 verwendet eine sitzungsorientierte Methode, um die Authentifizierung beim Abrufen von Metadateninformationen von Instances besser handhaben zu können. Sie sollten Ihre Instanzen so konfigurieren, dass sie IMDSv2-Aufrufe tätigen, indem Sie benutzerverwaltete Ressourcen mit -2.0 erstellen. TaskRunner

Amazon EMR 5.32 oder höher und Amazon EMR 6.x

Die Release-Serien Amazon EMR 5.32 oder höher und 6.x verwenden Hadoop Version 3.x, wodurch grundlegende Änderungen bei der Bewertung des Klassenpfads von Hadoop im Vergleich zu Hadoop-Version 2.x eingeführt wurden. Gängige Bibliotheken wie Joda-Time wurden aus dem Klassenpfad entfernt.

Wenn EmrActivity oder eine HadoopActivity Jar-Datei ausführt, die Abhängigkeiten von einer Bibliothek hat, die in Hadoop 3.x entfernt wurde, schlägt der Schritt mit dem Fehler oder fehl. `java.lang.NoClassDefFoundError` `java.lang.ClassNotFoundException` Dies kann bei Jar-Dateien passieren, die mit den Release-Versionen von Amazon EMR 5.x problemlos ausgeführt wurden.

Um das Problem zu beheben, müssen Sie Abhängigkeiten von Jar-Dateien in den Hadoop-Klassenpfad eines `EmrCluster` Objekts kopieren, bevor Sie das oder das starten. `EmrActivity` `HadoopActivity` Dafür stellen wir ein Bash-Skript zur Verfügung. Das Bash-Skript ist an der folgenden Stelle verfügbar, beispielsweise in der AWS Region, in der Ihr `EmrCluster` Objekt ausgeführt wird. *`MyRegion`*`us-west-2`

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

Die Art und Weise, wie das Skript ausgeführt wird, hängt davon ab, ob `EmrActivity` es auf einer Ressource `HadoopActivity` ausgeführt wird, die von einer selbst verwalteten Ressource verwaltet wird, AWS Data Pipeline oder ob es auf einer selbst verwalteten Ressource ausgeführt wird.

Wenn Sie eine Ressource verwenden, die von verwaltet wirdAWS Data Pipeline, fügen Sie dem `EmrCluster` Objekt eine `bootstrapAction` hinzu. Das `bootstrapAction` gibt das Skript und die Jar-Dateien an, die als Argumente kopiert werden sollen. Sie können bis zu 255 `bootstrapAction` Felder pro `EmrCluster` Objekt hinzufügen, und Sie können ein

`bootstrapAction` Feld zu einem `EmrCluster` Objekt hinzufügen, das bereits über Bootstrap-Aktionen verfügt.

Um dieses Skript als Bootstrap-Aktion anzugeben, verwenden Sie die folgende Syntax: Dabei `JarFileRegion` handelt es sich um die Region, in der die Jar-Datei gespeichert ist, und jedes *`MyJarFilen`* ist der absolute Pfad einer Jar-Datei in Amazon S3, die in den Hadoop-Klassenpfad kopiert werden soll. Geben Sie standardmäßig keine Jar-Dateien an, die sich im Hadoop-Klassenpfad befinden.

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-
hadoop-classpath.sh,JarFileRegion,MyJarFile1,MyJarFile2[, ...]
```

Das folgende Beispiel spezifiziert eine Bootstrap-Aktion, die zwei Jar-Dateien in Amazon S3 kopiert: `my-jar-file.jar` und dieemr-dynamodb-tool-4.14.0-jar-with-dependencies.jar. Die im Beispiel verwendete Region ist 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"]
}
```

Sie müssen die Pipeline speichern und aktivieren, damit die Änderung an der neuen `bootstrapAction` Pipeline wirksam wird.

Wenn Sie eine selbstverwaltete Ressource verwenden, können Sie das Skript auf die Clusterinstanz herunterladen und es über die Befehlszeile mit SSH ausführen. Das Skript erstellt ein Verzeichnis mit dem Namen `/etc/hadoop/conf/shellprofile.d` und eine `datapipeline-jars.sh` in diesem Verzeichnis benannte Datei. Die als Befehlszeilenargumente bereitgestellten JAR-Dateien werden in ein Verzeichnis kopiert, das das Skript mit dem Namen erstellt. `/home/hadoop/`

`datapipeline_jars` Wenn Ihr Cluster anders eingerichtet ist, ändern Sie das Skript nach dem Herunterladen entsprechend.

Die Syntax für die Ausführung des Skripts in der Befehlszeile unterscheidet sich geringfügig von der im vorherigen Beispiel `bootstrapAction` gezeigten Syntax. Verwenden Sie Leerzeichen anstelle von Kommas zwischen Argumenten, wie im folgenden Beispiel gezeigt.

```
./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
```

Amazon EMR-Berechtigungen

Wenn Sie eine benutzerdefinierte IAM-Rolle erstellen, sollten Sie sorgfältig die Mindestberechtigungen berücksichtigen, die Ihr Cluster zur Ausführung seiner Aufgaben benötigt. Stellen Sie sicher, dass Sie Zugriff auf die erforderlichen Ressourcen gewähren, z. B. Dateien in Amazon S3 oder Daten in Amazon RDS, Amazon Redshift oder DynamoDB. Wenn Sie `visibleToAllUsers` auf „False“ festlegen möchten, muss Ihre Rolle über die entsprechenden Berechtigungen verfügen. Beachten Sie, dass `DataPipelineDefaultRole` nicht über diese Berechtigungen verfügt. Sie müssen entweder eine Vereinigung der `DataPipelineDefaultRole` Rollen `DefaultDataPipelineResourceRole` und als `EmrCluster` Objektrolle angeben oder zu diesem Zweck Ihre eigene Rolle erstellen.

Syntax

Objektaufruf-Felder	Beschreibung	Slot-Typ
schedule	Dieses Objekt wird innerhalb der Ausführung eines Zeitplanintervalls aufgerufen. Sie müssen einen Zeitplanverweis auf ein anderes Objekt angeben, um die Abhängigkeitsausführungsreihenfolge für dieses Objekt festzulegen. Sie können diese Anforderung erfüllen, indem Sie explizit einen Zeitplan für das Objekt festlegen, indem sie beispielsweise `"schedule": {"ref": "DefaultSchedule"}` angeben. In den meisten Fällen ist es besser, den Zeitplanverweis auf das Standard-	Referenzobjekt, z. B. `"schedule":{"ref": "myScheduleId"}`

Objektaufruf-Felder	Beschreibung	Slot-Typ
	Pipeline-Objekt zu setzen, damit alle Objekte diesen Zeitplan erben. Wenn die Pipeline über einen Baum mit Zeitplänen verfügt (Zeitplän e innerhalb des Hauptplans), können Sie ein übergeordnetes Objekt mit Zeitplänenreferenz erstellen. Weitere Informationen zu optionale n Beispiel-Zeitplankonfigurationen finden Sie unter https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Optionale Felder	Beschreibung	Slot-Typ
actionOnResourceFehlschlag	Die Aktion, die nach einem Ressourcenfehler dieser Ressource ausgeführt wird. Gültige Werte sind "`retryall`", der für die festgelegte Dauer wiederholt versucht, alle Aufgaben des Clusters durchzuführen, und "`retrynone` ".	String
actionOnTaskFehlschlag	Die Aktion, die nach einem Aufgabenfehler dieser Ressource ausgeführt wird. Gültige Werte sind "continue", was bedeutet, dass der Cluster nicht beendet wird, und "terminate".	String
additionalMasterSecurityGroupIds	Die ID zusätzlicher Master-Sicherheitsgruppen des EMR-Clusters, die dem Format sg-01XXXX6a entspricht. Weitere Informationen finden Sie unter Zusätzliche Amazon EMR-Sicherheitsgruppen im Amazon EMR Management Guide.	String
additionalSlaveSecurityGroupIds	Die ID zusätzlicher Slave-Sicherheitsgruppen des EMR-Clusters, die dem Format `sg-01XXXX6a` entspricht.	String

Optionale Felder	Beschreibung	Slot-Typ
amiVersion	Die Amazon Machine Image (AMI) -Version, die Amazon EMR zur Installation der Clusterknoten verwendet. Weitere Informationen finden Sie im Amazon EMR-Managementhandbuch.	String
applications	Anwendungen, die im Cluster mit durch Kommas getrennten Argumenten installiert werden sollen. Hive und Pig sind standardm äßig installiert. Dieser Parameter gilt nur für Amazon EMR Version 4.0 und höher.	String
attemptStatus	Der zuletzt gemeldete Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlo ssen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
availabilityZone	Die Availability Zone, in der der Cluster gestartet werden soll.	String
bootstrapAction	Eine Aktion, die ausgeführt werden soll, wenn der Cluster startet. Sie können durch Kommas getrennte Argumente festlegen. Wenn Sie mehrere Aktionen angeben möchten (maximal 255), fügen Sie die entsprechende Anzahl von `bootstrapAction` -Feldern hinzu. Standardmäßig wird der Cluster ohne Bootstrap -Aktionen gestartet.	String

Optionale Felder	Beschreibung	Slot-Typ
Konfiguration	Konfiguration für den Amazon EMR-Cluster. Dieser Parameter gilt nur für Amazon EMR Version 4.0 und höher.	Referenzobjekt, z. B. `"configuration":{"ref":"myEmrConfigurationId"}`
coreInstanceBidPreis	Der maximale Spot-Preis, den Sie bereit sind, für Amazon EC2 EC2-Instances zu zahlen. Wenn ein Angebotspreis angegeben ist, verwendet Amazon EMR Spot-Instances für die Instance-Gruppe. Angegeben in USD.	String
coreInstanceCount	Gibt an, wie viele Core-Knoten für den Cluster verwendet werden sollen.	Ganzzahl
coreInstanceType	Der Typ der Amazon EC2 EC2-Instance, die für Core-Knoten verwendet werden soll. Siehe Unterstützte Amazon EC2-Instances für Amazon EMR-Cluster .	String
coreGroupConfiguration	Die Konfiguration für die Amazon EMR-Cluster-Core-Instance-Gruppe. Dieser Parameter gilt nur für Amazon EMR Version 4.0 und höher.	Referenzobjekt, z. B. `“configuration”: {“ref”: “myEmrConfigurationId”}`
coreEbsConfiguration	Die Konfiguration für Amazon EBS-Volumes, die an jeden der Kernknoten in der Kerngruppe im Amazon EMR-Cluster angehängt werden. Weitere Informationen finden Sie unter Instance-Typen, die die EBS-Optimierung Support im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.	Referenzobjekt, z. B. `“coreEbsConfiguration”: {“ref”: “myEbsConfiguration”}`

Optionale Felder	Beschreibung	Slot-Typ
customAmiId	Gilt nur für Amazon EMR-Release-Version 5.7.0 und höher. Gibt die AMI-ID eines benutzerdefinierten AMI an, das verwendet werden soll, wenn Amazon EMR Amazon EC2 EC2-Instances bereitstellt. Sie kann auch anstelle von Bootstrap-Aktionen verwendet werden, um Cluster-Knotenkonfigurationen anzupassen. Weitere Informationen finden Sie unter dem folgenden Thema im Amazon EMR Management Guide. Verwenden eines benutzerdefinierten AMI	String
EbsBlockDeviceConfig	Die Konfiguration eines angeforderten Amazon EBS-Blockgeräts, das der Instanzgruppe zugeordnet ist. Diese umfasst eine feste Anzahl an Volumes, die jeder Instance in der Instance-Gruppe zugeordnet wird. Sie umfasst `volumesPerInstance` und `volumeSpecification`, wobei: • `volumesPerInstance` die Anzahl der EBS-Volumes mit einer bestimmten Volume-Konfiguration für alle zugeordneten Instances in der Instance-Gruppe ist. • `volumeSpecification` sind die Amazon EBS-Volume-Spezifikationen, wie Volume-Typ, IOPS und Größe in Gigabytes (GiB), die für das EBS-Volume angefordert werden, das an eine EC2-Instance im Amazon EMR-Cluster angehängt ist.	Referenzobjekt, z. B. `“EbsBlockDeviceConfig”: {“ref”: “myEbsBlockDeviceConfig”}`

Optionale Felder	Beschreibung	Slot-Typ
emrManagedMasterSecurityGroupId	Die ID der Master-Sicherheitsgruppe des Amazon EMR-Clusters, die der Form von `sg-01XXXX6a` folgt. Weitere Informationen finden Sie unter Configure Security Groups im Amazon EMR Management Guide.	String
emrManagedSlaveSecurityGroupId	Die ID der Slave-Sicherheitsgruppe des Amazon EMR-Clusters, die dem Formular `sg-01XXXX6a` folgt.	String
enableDebugging	Aktiviert das Debuggen auf dem Amazon EMR-Cluster.	String
failureAndRerunMode	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
hadoopSchedulerType	Der Scheduler-Typ des Clusters. Gültige Typen sind: `PARALLEL_FAIR_SCHEDULING` `PARALLEL_CAPACITY_SCHEDULING` und `DEFAULT_SCHEDULER` .	Aufzählung
httpProxy	Der Proxy-Host, der von Clients zum Verbinden mit den AWS-Services verwendet wird.	Referenzobjekt, zum Beispiel „HttpProxy“: {"ref“:“ myHttpProxy Id "}
initTimeout	Die Zeit, die auf den Start der Ressource gewartet wird.	Intervall
keyPair	Das Amazon EC2 EC2-Schlüsselpaar, das für die Anmeldung am Master-Knoten des Amazon EMR-Clusters verwendet werden soll.	String

Optionale Felder	Beschreibung	Slot-Typ
lateAfterTimeout	Die verstrichene Zeit nach dem Start der Pipeline, innerhalb derer das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
masterInstanceBidPreis	Der maximale Spot-Preis, den Sie bereit sind, für Amazon EC2 EC2-Instances zu zahlen. Es handelt sich um einen Dezimalwert zwischen 0 und einschließlich 20,00. Angegeben in USD. Wenn Sie diesen Wert festlegen, werden Spot-Instances für den Master-Knoten des Amazon EMR-Clusters aktiviert. Wenn ein Angebotspreis angegeben ist, verwendet Amazon EMR Spot-Instances für die Instance-Gruppe.	String
masterInstanceType	Der Typ der Amazon EC2 EC2-Instance, die für den Master-Knoten verwendet werden soll. Siehe Unterstützte Amazon EC2-Instances für Amazon EMR-Cluster .	String
masterGroupConfiguration	Die Konfiguration für die Amazon EMR-Cluster-Master-Instance-Gruppe. Dieser Parameter gilt nur für Amazon EMR Version 4.0 und höher.	Referenzobjekt, z. B. `“configuration”: {“ref”: “myEmrConfigurationId”}`
masterEbsConfiguration	Die Konfiguration für Amazon EBS-Volumes, die an jeden der Master-Knoten in der Master-Gruppe im Amazon EMR-Cluster angehängt werden. Weitere Informationen finden Sie unter Instance-Typen, die die EBS-Optimierung Support im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.	Referenzobjekt, z. B. `“masterEbsConfiguration”: {“ref”: “myEbsConfiguration”}`

Optionale Felder	Beschreibung	Slot-Typ
maxActiveInstances	Die maximale Anzahl gleichzeitiger aktiver Instances einer Komponente. Wiederholungen zählen nicht zur Anzahl der aktiven Instances.	Ganzzahl
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. `"onFail": {"ref":"myActionId"}`
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. `"onLateAction": {"ref":"myActionId"}`
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. `"onSuccess": {"ref":"myActionId"}`
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	Die Amazon S3 S3-URI (z. B. 's3://BucketName/Key/ ') zum Hochladen von Protokollen für die Pipeline.	String

Optionale Felder	Beschreibung	Slot-Typ
Region	Der Code für die Region, in der der Amazon EMR-Cluster ausgeführt werden soll. Standardmäßig wird der Cluster in derselben Region wie die Pipeline ausgeführt. Sie können den Cluster in derselben Region als abhängige s Datenset ausführen.	Aufzählung
releaseLabel	Versionsbezeichnung für den EMR-Cluster	String
reportProgressTime out	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in `reportProgress` . Wenn diese Option aktiviert ist, werden Remote-Ak tivitäten, die den Fortschritt für den angegeben en Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
resourceRole	Die IAM-Rolle, die zur Erstellung des Amazon EMR-Clusters AWS Data Pipeline verwendet wird. Die Standardrolle ist `DataPipel ineDefaultRole` .	String
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Zeitraum
role	Die IAM-Rolle wurde an Amazon EMR übergeben, um EC2-Knoten zu erstellen.	String
runsOn	Dieses Feld ist für dieses Objekt nicht zulässig.	Referenzobjekt, z. B. `"runsOn": {"ref":"m yResourceId"}`
Sicherheitskonfigu ration	Die ID der EMR-Sicherheitskonfiguration, die auf den Cluster angewendet wird. Dieser Parameter gilt nur für Amazon EMR Version 4.8.0 und höher.	String

Optionale Felder	Beschreibung	Slot-Typ
serviceAccessSecurityGroupId	Die ID für die Sicherheitsgruppe für den Servicezugriff des Amazon EMR-Clusters.	Zeichenfolge. Sie hat das Format `sg-01XXXX6a` , z. B. `sg-1234abcd` .
scheduleType	Mit dem Zeitplantyp können Sie angeben, ob die Objekte in Ihrer Pipeline-Definition am Anfang des Intervalls oder am Ende des Intervalls geplant werden sollen. Werte sind `cron`, `ondemand` und `timeseries` . Die `timeseries` -Planung bedeutet, dass Instances am Ende jedes Intervalls geplant sind. Die `cron`-Planung bedeutet, dass Instances am Anfang jedes Intervalls geplant sind. Ein `ondemand`-Zeitplan ermöglicht es Ihnen, eine Pipeline einmal pro Aktivierung auszuführen. Sie müssen die Pipeline nicht klonen oder neu erstellen, um sie erneut auszuführen. Wenn Sie einen `ondemand`-Zeitplan verwenden, muss er im Standardobjekt angegeben werden und der einzige für die Objekte in der Pipeline angegebene `scheduleType` sein. Um `ondemand`-Pipelines zu verwenden, rufen Sie einfach den `ActivatePipeline` -Vorgang für jeden nachfolgenden Lauf auf.	Aufzählung
subnetId	Die ID des Subnetzes, in dem der Amazon EMR-Cluster gestartet werden soll.	String
supportedProducts	Ein Parameter, der Software von Drittanbietern auf einem Amazon EMR-Cluster installiert, z. B. eine Drittanbieter-Distribution von Hadoop.	String

Optionale Felder	Beschreibung	Slot-Typ
taskInstanceBidPreis	Der maximale Spot-Preis, den Sie für EC2-Instances zu zahlen bereit sind. Geben Sie eine Dezimalzahl von 0 bis 20,00 ein. Angegeben in USD. Wenn ein Angebotspreis angegeben ist, verwendet Amazon EMR Spot-Instances für die Instance-Gruppe.	String
taskInstanceCount	Die Anzahl der Task-Knoten, die für den Amazon EMR-Cluster verwendet werden sollen.	Ganzzahl
taskInstanceType	Der Typ der Amazon EC2 EC2-Instance, die für Task-Knoten verwendet werden soll.	String
taskGroupConfiguration	Die Konfiguration für die Amazon EMR-Cluster-Task-Instance-Gruppe. Dieser Parameter gilt nur für Amazon EMR Version 4.0 und höher.	Referenzobjekt, z. B. `“configuration”: {“ref”: “myEmrConfigurationId”}`
taskEbsConfiguration	Die Konfiguration für Amazon EBS-Volumes, die an jeden der Task-Knoten in der Aufgabengruppe im Amazon EMR-Cluster angehängt werden. Weitere Informationen finden Sie unter Instance-Typen, die die EBS-Optimierung Support im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.	Referenzobjekt, z. B. `“taskEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
terminateAfter	Die Zeitspanne in Stunden, nach der die Ressource beendet wird.	Ganzzahl

Optionale Felder	Beschreibung	Slot-Typ
VolumeSpecification	Die Amazon EBS-Volumenspezifikationen, wie Volumetyp, IOPS und Größe in Gigabytes (GiB), die für das Amazon EBS-Volume angefordert werden, das an eine Amazon EC2-Instance im Amazon EMR-Cluster angehängt ist. Der Knoten kann ein Core-, Master- oder Aufgabenknoten sein. `VolumeSpecification` enthält: • `iops()` Ganzzahl. Die Anzahl der I/O-Operationen pro Sekunde (IOPS), die das Amazon EBS-Volume unterstützt, z. B. 1000. Weitere Informationen finden Sie unter EBS I/O Characteristics im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances. • `sizeinGB()` . Ganzzahl. Die Größe des Amazon EBS-Volumes in Gibibyte (GiB), zum Beispiel 500. Informationen zu gültigen Kombinationen von Volumetypen und Festplattengrößen finden Sie unter EBS-Volumetypen im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances. • `volumetType` . Schnur. Der Amazon EBS-Volumetyp, zum Beispiel gp2. Es werden die Volume-Typen standard, gp2, io1, st1, sc1 sowie weitere Typen unterstützt. Weitere Informationen finden Sie unter EBS-Volumetypen im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.	Referenzobjekt, z. B. `"VolumeSpecification": {"ref": "myVolumeSpecification"}`

Optionale Felder	Beschreibung	Slot-Typ
useOnDemandOnLastAttempt	Dieses Feld bestimmt, ob beim letzten Versuch, eine Ressource anzufordern, eine On-Demand-Instance statt einer Spot-Instance angefordert wird. Auf diese Weise wird sichergestellt, dass wenn die vorherigen Versuche fehlgeschlagen sind, der letzte Versuch nicht unterbrochen wird.	Boolesch
workerGroup	Dieses Feld ist bei diesem Objekt nicht zulässig.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, zum Beispiel „ActiveInstances“: {"ref“:“ Id "} myRunnableObject
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Zuständigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, zum Beispiel "cascadeFailedOn„: {" ref“:“ myRunnableObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
emrStepLog	Schrittprotokolle sind nur bei Amazon EMR-Aktivitätsversuchen verfügbar.	String
errorId	Die Fehler-ID, wenn dieses Objekt fehlgesch lagen ist.	String
errorMessage	Die Fehlermeldung, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
@failureReason	Der Grund für den Ressourcenfehler.	String
@finishedTime	Der Zeitpunkt, zu der dieses Objekt seine Ausführung beendet hat.	DateTime
hadoopJobLog	Hadoop-Jobprotokolle sind bei Versuchen für Amazon EMR-Aktivitäten verfügbar.	String
@healthStatus	Der Integritätsstatus des Objekts, der Erfolg oder Misserfolg der letzten Objekt-Instance widerspiegelt, die einen beendeten Zustand erreicht hat.	String
@healthStatusFromI nstanceId	ID des Objekts der letzten Instance, das einen beendeten Zustand erreicht hat.	String
@ Zeit healthSta tusUpdated	Zeitpunkt, zu dem der Servicestatus beim letzten Mal aktualisiert wurde.	DateTime
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
@lastDeactivatedTi me	Zeitpunkt, zu dem dieses Objekt zuletzt deaktiviert wurde.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@ latestCom pletedRun Zeit	Zeitpunkt des letzten Laufs, für den die Ausführung abgeschlossen wurde.	DateTime
@latestRunTime	Zeitpunkt des letzten Laufs, für den die Ausführung geplant war.	DateTime
@nextRunTime	Zeitpunkt des Laufs, der als nächstes geplant werden soll	DateTime
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, zum Beispiel „WaitingO n“: {"ref“:“ myRunnabl eObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Position eines Objekts im Lebenszyk lus. Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Beispiele

Es folgen Beispiele für diesen Objekttyp.

Inhalt

Starten Sie einen Amazon EMR-Cluster mit HadoopVersion

Example

Im folgenden Beispiel wird ein Amazon EMR-Cluster mit AMI-Version 1.0 und Hadoop 0.20 gestartet.

```
{
  "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"]
}
```

Starten Sie einen Amazon EMR-Cluster mit dem Release-Label emr-4.x oder höher

Example

Im folgenden Beispiel wird ein Amazon EMR-Cluster mit dem neueren `releaseLabel` Feld gestartet:

```
{
  "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"}
}
```

Installieren Sie zusätzliche Software auf Ihrem Amazon EMR-Cluster

Example

`EmrCluster`stellt das `supportedProducts` Feld bereit, das Drittanbieter-Software auf einem Amazon EMR-Cluster installiert. Damit können Sie beispielsweise eine benutzerdefinierte Distribution von Hadoop wie MapR installieren. Er akzeptiert eine durch Kommas getrennte Liste von Argumenten. Die Drittanbieter-Software kann diese Argumente lesen und darauf reagieren. Das folgende Beispiel zeigt, wie Sie mit dem Feld `supportedProducts` von `EmrCluster` einen benutzerdefinierten Cluster der MapR M3-Edition mit Karmasphere Analytics erstellen und ein `EmrActivity`-Objekt darauf ausführen.

```
{
    "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"
}
```

Deaktivieren der serverseitigen Verschlüsselung auf 3.x-Versionen

Example

Eine `EmrCluster`-Aktivität mit einer Hadoop Version 2.x, die von AWS Data Pipeline erstellt wurde, ermöglicht standardmäßig eine Verschlüsselung auf dem Server. Wenn Sie die serverseitige Verschlüsselung deaktivieren möchten, müssen Sie eine Bootstrap-Aktion in der Cluster-Objektdefinition festlegen.

Das folgende Beispiel erstellt eine `EmrCluster`-Aktivität, bei der die serverseitige Verschlüsselung deaktiviert ist:

```
{
   "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"]
}
```

Deaktivieren der serverseitigen Verschlüsselung auf 4.x-Versionen

Example

Sie müssen die serverseitige Verschlüsselung mit einem `EmrConfiguration`-Objekt deaktivieren.

Das folgende Beispiel erstellt eine `EmrCluster`-Aktivität, bei der die serverseitige Verschlüsselung deaktiviert ist:

```
{
   "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"
 }
```

Konfigurieren von Hadoop KMS ACLs und Erstellen von Verschlüsselungszonen in HDFS

Example

Die folgenden Objekte erstellen ACLs für Hadoop KMS sowie Verschlüsselungszonen und die entsprechenden Verschlüsselungsschlüssel in 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"
    }
```

Festlegen benutzerdefinierter IAM-Rollen

Example

Wird standardmäßig `DataPipelineDefaultRole` als Amazon EMR-Servicerolle und `DataPipelineDefaultResourceRole` als Amazon EC2 EC2-Instance-Profil AWS Data Pipeline übergeben, um Ressourcen in Ihrem Namen zu erstellen. Sie können jedoch eine benutzerdefinierte Amazon EMR-Servicerolle und ein benutzerdefiniertes Instance-Profil erstellen und diese stattdessen verwenden. AWS Data Pipelinesollte über ausreichende Berechtigungen verfügen, um Cluster mithilfe der benutzerdefinierten Rolle zu erstellen, und Sie müssen sie AWS Data Pipeline als vertrauenswürdige Entität hinzufügen.

Das folgende Beispielobjekt spezifiziert benutzerdefinierte Rollen für den Amazon EMR-Cluster:

```
{
   "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"
}
```

EmrCluster Ressource im AWS SDK for Java verwenden

Example

Das folgende Beispiel zeigt, wie Sie einen `EmrCluster` und verwenden`EmrActivity`, um einen Amazon EMR 4.x-Cluster zu erstellen, um einen Spark-Schritt mithilfe des Java-SDK auszuführen:

```
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);

    }

}
```

Einen Amazon EMR-Cluster in einem privaten Subnetz konfigurieren

Example

Dieses Beispiel enthält eine Konfiguration, mit der der Cluster in einem privaten Subnetz in einer VPC gestartet wird. Weitere Informationen finden Sie unter Starten von Amazon EMR-Clustern in einer VPC im Amazon EMR Management Guide. Diese Konfiguration ist optional. Sie können sie in einer beliebigen Pipeline verwenden, die ein `EmrCluster`-Objekt nutzt.

Um einen Amazon EMR-Cluster in einem privaten Subnetz zu starten, geben Sie`SubnetId`, `emrManagedMasterSecurityGroupIdemrManagedSlaveSecurityGroupId`, und `serviceAccessSecurityGroupId` in Ihrer `EmrCluster` Konfiguration an.

```
{
  "objects": [
    {
      "output": {
        "ref": "S3BackupLocation"
      },
      "input": {
        "ref": "DDBSourceTable"
      },
      "maximumRetries": "2",
      "name": "TableBackupActivity",
      "step": "s3://dynamodb-emr-#{myDDBRegion}/emr-ddb-storage-handler/2.1.0/emr-
ddb-2.1.0.jar,org.apache.hadoop.dynamodb.tools.DynamoDbExport,#{output.directoryPath},#{input.t
      "id": "TableBackupActivity",
      "runsOn": {
        "ref": "EmrClusterForBackup"
      },
      "type": "EmrActivity",
      "resizeClusterBeforeRunning": "false"
    },
    {
      "readThroughputPercent": "#{myDDBReadThroughputRatio}",
      "name": "DDBSourceTable",
      "id": "DDBSourceTable",
      "type": "DynamoDBDataNode",
      "tableName": "#{myDDBTableName}"
    },
    {
      "directoryPath": "#{myOutputS3Loc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-
mm-ss')}",
```

```
      "name": "S3BackupLocation",
      "id": "S3BackupLocation",
      "type": "S3DataNode"
    },
    {
      "name": "EmrClusterForBackup",
      "coreInstanceCount": "1",
      "taskInstanceCount": "1",
      "taskInstanceType": "m4.xlarge",
      "coreInstanceType": "m4.xlarge",
      "releaseLabel": "emr-4.7.0",
      "masterInstanceType": "m4.xlarge",
      "id": "EmrClusterForBackup",
      "subnetId": "#{mySubnetId}",
      "emrManagedMasterSecurityGroupId": "#{myMasterSecurityGroup}",
      "emrManagedSlaveSecurityGroupId": "#{mySlaveSecurityGroup}",
      "serviceAccessSecurityGroupId": "#{myServiceAccessSecurityGroup}",
      "region": "#{myDDBRegion}",
      "type": "EmrCluster",
      "keyPair": "user-key-pair"
    },
    {
      "failureAndRerunMode": "CASCADE",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "#{myPipelineLogUri}",
      "scheduleType": "ONDEMAND",
      "name": "Default",
      "id": "Default"
    }
  ],
  "parameters": [
    {
      "description": "Output S3 folder",
      "id": "myOutputS3Loc",
      "type": "AWS::S3::ObjectKey"
    },
    {
      "description": "Source DynamoDB table name",
      "id": "myDDBTableName",
      "type": "String"
    },
    {
      "default": "0.25",
```

```
      "watermark": "Enter value between 0.1-1.0",
      "description": "DynamoDB read throughput ratio",
      "id": "myDDBReadThroughputRatio",
      "type": "Double"
    },
    {
      "default": "us-east-1",
      "watermark": "us-east-1",
      "description": "Region of the DynamoDB table",
      "id": "myDDBRegion",
      "type": "String"
    }
  ],
  "values": {
     "myDDBRegion": "us-east-1",
      "myDDBTableName": "ddb_table",
      "myDDBReadThroughputRatio": "0.25",
      "myOutputS3Loc": "s3://s3_path",
      "mySubnetId": "subnet_id",
      "myServiceAccessSecurityGroup":  "service access security group",
      "mySlaveSecurityGroup": "slave security group",
      "myMasterSecurityGroup": "master security group",
      "myPipelineLogUri": "s3://s3_path"
  }
}
```

EBS-Volumes zu Cluster-Knoten hinzufügen

Example

Sie können EBS-Volumes an beliebige Knoten im EMR-Cluster innerhalb der Pipeline anfügen. Verwenden Sie zum Anfügen von EBS-Volumes an Knoten `coreEbsConfiguration`, `masterEbsConfiguration` und `TaskEbsConfiguration` in Ihrer `EmrCluster`-Konfiguration.

Dieses Beispiel für den Amazon EMR-Cluster verwendet Amazon EBS-Volumes für seine Master-, Task- und Core-Knoten. Weitere Informationen finden Sie unter Amazon EBS-Volumes in Amazon EMR im Amazon EMR Management Guide.

Diese Konfigurationen sind optional. Sie können sie in beliebigen Pipelines verwenden, die ein `EmrCluster`-Objekt nutzen.

Klicken Sie in der Pipeline auf die `EmrCluster`-Objektkonfiguration und dann auf Master EBS Configuration (Master-EBS-Konfiguration), Core EBS Configuration (Core-EBS-Konfiguration) oder Task EBS Configuration (Aufgaben-EBS-Konfiguration) und geben Sie die Konfigurationsdetails wie im folgenden Beispiel ein.

```
{
  "objects": [
    {
      "output": {
        "ref": "S3BackupLocation"
      },
      "input": {
        "ref": "DDBSourceTable"
      },
      "maximumRetries": "2",
      "name": "TableBackupActivity",
      "step": "s3://dynamodb-emr-#{myDDBRegion}/emr-ddb-storage-handler/2.1.0/emr-
ddb-2.1.0.jar,org.apache.hadoop.dynamodb.tools.DynamoDbExport,#{output.directoryPath},#{input.t
      "id": "TableBackupActivity",
      "runsOn": {
        "ref": "EmrClusterForBackup"
      },
      "type": "EmrActivity",
      "resizeClusterBeforeRunning": "false"
    },
    {
      "readThroughputPercent": "#{myDDBReadThroughputRatio}",
      "name": "DDBSourceTable",
      "id": "DDBSourceTable",
      "type": "DynamoDBDataNode",
      "tableName": "#{myDDBTableName}"
    },
    {
      "directoryPath": "#{myOutputS3Loc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-
mm-ss')}",
      "name": "S3BackupLocation",
      "id": "S3BackupLocation",
      "type": "S3DataNode"
    },
    {
      "name": "EmrClusterForBackup",
      "coreInstanceCount": "1",
      "taskInstanceCount": "1",
```

```
      "taskInstanceType": "m4.xlarge",
      "coreInstanceType": "m4.xlarge",
      "releaseLabel": "emr-4.7.0",
      "masterInstanceType": "m4.xlarge",
      "id": "EmrClusterForBackup",
      "subnetId": "#{mySubnetId}",
      "emrManagedMasterSecurityGroupId": "#{myMasterSecurityGroup}",
      "emrManagedSlaveSecurityGroupId": "#{mySlaveSecurityGroup}",
      "region": "#{myDDBRegion}",
      "type": "EmrCluster",
      "coreEbsConfiguration": {
        "ref": "EBSConfiguration"
      },
      "masterEbsConfiguration": {
        "ref": "EBSConfiguration"
      },
      "taskEbsConfiguration": {
        "ref": "EBSConfiguration"
      },
      "keyPair": "user-key-pair"
    },
    {
       "name": "EBSConfiguration",
        "id": "EBSConfiguration",
        "ebsOptimized": "true",
        "ebsBlockDeviceConfig" : [
            { "ref": "EbsBlockDeviceConfig" }
        ],
        "type": "EbsConfiguration"
    },
    {
        "name": "EbsBlockDeviceConfig",
        "id": "EbsBlockDeviceConfig",
        "type": "EbsBlockDeviceConfig",
        "volumesPerInstance" : "2",
        "volumeSpecification" : {
            "ref": "VolumeSpecification"
        }
    },
    {
      "name": "VolumeSpecification",
      "id": "VolumeSpecification",
      "type": "VolumeSpecification",
      "sizeInGB": "500",
```

```
      "volumeType": "io1",
      "iops": "1000"
    },
    {
      "failureAndRerunMode": "CASCADE",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "#{myPipelineLogUri}",
      "scheduleType": "ONDEMAND",
      "name": "Default",
      "id": "Default"
    }
  ],
  "parameters": [
    {
      "description": "Output S3 folder",
      "id": "myOutputS3Loc",
      "type": "AWS::S3::ObjectKey"
    },
    {
      "description": "Source DynamoDB table name",
      "id": "myDDBTableName",
      "type": "String"
    },
    {
      "default": "0.25",
      "watermark": "Enter value between 0.1-1.0",
      "description": "DynamoDB read throughput ratio",
      "id": "myDDBReadThroughputRatio",
      "type": "Double"
    },
    {
      "default": "us-east-1",
      "watermark": "us-east-1",
      "description": "Region of the DynamoDB table",
      "id": "myDDBRegion",
      "type": "String"
    }
  ],
  "values": {
     "myDDBRegion": "us-east-1",
      "myDDBTableName": "ddb_table",
      "myDDBReadThroughputRatio": "0.25",
      "myOutputS3Loc": "s3://s3_path",
```

```
      "mySubnetId": "subnet_id",
      "mySlaveSecurityGroup": "slave security group",
      "myMasterSecurityGroup": "master security group",
      "myPipelineLogUri": "s3://s3_path"
  }
}
```

Weitere Informationen finden Sie unter:

- EmrActivity

HttpProxy

HttpProxy ermöglicht es Ihnen, Ihren eigenen Proxy zu konfigurieren und Task Runner über ihn auf den AWS Data Pipeline Dienst zugreifen zu lassen. Es ist nicht erforderlich, einen ausgeführten TaskRunner mit diesen Informationen zu konfigurieren.

Beispiel für ein HttpProxy in TaskRunner

Die folgenden Pipeline-Definition zeigt ein `HttpProxy`-Objekt:

```
{
  "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": []
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
hostname	Der Host des Proxys, über den Clients eine Verbindung zu AWS-Services herstellen.	String
port	Port des Proxy-Hosts, den die Clients verwenden, um eine Verbindung zu AWS-Servi ces herzustellen.	String

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
*Passwort	Passwort für den Proxy.	String
s3 NoProxy	Deaktiviert den HTTP-Proxy, wenn eine Verbindung zu Amazon S3 hergestellt wird	Boolesch
username	Benutzername für den Proxy.	String
windowsDomain	Der Windows-Domänenname für NTLM Proxy.	String
windowsWorkgroup	Der Windows-Arbeitsgruppenname für NTLM Proxy.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Vorbedingungen

Nachfolgend sind die AWS Data Pipeline-Vorbedingungsobjekte aufgelistet:

Objekte

- DynamoDB DataExists
- DynamoDB TableExists
- Vorhanden
- S3 KeyExists
- S3 PrefixNotEmpty
- ShellCommandPrecondition

DynamoDB DataExists

Eine Vorbedingung, um zu überprüfen, ob Daten in einer DynamoDB-Tabelle vorhanden sind.

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
role	Legt die Rolle für die Ausführung der Vorbedingung fest.	String

Pflichtfelder	Beschreibung	Slot-Typ
tableName	Die zu prüfende DynamoDB-Tabelle.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. ondemand	Intervall
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
preconditionTimeout	Der Zeitraum ab dem die Vorbedingung als fehlgeschlagen gekennzeichnet ist, wenn sie noch nicht erfüllt ist	Intervall
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@cascadeFailedOn	Beschreibung der Zuständigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
currentRetryCount	Anzahl, wie oft die Vorbedingung in diesem Versuch probiert wurde.	String
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
lastRetryTime	Das letzte Mal, dass die Vorbedingung in diesem Versuch probiert wurde.	String
node	Der Knoten, für den diese Vorbedingung ausgeführt wird	Referenzobjekt, z. B. „node“: {"ref“:“ myRunnableObject Id "}
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime

Laufzeitfelder	Beschreibung	Slot-Typ
@scheduledEndTime	Endzeit für Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „WaitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

DynamoDB TableExists

Eine Vorbedingung, um zu überprüfen, ob die DynamoDB-Tabelle existiert.

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
role	Legt die Rolle für die Ausführung der Vorbedingung fest.	String
tableName	Die zu prüfende DynamoDB-Tabelle.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}

Optionale Felder	Beschreibung	Slot-Typ
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
preconditionTimeout	Der Zeitraum ab dem die Vorbedingung als fehlgeschlagen gekennzeichnet ist, wenn sie noch nicht erfüllt ist	Intervall
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeInstances“: {"ref“:“ myRunnableObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Zuständigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
currentRetryCount	Anzahl, wie oft die Vorbedingung in diesem Versuch probiert wurde.	String
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
lastRetryTime	Das letzte Mal, dass die Vorbedingung in diesem Versuch probiert wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
node	Der Knoten, für den diese Vorbedingung ausgeführt wird	Referenzobjekt, z. B. „node“: {"ref“:“ myRunnableObject Id "}
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „WaitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Vorhanden

Prüft, ob eine Datenknotenobjekt vorhanden ist.

> **Note**
>
> Wir empfehlen, stattdessen die vom System verwalteten Vorbedingungen zu verwenden. Weitere Informationen finden Sie unter Vorbedingungen.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Das `InputData`-Objekt verweist auf dieses Objekt, `Ready`, und auf ein anderes Objekt, das Sie in derselben Pipeline-Definitionsdatei definieren. `CopyPeriod` ist ein `Schedule`-Objekt.

```
{
  "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"
}
```

Syntax

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der	Intervall

Optionale Felder	Beschreibung	Slot-Typ
	festgelegten Startzeit abgeschlossen wird, wiederholt werden.	
failureAndRerunMod us	Beschreibt das Verhalten des Konsument enknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
preconditionTimeout	Der Zeitraum ab dem die Vorbedingung als fehlgeschlagen gekennzeichnet ist, wenn sie noch nicht erfüllt ist	Intervall

Optionale Felder	Beschreibung	Slot-Typ
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String

Laufzeitfelder	Beschreibung	Slot-Typ
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
hostname	Der Hostname des Clients, der den Aufgabenversuch aufnimmt.	String
node	Der Knoten, für den diese Vorbedingung ausgeführt wird.	Referenzobjekt, z. B. „node“: {"ref“:“ myRunnableObject Id "}
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „WaitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- ShellCommandPrecondition

S3 KeyExists

Prüft, ob ein Schlüssel in einem Amazon S3-Datenknoten vorhanden ist.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Die Vorbedingung wird ausgelöst, wenn der Schlüssel, `s3://mybucket/mykey`, auf den der `s3Key`-Parameter verweist, vorhanden ist.

```
{
"id" : "InputReady",
"type" : "S3KeyExists",
"role" : "test-role",
"s3Key" : "s3://mybucket/mykey"
}
```

Sie können `S3KeyExists` auch als Voraussetzung für die zweite Pipeline verwenden, die darauf wartet, dass die erste Pipeline abgeschlossen wird. Gehen Sie hierzu wie folgt vor:

1. Schreiben Sie am Ende der Fertigstellung der ersten Pipeline eine Datei in Amazon S3.
2. Erstellen Sie eine `S3KeyExists`-Vorbedingung für die zweite Pipeline.

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
role	Legt die Rolle für die Ausführung der Vorbedingung fest.	String
s3Key	Der Amazon S3 S3-Schlüssel.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout vor dem Versuch, die Remote-Arbeit noch einmal auszuführen. Wenn diese Option aktiviert ist, wird erneut versucht, eine Remote-Aktivität durchzuführen, die nach dem Start nicht innerhalb der festgelegten Zeit abgeschlossen wird.	Intervall
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden.	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maximumRetries	Maximale Anzahl der Versuche, die bei einem Fehler initiiert werden.	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}

Optionale Felder	Beschreibung	Slot-Typ
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
preconditionTimeout	Der Zeitraum ab dem die Vorbedingung als fehlgeschlagen gekennzeichnet ist, wenn sie noch nicht erfüllt ist.	Intervall
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in `reportProgress` . Wenn diese Option aktiviert ist, werden Remote-Aktivitäten, die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei aufeinander folgenden Versuchen.	Intervall

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeInstances“: {"ref“:“ myRunnableObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
currentRetryCount	Anzahl, wie oft die Vorbedingung in diesem Versuch probiert wurde.	String
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
lastRetryTime	Das letzte Mal, dass die Vorbedingung in diesem Versuch probiert wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
node	Der Knoten, für den diese Vorbedingung ausgeführt wird	Referenzobjekt, z. B. „node“: {"ref“:“ myRunnableObject Id "}
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Aktivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „WaitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Weitere Informationen finden Sie unter:

- ShellCommandPrecondition

S3 PrefixNotEmpty

Eine Voraussetzung, um zu überprüfen, ob die Amazon S3 S3-Objekte mit dem angegebenen Präfix (dargestellt als URI) vorhanden sind.

Beispiel

Es folgt ein Beispiel für die Verwendung dieses Objekttyps mit erforderlichen, optionalen und Ausdrucksfeldern.

```
{
  "id" : "InputReady",
  "type" : "S3PrefixNotEmpty",
  "role" : "test-role",
  "s3Prefix" : "#{node.filePath}"
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
role	Legt die Rolle für die Ausführung der Vorbedingung fest.	String
s3Prefix	Das Amazon S3 S3-Präfix zur Überprüfung der Existenz von Objekten.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. `ondemand`	Intervall
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
preconditionTimeout	Der Zeitraum ab dem die Vorbedingung als fehlgeschlagen gekennzeichnet ist, wenn sie noch nicht erfüllt ist	Intervall

Optionale Felder	Beschreibung	Slot-Typ
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeInstances“: {"ref“:“ myRunnabl eObject Id "}
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
currentRetryCount	Anzahl, wie oft die Vorbedingung in diesem Versuch probiert wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
lastRetryTime	Das letzte Mal, dass die Vorbedingung in diesem Versuch probiert wurde.	String
node	Der Knoten, für den diese Vorbedingung ausgeführt wird.	Referenzobjekt, z. B. „node“: {"ref“:“ myRunnableObject Id "}
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen.	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen.	DateTime
@Status	Der Status des Objekts.	String
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „WaitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Weitere Informationen finden Sie unter:

- ShellCommandPrecondition

ShellCommandPrecondition

Ein Unix-/Linux-Shell-Befehl, der als Voraussetzung ausgeführt werden kann.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "id" : "VerifyDataReadiness",
  "type" : "ShellCommandPrecondition",
  "command" : "perl check-data-ready.pl"
}
```

Syntax

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
command	Den auszuführenden Befehl. Dieser Wert und alle zugehörigen Parameter müssen in der Umgebung funktionieren, in der Sie den Task-Runner ausführen.	String
scriptUri	Ein Amazon S3-URI-Pfad für eine Datei, die heruntergeladen und als Shell-Befehl ausgeführt werden soll. Nur das Feld scriptUri oder das Befehlsfeld sollten vorhanden sein. scriptUri kann keine Parameter verwenden. Verwenden Sie stattdessen das Befehlsfeld.	String

Optionale Felder	Beschreibung	Slot-Typ
attemptStatus	Zuletzt gemeldeter Status von der Remote-Aktivität.	String
attemptTimeout	Timeout für die Remote-Arbeit abgeschlossen. Wenn diese Option aktiviert ist, kann eine Remote-Aktivität, die nicht innerhalb der festgelegten Startzeit abgeschlossen wird, wiederholt werden.	Intervall
failureAndRerunModus	Beschreibt das Verhalten des Konsumentenknotens, wenn Abhängigkeiten fehlschlagen oder erneut ausgeführt werden	Aufzählung
lateAfterTimeout	Die nach dem Start der Pipeline verstrichene Zeit, innerhalb der das Objekt abgeschlossen	Intervall

Optionale Felder	Beschreibung	Slot-Typ
	werden muss. Sie wird nur ausgelöst, wenn der Zeitplantyp nicht auf eingestellt ist. ondemand	
maximumRetries	Maximale Anzahl von Versuchen bei Ausfällen	Ganzzahl
onFail	Eine Aktion, die ausgeführt werden soll, wenn das aktuelle Objekt fehlschlägt.	Referenzobjekt, z. B. „onFail“: {"ref“:“ myActionId „}
onLateAction	Aktionen, die ausgelöst werden sollen, wenn ein Objekt noch nicht geplant oder noch nicht abgeschlossen wurde.	Referenzobjekt, z. B. "onLateAction„: {" ref“:“ myActionId „}
onSuccess	Eine Aktion, die ausgeführt wird, wenn das aktuelle Objekt erfolgreich ist.	Referenzobjekt, z. B. „onSuccess“: {"ref“:“ myActionId „}
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
preconditionTimeout	Der Zeitraum ab dem die Vorbedingung als fehlgeschlagen gekennzeichnet ist, wenn sie noch nicht erfüllt ist	Intervall
reportProgressTimeout	Timeout für aufeinanderfolgende Aufrufe von Remote-Arbeit in reportProgress. Wenn diese Option aktiviert ist, werden Remote-Aktivitäten , die den Fortschritt für den angegebenen Zeitraum nicht melden, als fehlgeschlagen angesehen und es wird erneut versucht.	Intervall
retryDelay	Die Zeitüberschreitungsdauer zwischen zwei Wiederholungsversuchen.	Intervall
scriptArgument	Argument, das an ein Shell-Skript übergeben werden soll	String

Optionale Felder	Beschreibung	Slot-Typ
stderr	Der Amazon S3 S3-Pfad, der umgeleitete Systemfehlermeldungen vom Befehl empfängt. Wenn Sie das `runsOn` Feld verwenden , muss es sich um einen Amazon S3 S3-Pfad handeln, da die Ressource, auf der Ihre Aktivität ausgeführt wird, vorübergehend ist. Wenn Sie jedoch das Feld `workerGroup` angeben, ist ein lokaler Dateipfad zulässig.	String
stdout	Der Amazon S3 S3-Pfad, der die umgeleite te Ausgabe des Befehls empfängt. Wenn Sie das `runsOn` Feld verwenden, muss es sich um einen Amazon S3 S3-Pfad handeln, da die Ressource, auf der Ihre Aktivität ausgeführt wird, vorübergehend ist. Wenn Sie jedoch das Feld `workerGroup` angeben, ist ein lokaler Dateipfad zulässig.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@activeInstances	Liste der aktuell geplanten aktiven Instance-Objekte.	Referenzobjekt, z. B. „activeIn stances“: {"ref“:“ Id "} myRunnableObject
@actualEndTime	Zeitpunkt, zu dem die Ausführung dieses Objekts abgeschlossen wurde.	DateTime
@actualStartTime	Zeitpunkt, zu dem die Ausführung dieses Objekts gestartet wurde.	DateTime
cancellationReason	Die cancellationReason, wenn dieses Objekt storniert wurde.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@cascadeFailedOn	Beschreibung der Abhängigkeitskette, bei der das Objekt fehlgeschlagen ist.	Referenzobjekt, z. B. "cascadeFailedOn„: {" ref“:“ myRunnabl eObject Id "}
emrStepLog	EMR-Schrittprotokolle nur bei EMR-Aktiv itätsversuchen verfügbar	String
errorId	Die errorId, wenn dieses Objekt fehlgeschlagen ist.	String
errorMessage	Die errorMessage, wenn dieses Objekt fehlgeschlagen ist.	String
errorStackTrace	Die Fehler-Stack-Ablaufverfolgung., wenn dieses Objekt fehlgeschlagen ist.	String
hadoopJobLog	Hadoop-Jobprotokolle für Versuche für EMR-basierte Aktivitäten verfügbar.	String
hostname	Der Hostname des Clients, der den Aufgabenv ersuch aufnimmt.	String
node	Der Knoten, für den diese Vorbedingung ausgeführt wird	Referenzobjekt, z. B. „node“: {"ref“:“ myRunnableObject Id "}
reportProgressTime	Der letzte Zeitpunkt, an dem die Remote-Ak tivität einen Fortschritt gemeldet hat.	DateTime
@scheduledEndTime	Endzeit für Objekt einplanen	DateTime
@scheduledStartTime	Startzeit für Objekt einplanen	DateTime
@Status	Der Status des Objekts.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String
@waitingOn	Beschreibung der Liste der Abhängigkeiten, auf die dieses Objekt wartet.	Referenzobjekt, z. B. „WaitingOn“: {"ref“:“ myRunnableObject Id "}

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Weitere Informationen finden Sie unter:

- ShellCommandActivity
- Vorhanden

Datenbanken

Nachfolgend sind die AWS Data Pipeline-Datenbankobjekte aufgelistet:

Objekte

- JdbcDatabase
- RdsDatabase

- RedshiftDatabase

JdbcDatabase

Definiert eine JDBC-Datenbank.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "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"
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
connectionString	Die JDBC-Verbindungszeichenfolge für den Zugriff auf die Datenbank.	String
jdbcDriverClass	Die Treiberklasse, die vor dem Herstellen der JDBC-Verbindung geladen werden soll.	String
*Passwort	Das anzugebende Passwort.	String
username	Der Benutzername, der anzugeben ist, wenn eine Verbindung zur Datenbank hergestellt wird.	String

Optionale Felder	Beschreibung	Slot-Typ
databaseName	Name der logischen Datenbank für das Anfügen.	String
jdbcDriverJarUri	Der Amazon S3-Speicherort der JAR-Datei des JDBC-Treibers für die Verbindung mit der Datenbank. AWS Data Pipeline muss über die Leseberechtigung für diese JAR-Datei verfügen.	String
jdbcProperties	Paare der Form A=B, die als Eigenschaften für JDBC-Verbindungen für diese Datenbank festgelegt werden	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte	String

Systemfelder	Beschreibung	Slot-Typ
	ergeben Instance-Objekte, die Versuchsobjekte ausführen.	

RdsDatabase

Definiert eine Amazon RDS-Datenbank.

> **Note**
>
> RdsDatabase unterstützt Aurora nicht. Verwenden Sie es stattdessen the section called "JdbcDatabase" für Aurora.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "id" : "MyRdsDatabase",
  "type" : "RdsDatabase",
  "region" : "us-east-1",
  "username" : "user_name",
  "*password" : "my_password",
  "rdsInstanceId" : "my_db_instance_identifier"
}
```

Für die Oracle-Engine ist das Feld `jdbcDriverJarUri` eine Pflichtangabe. Sie können den folgenden Treiber festlegen: `http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html`. Für die SQL-Server-Engine ist das Feld `jdbcDriverJarUri` eine Pflichtangabe. Sie können den folgenden Treiber festlegen: `https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774`. Für die MySQL- und PostgreSQL-Engines ist das Feld `jdbcDriverJarUri` optional.

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
*Passwort	Das anzugebende Passwort.	String
rdsInstanceId	Die `DBInstanceIdentifier` Eigenschaft der DB-Instance.	String
username	Der Benutzername, der anzugeben ist, wenn eine Verbindung zur Datenbank hergestellt wird.	String

Optionale Felder	Beschreibung	Slot-Typ
databaseName	Name der logischen Datenbank für das Anfügen.	String
jdbcDriverJarUri	Der Amazon S3-Speicherort der JAR-Datei des JDBC-Treibers für die Verbindung mit der Datenbank. AWS Data Pipeline muss über die Leseberechtigung für diese JAR-Datei verfügen. Für MySQL- und PostgreSQL-Engines wird der Standardtreiber verwendet, wenn dieses Feld nicht angegeben ist. Sie können den Standardwert jedoch mit diesem Feld überschreiben. Für die Oracle- und SQL Server-Engines ist dieses Feld eine Pflichtangabe.	String
jdbcProperties	Paare der Form A=B, die als Eigenschaften für JDBC-Verbindungen für diese Datenbank festgelegt werden	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“:

Optionale Felder	Beschreibung	Slot-Typ
		{"ref“:“ myBaseObject Id "}
Region	Der Code für die Region, in der die Datenbank vorhanden ist. Beispiel: us-east-1.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

RedshiftDatabase

Definiert eine Amazon Redshift Redshift-Datenbank. `RedshiftDatabase`stellt die Eigenschaften der Datenbank dar, die von Ihrer Pipeline verwendet wird.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
```

```
  "id" : "MyRedshiftDatabase",
  "type" : "RedshiftDatabase",
  "clusterId" : "myRedshiftClusterId",
  "username" : "user_name",
  "*password" : "my_password",
  "databaseName" : "database_name"
}
```

Standardmäßig nutzt das Objekt den Postgres-Treiber, für den das Feld `clusterId` erforderlich ist. Um den Amazon Redshift Redshift-Treiber zu verwenden, geben Sie stattdessen die Amazon Redshift Redshift-Datenbankverbindungszeichenfolge aus der Amazon Redshift Redshift-Konsole (beginnt mit „jdbc:redshift:“) in das Feld ein. `connectionString`

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
*Passwort	Das anzugebende Passwort.	String
username	Der Benutzername, der anzugeben ist, wenn eine Verbindung zur Datenbank hergestellt wird.	String

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
clusterId	Die ID, die der Benutzer bei der Erstellung des Amazon Redshift Redshift-Clusters angegeben hat. Wenn der Endpunkt für Ihren Amazon Redshift Redshift-Cluster beispielsweise mydb.example.us-east-1.redshift.amazonaws.com lautet, lautet die korrekte ID. `mydb` Sie können diesen Wert in der Amazon Redshift-Konsole über "Cluster Identifier" oder "Cluster Name" ermitteln.	String

Erforderliche Gruppe (mindestens eine der folgenden ist erforderlich)	Beschreibung	Slot-Typ
connectionString	Der JDBC-Endpunkt für die Verbindung mit einer Amazon Redshift Redshift-Instance, die einem anderen Konto als der Pipeline gehört. Sie können nicht sowohl `connectionString` als auch `clusterId` angeben.	String

Optionale Felder	Beschreibung	Slot-Typ
databaseName	Name der logischen Datenbank für das Anfügen.	String
jdbcProperties	Paare der Form A=B müssen als Eigenschaften für JDBC-Verbindungen für diese Datenbank festgelegt werden.	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“: {"ref“:“ myBaseObject Id "}
Region	Der Code für die Region, in der die Datenbank vorhanden ist. Beispiel: us-east-1.	Aufzählung

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Datenformate

Nachfolgend sind die AWS Data Pipeline-Datenformatobjekte aufgelistet:

Objekte

- CSV-Datenformate
- Custom Data Format
- DynamoDB DataFormat
- DynamoDB ExportDataFormat
- RegEx Datenformat
- TSV-Datenformate

CSV-Datenformate

Ein durch Kommas getrenntes Datenformat, bei dem das Trennzeichen für Spalten ein Komma und das Datensatztrennzeichen ein Zeilenumbruch ist.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "id" : "MyOutputDataType",
  "type" : "CSV",
```

```
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Syntax

Optionale Felder	Beschreibung	Slot-Typ
column	Spaltenname mit Datentyp, der von jedem Feld für die Daten angegeben wird, die von diesem Datenknoten beschrieben werden. Beispiel: Bei Hostname STRING verwenden Sie für mehrere Werte Spaltennamen und Datentypen, die durch ein Leerzeichen getrennt sind.	String
escapeChar	Ein Zeichen (z. B."\"), das den Parser anweist, das nächste Zeichen zu ignorieren.	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String

Systemfelder	Beschreibung	Slot-Typ
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Custom Data Format

Ein benutzerdefiniertes Datenformat, das auf einer Kombination eines bestimmten Spaltentrennzeichens, Datensatztrennzeichens und des Escape-Zeichens basiert.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "id" : "MyOutputDataType",
  "type" : "Custom",
  "columnSeparator" : ",",
  "recordSeparator" : "\n",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
columnSeparator	Das Zeichen, mit dem das Ende einer Spalte in einer Datendatei kennzeichnet wird.	String

Optionale Felder	Beschreibung	Slot-Typ
column	Spaltenname mit Datentyp, der von jedem Feld für die Daten angegeben wird, die von diesem Datenknoten beschrieben werden. Beispiel: Bei Hostname STRING verwenden Sie für mehrere Werte Spaltennamen und Datentypen, die durch ein Leerzeichen getrennt sind.	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}
recordSeparator	Das Zeichen, mit dem das Ende einer Zeile in einer Datendatei kennzeichnet wird, z. B. "\n". Es werden nur einzelne Zeichen unterstützt.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

DynamoDB DataFormat

Wendet ein Schema auf eine DynamoDB-Tabelle an, um sie über eine Hive-Abfrage zugänglich zu machen. `DynamoDBDataFormat`wird mit einem `HiveActivity` Objekt und einer `DynamoDBDataNode` Eingabe und Ausgabe verwendet. `DynamoDBDataFormat`erfordert, dass Sie alle Spalten in Ihrer Hive-Abfrage angeben. Mehr Flexibilität bei der Angabe bestimmter Spalten in einer Hive-Abfrage oder Amazon S3 S3-Unterstützung finden Sie unterDynamoDB ExportDataFormat.

> **Note**
>
> Boolesche DynamoDB-Typen sind nicht zu booleschen Hive-Typen zugeordnet. Es ist aber möglich, DynamoDB-Ganzzahlwerte von 0 oder 1 den booleschen Hive-Typen zuzuordnen.

Beispiel

Das folgende Beispiel zeigt, wie Sie mit `DynamoDBDataFormat` ein Schema einer `DynamoDBDataNode`-Eingabe zuweisen, wodurch ein `HiveActivity`-Objekt auf die Daten nach benannten Spalten zugreifen und die Daten in eine `DynamoDBDataNode`-Ausgabe kopieren kann.

```
{
  "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"
    }
  ]
}
```

Syntax

Optionale Felder	Beschreibung	Slot-Typ
column	Der Spaltenname mit dem Datentyp, der von jedem Feld für die Daten angegeben wird, die von diesem Datenknoten beschrieben werden. Zum Beispiel `hostname STRING`. Verwenden Sie für mehrere Werte Spaltennamen und Datentypen, die durch ein Leerzeichen getrennt sind.	String
übergeordneter	Das übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Die Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Der Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Die Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

DynamoDB ExportDataFormat

Wendet ein Schema auf eine DynamoDB-Tabelle an, um sie über eine Hive-Abfrage zugänglich zu machen. Verwenden Sie `DynamoDBExportDataFormat` zusammen mit einem `HiveCopyActivity`-Objekt und `DynamoDBDataNode` oder der `S3DataNode`-Ein- und Ausgabe. `DynamoDBExportDataFormat` hat folgende Vorteile:

- Bietet sowohl DynamoDB- als auch Amazon S3 S3-Unterstützung
- Ermöglicht das Filtern von Daten nach bestimmten Spalten in der Hive-Abfrage
- Exportiert alle Attribute aus DynamoDB, auch wenn Sie ein dünnes Schema haben

> **Note**
>
> Boolesche DynamoDB-Typen sind nicht zu booleschen Hive-Typen zugeordnet. Es ist aber möglich, DynamoDB-Ganzzahlwerte von 0 oder 1 den booleschen Hive-Typen zuzuordnen.

Beispiel

Das folgende Beispiel zeigt, wie Sie mit `HiveCopyActivity` und `DynamoDBExportDataFormat` Daten von einem `DynamoDBDataNode` auf einen anderen kopieren können, während gleichzeitig Daten basierend auf einem Zeitstempel gefiltert werden.

```
{
  "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"
    }
  ]
}
```

Syntax

Optionale Felder	Beschreibung	Slot-Typ
column	Spaltenname mit Datentyp, der von jedem Feld für die Daten angegeben wird, die von diesem Datenknoten beschrieben werden. Beispiel: hostname STRING	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

RegEx Datenformat

Ein benutzerdefiniertes Datenformat, das durch einen regulären Ausdruck definiert wird.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "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"
  ]
}
```

Syntax

Optionale Felder	Beschreibung	Slot-Typ
column	Spaltenname mit Datentyp, der von jedem Feld für die Daten angegeben wird, die von diesem Datenknoten beschrieben werden. Beispiel: Bei Hostname STRING verwenden Sie für mehrere Werte Spaltennamen und Datentypen, die durch ein Leerzeichen getrennt sind.	String
inputRegEx	Der reguläre Ausdruck zum Analysieren einer S3-Eingabedatei. inputRegEx bietet eine Möglichkeit, Spalten aus relativ unstrukturierten Daten in einer Datei abzurufen.	String

Optionale Felder	Beschreibung	Slot-Typ
outputFormat	Die Spaltenfelder wurden von %1$s %2$s abgerufen inputRegEx, aber mithilfe der Java-Formatierungssyntax als %1$s %2$s referenziert.	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ Id "} myBaseObject

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

TSV-Datenformate

Ein durch Kommas getrenntes Datenformat, bei dem das Trennzeichen für Spalten ein Tabulatorzeichen und das Datensatztrennzeichen ein Zeilenumbruch ist.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp.

```
{
  "id" : "MyOutputDataType",
  "type" : "TSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Syntax

Optionale Felder	Beschreibung	Slot-Typ
column	Spaltenname und Datentyp der Daten, die von diesem Datenknoten beschrieben werden. So gibt `"Name STRING"` eine Spalte mit dem Namen `Name` und dem Datentyp `STRING` an. Trennen Sie mehrere Spaltenname/Datentyp-Paare durch Kommas (wie im Beispiel gezeigt).	String
columnSeparator	Das Zeichen, mit dem die Felder einer Spalte von den Feldern der nächsten Spalte getrennt werden. Standardeinstellung: "\t".	String
escapeChar	Ein Zeichen (z. B."\"), das den Parser anweist, das nächste Zeichen zu ignorieren.	String
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“: {"ref“:“ myBaseObject Id "}
recordSeparator	Das Zeichen, das die Datensätze voneinander trennt. Standardeinstellung: "\n".	String

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Aktionen

Nachfolgend sind die AWS Data Pipeline-Aktionsobjekte aufgelistet:

Objekte

- SnsAlarm
- Beenden

SnsAlarm

Sendet eine Amazon SNS SNS-Benachrichtigung, wenn eine Aktivität fehlschlägt oder erfolgreich abgeschlossen wird.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Die Werte für `node.input` und `node.output` stammen vom Datenknoten oder der Aktivität, die im Feld `onSuccess` auf dieses Objekt verweist.

```
{
  "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}."
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
Nachricht	Der Textkörper der Amazon SNS-Benac hrichtigung.	String
role	Die IAM-Rolle für die Erstellung des Amazon SNS-Alarms.	String
subject	Die Betreffzeile der Amazon SNS-Benac hrichtigung.	String
topicArn	Der Amazon SNS-Thema-Ziel-ARN für die Nachricht.	String

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
node	Der Knoten, für den diese Aktion ausgeführt wird.	Referenzobjekt, z. B. „node“: {"ref“:“

Laufzeitfelder	Beschreibung	Slot-Typ
		myRunnableObject Id "}
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Beenden

Eine Aktion, die eine Stornierung von ausstehenden oder nicht abgeschlossenen Aktivitäten, Ressourcen oder Datenknoten auslöst. AWS Data Pipeline versucht, die Aktivität, Ressource oder den Datenknoten in den Status CANCELLED zu versetzen, wenn sie nicht durch den Wert `lateAfterTimeout` gestartet wird.

Sie können keine Aktionen beenden, die `onSuccess`-, `OnFail`- oder `onLateAction`-Ressourcen beinhalten.

Beispiel

Es folgt ein Beispiel für diesen Objekttyp. Bei diesem Beispiel enthält das Feld `onLateAction` `MyActivity` einen Verweis auf die Aktion `DefaultAction1`. Wenn Sie eine Aktion für `onLateAction` bereitstellen, müssen Sie auch einen `lateAfterTimeout`-Wert für den Zeitraum seit dem geplanten Start der Pipeline festlegen, nach dem die Aktivität als verspätet betrachtet wird.

```
{
  "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"
}
```

Syntax

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
node	Der Knoten, für den diese Aktion ausgeführt wird.	Referenzobjekt, zum Beispiel „node“:

Laufzeitfelder	Beschreibung	Slot-Typ
		{"ref“:“ myRunnabl eObject Id "}
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Plan

Legt den Zeitplan für ein geplantes Ereignis fest, z. B. die Ausführung einer Aktivität.

> **Note**
>
> Wenn die Startzeit eines Zeitplans in der Vergangenheit liegt, gleicht AWS Data Pipeline Ihre Pipeline aus und fängt umgehend damit an, Ausführungen zu planen. Dabei beginnt die Planung zur angegebenen Startzeit. Wählen Sie für Tests/Entwicklung ein relativ kurzes Intervall. Andernfalls versucht AWS Data Pipeline, alle Ausführungen Ihrer Pipeline für diesen Zeitraum in die Warteschlange zu versetzen und zu planen. AWS Data Pipeline versucht, zufällige Ausgleichungen zu verhindern, wenn die Pipeline-Komponente `scheduledStartTime` länger als 1 Tag zurückliegt. Dazu wird die Pipeline-Aktivierung gesperrt.

Beispiele

Es folgt ein Beispiel für diesen Objekttyp. Es definiert einen Zeitplan für jede Stunde ab 00:00:00 Uhr am 01.09.2012 bis um 00:00:00 Uhr am 01.10.2012. Der erste Zeitraum endet um 01:00:00 Uhr am 01.09.2012.

```
{
  "id" : "Hourly",
  "type" : "Schedule",
  "period" : "1 hours",
  "startDateTime" : "2012-09-01T00:00:00",
  "endDateTime" : "2012-10-01T00:00:00"
}
```

Die folgende Pipeline startet um `FIRST_ACTIVATION_DATE_TIME` und wird jede Stunde bis um 22:00:00 Uhr am 25.04.2014 ausgeführt.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "endDateTime": "2014-04-25T22:00:00"
   }
```

Die folgende Pipeline startet um `FIRST_ACTIVATION_DATE_TIME` und wird jede Stunde ausgeführt. Nach dreimaliger Ausführung ist sie abgeschlossen.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

Die folgende Pipeline beginnt um 22:00:00 Uhr am 25.04.2014, wird stündlich ausgeführt und endet nach dreimaliger Ausführung.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startDateTime": "2014-04-25T22:00:00",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

On-Demand mit dem Standardobjekt

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
}
```

On-demand mit explizitem Zeitplanobjekt

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
},
{
  "name": "DefaultSchedule",
  "type": "Schedule",
  "id": "DefaultSchedule",
  "period": "ONDEMAND_PERIOD",
  "startAt": "ONDEMAND_ACTIVATION_TIME"
},
```

Die folgenden Beispiele zeigen, wie ein Zeitplan vom Standardobjekt übernommen werden kann, explizit für das Objekt festgelegt werden kann oder durch eine übergeordnete Objektreferenz übergeben werden kann:

Zeitplan vom Standardobjekt übernommen

```
{
```

```
  "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!'"
    }
  ]
}
```

Expliziter Zeitplan für das Objekt

```
{
  "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!'"
    }
  ]
}
```

Zeitplan von übergeordneter Referenz

```
{
  "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!'"
     }
  ]
}
```

Syntax

Pflichtfelder	Beschreibung	Slot-Typ
Zeitraum	Die vorgesehene Häufigkeit der Pipeline-Ausführung. Das Format ist "N [Minuten | Stunden | Tage | Wochen | Monate]", wobei N eine Zahl gefolgt von einem der Zeitspezi	Intervall

Pflichtfelder	Beschreibung	Slot-Typ
	fizierer ist. Beispiel: "15 Minuten", führt die Pipeline alle 15 Minuten aus. Der Mindestze itraum beträgt 15 Minuten und der maximale Zeitraum beträgt 3 Jahre.	

Erforderliche Gruppe (mindestens eine der folgenden ist erforderl ich)	Beschreibung	Slot-Typ
startAt	Das Datum und der Zeitpunkt, an dem die geplante Pipeline gestartet werden soll. Der gültige Wert ist FIRST_ACTIVATION_D ATE_TIME, der zugunsten der Erstellung einer bedarfsgesteuerten Pipeline als veraltet markiert ist.	Aufzählung
startDateTime	Das Datum und die Uhrzeit zum Starten der geplanten Ausführungen. Sie müssen entweder startDateTime oder startAt verwenden, aber nicht beide.	DateTime

Optionale Felder	Beschreibung	Slot-Typ
endDateTime	Das Datum und die Uhrzeit zum Starten der geplanten Ausführungen. Muss ein Datum und eine Uhrzeit nach dem Wert von startDateTime oder StartAt liegen. Das Standardverhalten besteht darin, Ausführungen so lange zu planen, bis die Pipeline heruntergefahren wird.	DateTime

Optionale Felder	Beschreibung	Slot-Typ
Ereignisse	Gibt an, wie oft die Pipeline ausgeführt werden soll, nachdem sie aktiviert wurde. Sie können keine Vorkommen mit verwenden. endDateTime	Ganzzahl
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@firstActivationTime	Zeit der Objekterstellung.	DateTime
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Dienstprogramme

Die folgenden Dienstprogrammobjekte konfigurieren andere Pipeline-Objekte:

Themen

ShellScriptConfig

Wird zusammen mit einer Aktivität verwendet, um ein Shell-Skript für preActivityTask Config und postActivityTask Config auszuführen. Dieses Objekt ist für HadoopActivity, HiveActivityHiveCopyActivity, und verfügbar PigActivity. Sie geben einen S3-URI und eine Liste von Argumenten für das Skript an.

Beispiel

A ShellScriptConfig mit Argumenten:

```
{
   "id" : "ShellScriptConfig_1”,
   "name" : “prescript”,
   "type" : "ShellScriptConfig",
   "scriptUri": “s3://my-bucket/shell-cleanup.sh”,
   "scriptArgument" : ["arg1","arg2"]
 }
```

Syntax

Dieses Objekt enthält die folgenden Felder.

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“: {"ref“:“ myBaseObject Id "}
scriptArgument	Eine Liste der Argumente für das Shell-Skript	String

Optionale Felder	Beschreibung	Slot-Typ
scriptUri	Der URI des Skripts in Amazon S3, das heruntergeladen und ausgeführt werden soll.	String

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

EmrConfiguration

Das EmrConfiguration Objekt ist die Konfiguration, die für EMR-Cluster mit Versionen 4.0.0 oder höher verwendet wird. Konfigurationen (als Liste) sind ein Parameter für den RunJobFlow API-Aufruf. Die Konfigurations-API für Amazon EMR verwendet eine Klassifizierung und Eigenschaften. AWS Data Pipelineverwendet EmrConfiguration mit entsprechenden Property-Objekten, um eine EmrCluster Anwendung wie Hadoop, Hive, Spark oder Pig auf EMR-Clustern zu konfigurieren, die in einer Pipeline-Ausführung gestartet wurden. Da die Konfiguration nur für neue Cluster geändert werden kann, können Sie kein EmrConfiguration Objekt für vorhandene Ressourcen bereitstellen. Weitere Informationen finden Sie unter https://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/.

Beispiel

Das folgende Konfigurationsobjekt legt die `io.file.buffer.size` und `fs.s3.block.size` Eigenschaften in `core-site.xml` fest:

```
[
   {
      "classification":"core-site",
      "properties":
      {
         "io.file.buffer.size": "4096",
         "fs.s3.block.size": "67108864"
      }
   }
]
```

Die entsprechende Pipeline-Objektdefinition verwendet ein EmrConfiguration Objekt und eine Liste von Property-Objekten im `property` Feld:

```
{
  "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"
    }
  ]
}
```

Im folgenden Beispiel wird eine verschachtelte Konfiguration verwendet, um die Hadoop-Umgebung mit der `hadoop-env`-Klassifizierung festzulegen:

```
[
  {
    "classification": "hadoop-env",
    "properties": {},
    "configurations": [
      {
        "classification": "export",
        "properties": {
          "YARN_PROXYSERVER_HEAPSIZE": "2396"
        }
      }
    ]
  }
]
```

Nachfolgend ist das entsprechende Pipeline-Definitionsobjekt mit dieser Konfiguration:

```
{
  "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"
    },
  ]
}
```

Im folgenden Beispiel wird eine HIVE-spezifische Eigenschaft für einen EMR-Cluster geändert:

```
{
    "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"
        }
    ]
}
```

Syntax

Dieses Objekt enthält die folgenden Felder.

Pflichtfelder	Beschreibung	Slot-Typ
Klassifizierung	Klassifizierung für die Konfiguration.	String

Optionale Felder	Beschreibung	Slot-Typ
Konfiguration	Unterkonfiguration für diese Konfiguration.	Referenzobjekt, z. B. „configuration“: {"ref“:“ Id "} myEmrConf iguration
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, z. B. „parent“: {"ref“:“ myBaseObject Id "}

Optionale Felder	Beschreibung	Slot-Typ
property	Konfigurationseigenschaft	Referenzobjekt, z. B. „Eigenschaft“: {"ref“:“ myPropertyId „}

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde.	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts	String
@pipelineId	Id der Pipeline, zu der dieses Objekt gehört	String
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen	String

Weitere Informationen finden Sie unter:

- EmrCluster
- Eigenschaft
- Amazon EMR-Versionshinweise

Eigenschaft

Eine einzelne Schlüssel-Wert-Eigenschaft zur Verwendung mit einem Objekt EmrConfiguration .

Beispiel

Die folgende Pipeline-Definition zeigt ein EmrConfiguration Objekt und die entsprechenden Eigenschaftsobjekte zum Starten eines: EmrCluster

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      },
      {
        "ref": "fs-s3-block-size"
      }
      ]
    },
    {
      "name": "io-file-buffer-size",
      "id": "io-file-buffer-size",
      "type": "Property",
      "key": "io.file.buffer.size",
      "value": "4096"
    },
    {
      "name": "fs-s3-block-size",
      "id": "fs-s3-block-size",
      "type": "Property",
      "key": "fs.s3.block.size",
      "value": "67108864"
```

```
    }
  ]
}
```

Syntax

Dieses Objekt enthält die folgenden Felder.

Pflichtfelder	Beschreibung	Slot-Typ
Schlüssel	Schlüssel	Zeichenfolge
Wert	Wert	String

Optionale Felder	Beschreibung	Slot-Typ
übergeordneter	Übergeordnetes Objekt des aktuellen Objekts, aus dem Slots übernommen werden.	Referenzobjekt, zum Beispiel „parent“: {"ref“:“ myBaseObject Id "}

Laufzeitfelder	Beschreibung	Slot-Typ
@Version	Pipeline-Version, mit der das Objekt erstellt wurde	String

Systemfelder	Beschreibung	Slot-Typ
@error	Fehler mit einer Beschreibung des falsch formatierten Objekts.	String
@pipelineId	ID der Pipeline, zu der dieses Objekt gehört.	String

Systemfelder	Beschreibung	Slot-Typ
@sphere	Die Kugel eines Objekts bezeichnet seinen Platz im Lebenszyklus: Komponentenobjekte ergeben Instance-Objekte, die Versuchsobjekte ausführen.	String

Weitere Informationen finden Sie unter:

- EmrCluster
- EmrConfiguration
- Amazon EMR-Versionshinweise

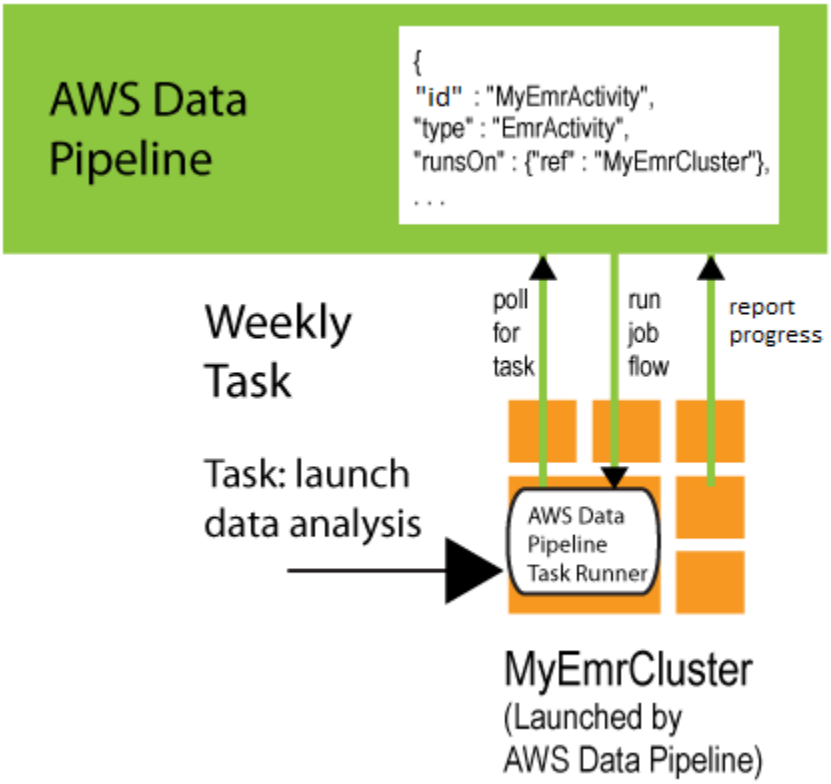
Arbeiten mit Task Runner

Task Runner ist eine Task-Agent-Anwendung, dieAWS Data Pipeline nach geplanten Aufgaben abfragt und diese auf Amazon EC2 EC2-Instances, Amazon EMR-Clustern oder anderen Rechenressourcen ausführt und dabei den Status meldet. Je nach Anwendung können Sie:

- ErlaubenAWS Data Pipeline Sie, eine oder mehrere Task Runner-Anwendungen für Sie zu installieren und zu verwalten. Wenn eine Pipeline aktiviert ist, wird der Standard`Ec2Instance` oder das`EmrCluster` Objekt, auf das ein Aktivitätsfeld „RunsOn“ verweist, automatisch erstellt. AWS Data Pipelinekümmert sich um die Installation von Task Runner auf einer EC2-Instance oder auf dem Master-Knoten eines EMR-Clusters. In diesem Muster kann AWS Data Pipeline den Großteil der Instance- oder Clusterverwaltung für Sie erledigen.
- Führen Sie alle oder Teile einer Pipeline für von Ihnen verwaltete Ressourcen aus. Zu den potenziellen Ressourcen gehören eine Amazon EC2 EC2-Instance mit langer Laufzeit, ein Amazon EMR-Cluster oder ein physischer Server. Sie können einen Task-Runner (der entweder Task Runner oder ein von Ihnen selbst entwickelter benutzerdefinierter Task-Agent sein kann) fast überall installieren, sofern er mit demAWS Data Pipeline Webservice kommunizieren kann. In diesem Muster übernehmen Sie fast die vollständige Kontrolle darüber, welche Ressourcen verwendet und wie sie verwaltet werden, und Sie müssen Task Runner manuell installieren und konfigurieren. Verwenden Sie dazu die Verfahren in diesem Abschnitt, wie in Ausführen von Arbeiten an vorhandenen Ressourcen mithilfe von Task Runner beschrieben.

Task Runner fürAWS Data Pipeline verwaltete Ressourcen

Wenn eine Ressource gestartet und verwaltet wirdAWS Data Pipeline, installiert der Webdienst automatisch Task Runner auf dieser Ressource, um Aufgaben in der Pipeline zu verarbeiten. Sie geben eine Rechenressource (entweder eine Amazon EC2 EC2-Instance oder ein Amazon EMR-Cluster) für das`runsOn` Feld eines Aktivitätsobjekts an. Wenn diese RessourceAWS Data Pipeline gestartet wird, installiert sie Task Runner auf dieser Ressource und konfiguriert sie so, dass sie alle Aktivitätsobjekte verarbeitet, deren`runsOn` Feld auf diese Ressource gesetzt ist. Wenn die RessourceAWS Data Pipeline beendet wird, werden die Task Runner-Protokolle an einem Amazon S3 S3-Standort veröffentlicht, bevor sie heruntergefahren wird.



Wenn Sie beispielsweise in einer Pipeline die `EmrActivity` verwenden und im Feld `runsOn` eine `EmrCluster`-Ressource angeben. Wenn diese AktivitätAWS Data Pipeline verarbeitet wird, wird ein Amazon EMR-Cluster gestartet und Task Runner auf dem Master-Knoten installiert. Dieser Task Runner verarbeitet dann die Aufgaben für Aktivitäten, deren`runsOn` Feld auf dieses`EmrCluster` Objekt gesetzt ist. Der folgende Ausschnitt aus einer Pipeline-Definition zeigt diese Beziehung zwischen den beiden Objekten.

```
{
  "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"
}
```

Informationen und Beispiele für die Ausführung dieser Aktivitäten finden Sie unter EmrActivity.

Wenn Sie mehrereAWS Data Pipeline verwaltete Ressourcen in einer Pipeline haben, ist Task Runner auf jeder dieser Ressourcen installiert, und alle fragenAWS Data Pipeline nach zu verarbeitenden Aufgaben ab.

Ausführen von Arbeiten an vorhandenen Ressourcen mithilfe von Task Runner

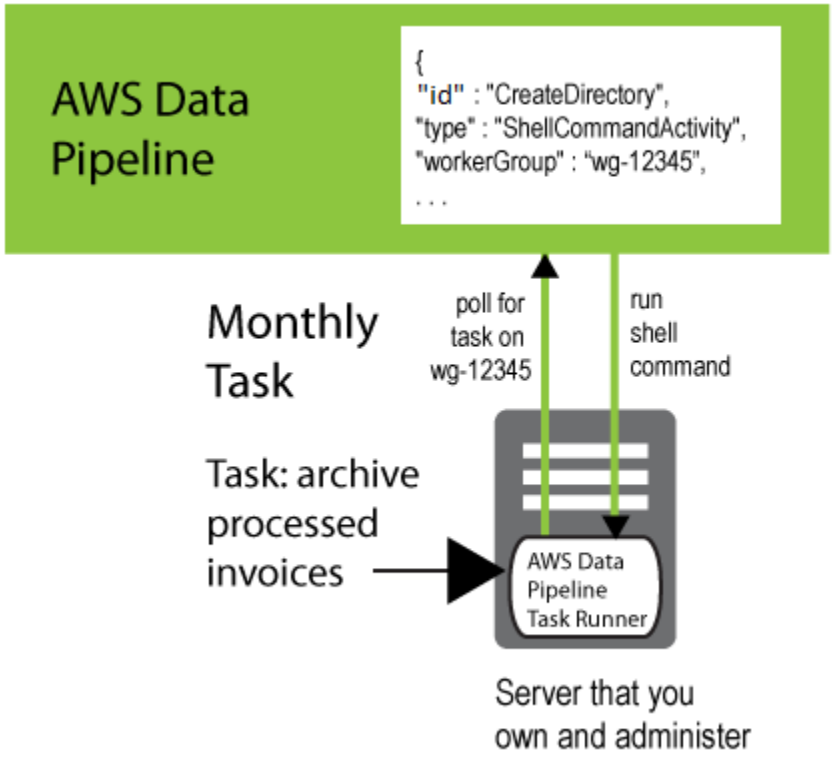
Sie können Task Runner auf von Ihnen verwalteten Rechenressourcen installieren, z. B. einer Amazon EC2 EC2-Instance oder einem physischen Server oder einer Workstation. Task Runner kann überall auf jeder kompatiblen Hardware oder jedem kompatiblen Betriebssystem installiert werden, sofern es mit demAWS Data Pipeline Webdienst kommunizieren kann.

Dieser Ansatz kann nützlich sein, wenn Sie beispielsweise AWS Data Pipeline verwenden möchten; um Daten zu verarbeiten, die in der Firewall Ihres Unternehmens gespeichert sind. Durch die Installation von Task Runner auf einem Server im lokalen Netzwerk können Sie sicher auf die lokale Datenbank zugreifen und dann abfragen,AWS Data Pipeline ob die nächste Aufgabe ausgeführt werden soll. Wenn die VerarbeitungAWS Data Pipeline beendet oder die Pipeline gelöscht wird, läuft die Task Runner-Instanz weiterhin auf Ihrer Rechenressource, bis Sie sie manuell herunterfahren. Die Task Runner-Protokolle bleiben bestehen, nachdem die Pipeline-Ausführung abgeschlossen ist.

Um Task Runner auf einer Ressource zu verwenden, die Sie verwalten, müssen Sie zuerst Task Runner herunterladen und dann mithilfe der Verfahren in diesem Abschnitt auf Ihrer Rechenressource installieren.

> **Note**
>
> Sie können Task Runner nur unter Linux, UNIX oder macOS installieren. Task Runner wird unter dem Windows-Betriebssystem nicht unterstützt.
> Um Task Runner 2.0 verwenden zu können, ist die Java-Mindestversion 1.7 erforderlich.

Um einen von Ihnen installierten Task Runner mit den Pipeline-Aktivitäten zu verbinden, die er verarbeiten soll, fügen Sie dem Objekt ein`workerGroup` Feld hinzu und konfigurieren Sie Task Runner so, dass er nach diesem Arbeitsgruppenwert abfragt. Dazu übergeben Sie die Workergroup-Zeichenfolge als Parameter (z. B.`--workerGroup=wg-12345`), wenn Sie die Task Runner-JAR-Datei ausführen.




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

Task Runner wird installiert

In diesem Abschnitt werden die Installation und Konfiguration von Task Runner und die zugehörigen Voraussetzungen erläutert. Die Installation ist ein einfacher manueller Prozess.

Um Task Runner zu installieren

1. Task Runner benötigt die Java-Versionen 1.6 oder 1.8. Verwenden Sie den folgenden Befehl, um festzustellen, ob Java installiert ist und welche Version ausgeführt wird:

   ```
   java -version
   ```

   Wenn Sie Java 1.6 oder 1.8 nicht auf Ihrem Computer installiert haben, laden Sie eine dieser Versionen von http://www.oracle.com/technetwork/java/index.html herunter. Laden Sie Java herunter und installieren Sie es und fahren Sie mit dem nächsten Schritt fort.
2. Laden Sie es`TaskRunner-1.0.jar` von https://s3.amazonaws.com/datapipeline-us-east -1/us-east-1/software/latest/TaskRunner/TaskRunner-1.0.jar herunter und kopieren Sie es dann in einen Ordner auf der Zielcomputerressource. Für Amazon EMR-Cluster, die`EmrActivity` Aufgaben ausführen, installieren Sie Task Runner auf dem Master-Knoten des Clusters.
3. Wenn Sie Task Runner verwenden, um eine Verbindung zumAWS Data Pipeline Webservice herzustellen, um Ihre Befehle zu verarbeiten, benötigen Benutzer programmatischen Zugriff auf eine Rolle, die über Berechtigungen zum Erstellen oder Verwalten von Datenpipelines verfügt. Weitere Informationen finden Sie unter Erteilen programmgesteuerten Zugriffs.
4. Task Runner stellt über HTTPS eine Verbindung zumAWS Data Pipeline Webdienst her. Wenn Sie eine AWS-Ressource verwenden, stellen Sie sicher, dass HTTPS in der entsprechenden Routing-Tabelle und in der Subnetz-ACL aktiviert ist. Wenn Sie eine Firewall oder einen Proxy verwenden, stellen Sie sicher, dass der Port 443 geöffnet ist.

(Optional) Task Runner-Zugriff auf Amazon RDS gewähren

Mit Amazon RDS können Sie den Zugriff auf Ihre DB-Instances mithilfe von Datenbank-Sicherheitsgruppen (DB-Sicherheitsgruppen) steuern. DB-Sicherheitsgruppen funktionieren wie eine Firewall. Sie steuern den Netzwerkzugriff auf die DB-Instance. Der Netzwerkzugriff auf Ihre DB-Instances ist standardmäßig deaktiviert. Sie müssen Ihre DB-Sicherheitsgruppen ändern, damit Task Runner auf Ihre Amazon RDS-Instances zugreifen kann. Task Runner erhält Amazon RDS-Zugriff

von der Instance, auf der es ausgeführt wird. Die Konten und Sicherheitsgruppen, die Sie zu Ihrer Amazon RDS-Instance hinzufügen, hängen also davon ab, wo Sie Task Runner installieren.

Um den Zugriff auf Task Runner in EC2-Classic Runner zu gewähren

1. Öffnen Sie die Amazon RDS-Konsole.
2. Klicken Sie im Navigationsbereich auf Instances (DB-Instances) und wählen Sie anschließend Ihre DB-Instance aus.
3. Wählen Sie unter Sicherheit und Netzwerk die Sicherheitsgruppe aus, die die Seite Sicherheitsgruppen mit dieser ausgewählten DB-Sicherheitsgruppe öffnet. Klicken Sie auf das Detailsymbol für die DB-Sicherheitsgruppe.
4. Legen Sie unter Sicherheitsgruppendetails eine Regel mit dem entsprechenden Verbindungstyp und Details an. Diese Felder hängen davon ab, wo Task Runner ausgeführt wird, wie hier beschrieben:

   - `Ec2Resource`
     - Verbindungstyp: `EC2 Security Group`

       Details: *`my-security-group-name`*(der Name der Sicherheitsgruppe, die Sie für die EC2-Instance erstellt haben)
   - `EmrResource`
     - Verbindungstyp: `EC2 Security Group`

       Details: `ElasticMapReduce-master`
     - Verbindungstyp: `EC2 Security Group`

       Details: `ElasticMapReduce-slave`
   - Ihre lokale Umgebung (lokal)
     - Verbindungstyp: `CIDR/IP`:

       Details: *`my-ip-address`*(die IP-Adresse Ihres Computers oder der IP-Adressbereich Ihres Netzwerks, falls sich Ihr Computer hinter einer Firewall befindet)
5. Klicken Sie auf Add (Hinzufügen).

Um Zugriff auf Task Runner in EC2-VPC zu gewähren

1. Öffnen Sie die Amazon RDS-Konsole.

2. Wählen Sie im Navigationsbereich Instances aus.
3. Klicken Sie auf das Detailsymbol für die DB-Sicherheitsgruppe. Öffnen Sie unter Sicherheit und Netzwerk den Link zur Sicherheitsgruppe, der Sie zur Amazon EC2 EC2-Konsole führt. Wenn Sie das alte Konsolendesign für Sicherheitsgruppen verwenden, wechseln Sie zum neuen Konsolendesign, indem Sie auf das Symbol klicken, das oben auf der Konsolenseite angezeigt wird.
4. Wählen Sie auf der Registerkarte Inbound die Option Edit, Add Rule. Geben Sie den Datenbankanschluss an, den Sie beim Starten der DB-Instance verwendet haben. Die Quelle hängt davon ab, wo Task Runner ausgeführt wird, wie hier beschrieben:

   - `Ec2Resource`
     - *`my-security-group-id`*(die ID der Sicherheitsgruppe, die Sie für die EC2-Instance erstellt haben)
   - `EmrResource`
     - *`master-security-group-id`*(die ID der`ElasticMapReduce-master` Sicherheitsgruppe)
     - *`slave-security-group-id`*(die ID der`ElasticMapReduce-slave` Sicherheitsgruppe)
   - Ihre lokale Umgebung (lokal)
     - *`ip-address`* (die IP-Adresse Ihres Computers oder den IP-Adressbereich Ihres Netzwerks, falls sich Ihr Computer hinter einer Firewall befindet)
5. Klicken Sie auf Speichern.

Task Runner starten

Starten Sie Task Runner in einem neuen Befehlszeilenfenster, das auf das Verzeichnis eingestellt ist, in dem Sie Task Runner installiert haben, mit dem folgenden Befehl.

```
java -jar TaskRunner-1.0.jar --config ~/credentials.json --workerGroup=myWorkerGroup --
region=MyRegion --logUri=s3://mybucket/foldername
```

Die Option `--config` zeigt auf Ihre Anmeldedaten-Datei.

Die Option `--workerGroup` gibt den Namen Ihrer Auftragnehmergruppe an, der für die zu verarbeitenden Aufgaben in Ihrer Pipeline den gleichen Wert haben muss.

Die Option `--region` gibt den Servicebereich an, von dem aus Aufgaben ausgeführt werden sollen.

`--logUri`Diese Option wird verwendet, um Ihre komprimierten Protokolle an einen Speicherort in Amazon S3 zu übertragen.

Wenn Task Runner aktiv ist, druckt es den Pfad, in den die Protokolldateien geschrieben werden, im Terminalfenster. Im Folgenden wird ein Beispiel gezeigt.

```
Logging to /Computer_Name/.../output/logs
```

Task-Runner sollte von Ihrer Anmelde-Shell getrennt ausgeführt werden. Wenn Sie eine Terminalanwendung verwenden, um eine Verbindung zu Ihrem Computer herzustellen, müssen Sie möglicherweise ein Dienstprogramm wie nohup oder screen verwenden, um zu verhindern, dass die Task-Runner-Anwendung beendet wird, wenn Sie sich abmelden. Weitere Informationen zu diesen Befehlszeilenoptionen finden Sie unter Task-Runner-Konfigurationsoptionen.

Überprüfung der Task-Runner-Protokollierung

Der einfachste Weg, um zu überprüfen, ob Task Runner funktioniert, besteht darin, zu überprüfen, ob er Protokolldateien schreibt. Task Runner schreibt stündliche Protokolldateien in das Verzeichnis,`output/logs`, unter dem Verzeichnis, in dem Task Runner installiert ist. Der Dateiname ist `Task Runner.log.YYYY-MM-DD-HH`, wobei HH von 00 bis 23 in UDT läuft. Um Speicherplatz zu sparen, werden alle Protokolldateien, die älter als acht Stunden sind, mit GZip komprimiert.

Task Runner-Threads und Vorbedingungen

Task Runner verwendet für jede Aufgabe, Aktivität und Vorbedingung einen Threadpool. Die Standardeinstellung für `--tasks` ist 2. Das bedeutet, dass zwei Threads aus dem Aufgabenpool zugewiesen werden und jeder Thread den AWS Data Pipeline-Service nach neuen Aufgaben abfragen wird. Daher ist `--tasks` ein Attribut für die Leistungsoptimierung, mit dem der Pipeline-Durchsatz optimiert werden kann.

In Task Runner findet die Logik für Wiederholungsversuche in der Pipeline für die Vorbedingungen statt. Zwei Vorbedingungs-Threads fragen AWS Data Pipeline über Vorbedingungsobjekte ab. Task Runner berücksichtigt die Vorbedingungsobjekte RetryDelay und PreconditionTimeout, die Sie für Vorbedingungen definieren.

In vielen Fällen kann das Reduzieren des Timeouts für Abfragen und die Anzahl der Wiederholungen dazu beitragen, die Leistung Ihrer Anwendung zu verbessern. In ähnlicher Weise müssen

Anwendungen mit Langzeitvorbedingungen die Werte für Timeout und Wiederholung erhöhen. Weitere Informationen zu Vorbedingungsobjekten finden Sie unter Vorbedingungen.

Task-Runner-Konfigurationsoptionen

Dies sind die Konfigurationsoptionen, die über die Befehlszeile verfügbar sind, wenn Sie Task Runner starten.

Befehlszeilen-Parameter	Beschreibung
`--help`	Befehlszeilenhilfe. Beispiel: `Java -jar TaskRunner-1.0.jar --help`
`--config`	Pfad und Dateiname Ihrer Datei `credentials.json` .
`--accessId`	IhreAWS Zugangsschlüssel-ID, die Task Runner bei Anfragen verwenden kann. Die`--secretKey` Optionen`--accessID` und stellen eine Alternative zur Verwendung einer JSON-Datei mit Anmeldeinformationen dar. Wenn auch eine `credentials.json` -Datei vorgesehen ist, haben die Optionen `--accessID` und `--secretKey` Vorrang.
`--secretKey`	IhrAWS geheimer Schlüssel, den Task Runner bei Anfragen verwenden kann. Weitere Informationen finden Sie unter `--accessID` .
`--endpoint`	Ein Endpunkt ist eine URL, die als Eintrittspunkt für einen Webservice fungiert. Der AWS Data Pipeline-Service-Endpunkt in der Region, in der Sie Anfragen stellen. Optional. Im Allgemeinen reicht es aus, eine Region anzugeben, und Sie müssen den Endpunkt nicht festlegen . Eine Auflistung der AWS Data Pipeline-Regionen und -Endpunkte finden Sie unter

Befehlszeilen-Parameter	Beschreibung
	AWS Data Pipeline-Regionen und -Endpunkte im Allgemeine AWS-Referenz.
`--workerGroup`	Der Name der Workergruppe, für die Task Runner Aufträge abruft. Erforderlich. Wenn Task Runner den Webservice abfragt, verwendet er die von Ihnen angegeben en Anmeldeinformationen und den Wert von,`workerGroup` um auszuwählen, welche (falls vorhanden) Aufgaben abgerufen werden sollen. Sie können jeden Namen verwenden , der für Sie von Bedeutung ist. Die einzige Voraussetzung ist, dass die Zeichenfolge zwischen dem Task Runner und seinen entsprechenden Pipeline-Aktivitäten übereinst immen muss. Der Name der Auftragne hmergruppe ist an eine Region gebunden. Selbst wenn es in anderen Regionen identisch e Arbeitsgruppennamen gibt, ruft Task Runner immer Aufgaben aus der Region ab, in der angegeben ist`--region`.
`--taskrunnerId`	Die ID des Task-Runners für die Berichter stellung zum Fortschritt. Optional.
`--output`	Das Task Runner-Verzeichnis für Protokoll ausgabedateien. Optional. Protokolldateien werden in einem lokalen Verzeichnis gespeiche rt, bis sie an Amazon S3 übertragen werden. Diese Option überschreibt das Standard-Verzeichnis.

Befehlszeilen-Parameter	Beschreibung
`--region`	Die -Region, die verwendet werden soll. Optional, aber es wird empfohlen, die Region immer festzulegen. Wenn Sie die Region nicht angeben, ruft Task Runner Aufgaben aus der Standard-Serviceregion ab`us-east-1` . Andere unterstützte Regionen sind: `eu-west-1` , `ap-northeast-1` , `ap-southeast-2` , `us-west-2` .
`--logUri`	Der Amazon S3 S3-Zielpfad für Task Runner, um Protokolldateien auf jede Stunde zu sichern. Wenn Task Runner beendet wird, werden aktive Protokolle im lokalen Verzeichnis in den Amazon S3 S3-Zielordner verschoben.
`--proxyHost`	Der Host des Proxys, der von Task Runner-Clients für die Verbindung zu AWS-Services verwendet wird.
`--proxyPort`	Port des Proxy-Hosts, der von Task Runner-Clients für die Verbindung zu AWS-Services verwendet wird.
`--proxyUsername`	Der Benutzername für den Proxy
`--proxyPassword`	Das Passwort für den Proxy.
`--proxyDomain`	Der Windows-Domänenname für NTLM Proxy.
`--proxyWorkstation`	Der Windows-Arbeitsstationsname für NTLM Proxy.

Task-Runner mit einem Proxy verwenden

Wenn Sie einen Proxy-Host verwenden, können Sie beim Aufrufen von Task-Runner entweder seine Konfiguration angeben oder die Umgebungsvariable HTTPS_PROXY festlegen. Die mit Task-Runner verwendete Umgebungsvariable akzeptiert dieselbe Konfiguration, die für die AWS-Befehlszeilenschnittstelleverwendet wird.

Task Runner und benutzerdefinierte AMIs

Wenn Sie ein`Ec2Resource` Objekt für Ihre Pipeline angeben,AWS Data Pipeline wird eine EC2-Instance für Sie erstellt und dabei ein AMI verwendet, das Task Runner für Sie installiert und konfiguriert. In diesem Fall ist ein PV-kompatibler Instancetyp erforderlich. Alternativ können Sie mit Task Runner ein benutzerdefiniertes AMI erstellen und dann die ID dieses AMI mithilfe des`imageId` Feldes des`Ec2Resource` Objekts angeben. Weitere Informationen finden Sie unter Ec2Resource.

Ein benutzerdefiniertes AMI muss die folgenden Anforderungen erfüllen,AWS Data Pipeline um es erfolgreich für Task Runner verwenden zu können:

- Erstellen Sie das AMI in derselben Region, in der die Instances ausgeführt werden. Weitere Informationen finden Sie unter Creating Your Own AMI im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.
- Stellen Sie sicher, dass der Virtualisierungstyp des AMI vom Instancetyp unterstützt wird, den Sie verwenden möchten. Beispielsweise benötigen die Instancetypen I2 und G2 einen HVM AMI, und die Instancetypen T1, C1, M1 und M2 erfordern einen PV AMI. Weitere Informationen finden Sie unter Linux-AMI-Virtualisierungstypen im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.
- Installieren Sie die folgenden Software:
  - Linux
  - Bash
  - wget
  - unzip
  - Java 1.6 oder 1.8
  - cloud-init
- Erstellen und konfigurieren Sie einen Benutzer namens`ec2-user`.

Fehlerbehebung

Das häufigste Symptom bei einem Problem mit AWS Data Pipeline besteht darin, dass eine Pipeline nicht ausgeführt wird. Sie können die in der Konsole und der Befehlszeile angezeigten Informationen verwenden, um das Problem zu bestimmen und nach einer Lösung zu suchen.

Inhalt

Suchen von Fehlern in Pipelines

Die AWS Data Pipeline-Konsole ist ein praktisches Hilfsmittel zur visuellen Überwachung des Status der Pipelines und zum einfachen Lokalisieren von Fehlerinformationen in Zusammenhang mit fehlgeschlagenen oder unvollständigen Pipeline-Ausführungen.

So suchen Sie nach Fehlerinformationen zu fehlgeschlagenen oder unvollständigen Pipeline-Ausführungen.

1. Wenn auf der Seite List Pipelines in der Spalte Status einer Pipeline-Instance ein anderer Status als FINISHED angezeigt wird, wartet die betreffende Pipeline auf die Erfüllung einer Vorbedingung oder es ist ein Problem mit der Pipeline aufgetreten.
2. Suchen Sie auf der Seite List Pipelines (Pipelines auflisten) die Instance-Pipeline und klicken Sie auf das Dreieck links daneben, um den Detailbereich zu erweitern.
3. Klicken Sie unten in diesem Feld auf View execution details (Ausführungsdetails anzeigen); das Feld Instance summary (Instance-Zusammenfassung) wird geöffnet, um die Details der ausgewählten Instance anzuzeigen.
4. Klicken Sie im Bereich Instance summary (Instance-Zusammenfassung) auf das Dreieck neben der Instance, um zusätzliche Details zur Instance anzuzeigen, und wählen Sie Details, More.... Wenn der Status der ausgewählten Instance FAILED ist, enthält das Detailfeld Einträge für die Fehlermeldung, das `errorStackTrace` und weitere Informationen. Sie können diese Informationen in einer Datei speichern. Wählen Sie OK.

5. Klicken Sie im Bereich Instance summary (Instance-Zusammenfassung) auf Attempts (Versuche), um Details für jede Versuchszeile anzuzeigen.
6. Um eine Aktion mit der unvollständigen oder fehlgeschlagenen Instance durchzuführen, aktivieren Sie das Kontrollkästchen neben der Instance. Hiermit aktivieren Sie die Aktionen. Wählen Sie dann eine Aktion (`Rerun|Cancel|Mark Finished`).

Identifizierung des Amazon EMR-Clusters, der Ihre Pipeline bedient

Wenn ein `EMRCluster` oder `EMRActivity` fehlschlägt und die von der AWS Data Pipeline Konsole bereitgestellten Fehlerinformationen unklar sind, können Sie den Amazon EMR-Cluster, der Ihre Pipeline bereitstellt, mithilfe der Amazon EMR-Konsole identifizieren. Dies hilft Ihnen, die von Amazon EMR bereitgestellten Protokolle zu finden, um weitere Informationen zu auftretenden Fehlern zu erhalten.

Um detailliertere Amazon EMR-Fehlerinformationen zu sehen

1. Klicken Sie in der AWS Data Pipeline-Konsole auf das Dreieck neben der Pipeline-Instance, um die Instance-Details zu erweitern.
2. Klicken Sie auf View execution details (Ausführungsdetails anzeigen) und dann auf das Dreieck neben der Komponente.
3. Klicken Sie in der Spalte Details auf More... (Mehr…). Der Informationsbildschirm wird geöffnet und zeigt eine Liste der Details der Komponente an. Suchen und kopieren Sie den instanceParent-Wert auf dem Bildschirm, z. B.: `@EmrActivityId_xiFDD_2017-09-30T21:40:13`
4. Navigieren Sie zur Amazon EMR-Konsole, suchen Sie nach einem Cluster mit dem passenden InstanceParent-Wert im Namen, und wählen Sie dann Debug.

   > **Note**
   >
   > Damit die Debug-Schaltfläche funktioniert, muss Ihre Pipeline-Definition die EmrActivity `enableDebugging` Option auf `true` und die `EmrLogUri` Option auf einen gültigen Pfad gesetzt haben.

5. Da Sie nun wissen, welcher Amazon EMR-Cluster den Fehler enthält, der Ihren Pipeline-Ausfall verursacht hat, befolgen Sie die Tipps zur Fehlerbehebung im Amazon EMR Developer Guide.

Interpretieren der Pipeline-Statusdetails

Die verschiedenen in der AWS Data Pipeline-Konsole und -Befehlszeilenschnittstelle (CLI) angezeigten Statuswerte geben den Zustand einer Pipeline und ihrer Komponenten an. Der Pipeline-Status ist vereinfacht ausgedrückt ein Überblick über eine Pipeline. Wenn Sie weitere Informationen benötigen, zeigen Sie den Status der einzelnen Pipeline-Komponenten an. Sie können dazu in der Konsole auf die Pipeline und deren Komponenten klicken oder die Details der Pipeline-Komponenten über die CLI abrufen.

Statuscodes

`ACTIVATING`

Die Komponente oder Ressource wird gestartet, z. B. eine EC2-Instance.

`CANCELED`

Die Komponente wurde von einem Benutzer oder AWS Data Pipeline bevor sie ausgeführt werden konnte, storniert. Dies kann automatisch geschehen, wenn ein Fehler in einer anderen Komponente oder Ressource auftritt, von der diese Komponente abhängt.

`CASCADE_FAILED`

Die Komponente oder Ressource wurde aufgrund eines Kaskadenausfalls aus einer ihrer Abhängigkeiten storniert, aber die Komponente war wahrscheinlich nicht die ursprüngliche Ursache des Fehlers.

`DEACTIVATING`

Die Pipeline wird deaktiviert.

`FAILED`

Bei der Komponente oder Ressource ist ein Fehler aufgetreten und sie funktioniert nicht mehr. Wenn eine Komponente oder Ressource ausfällt, kann dies zu Abbrüchen und Ausfällen führen, die sich auf andere Komponenten auswirken, die von ihr abhängen.

`FINISHED`

Die Komponente hat ihre zugewiesene Arbeit abgeschlossen.

`INACTIVE`

Die Pipeline wurde deaktiviert.

`PAUSED`

Die Komponente wurde angehalten und führt derzeit ihre Arbeit nicht aus.

`PENDING`

Die Pipeline ist bereit, zum ersten Mal aktiviert zu werden.

`RUNNING`

Die Ressource läuft und ist bereit, Arbeit anzunehmen.

`SCHEDULED`

Die Ausführung der Ressource ist geplant.

`SHUTTING_DOWN`

Die Ressource wird heruntergefahren, nachdem sie ihre Arbeit erfolgreich abgeschlossen hat.

`SKIPPED`

Die Komponente hat Ausführungsintervalle übersprungen, nachdem die Pipeline aktiviert wurde. Dabei wurde ein Zeitstempel verwendet, der nach dem aktuellen Zeitplan liegt.

`TIMEDOUT`

Die Ressource hat den `terminateAfter` Schwellenwert überschritten und wurde angehaltenAWS Data Pipeline. Nachdem die Ressource diesen Status erreicht hat, AWS Data Pipeline werden die `retryTimeout` Werte `actionOnResourceFailureretryDelay`, und für diese Ressource ignoriert. Dieser Status gilt nur für Ressourcen.

`VALIDATING`

Die Pipeline-Definition wird von AWS Data Pipeline validiert.

`WAITING_FOR_RUNNER`

Die Komponente wartet darauf, dass ihr Worker-Client ein Arbeitselement abruft. Die Beziehung zwischen Komponente und Mitarbeiter und Kunde wird durch die `runsOn` `workerGroup` Oder-Felder gesteuert, die von dieser Komponente definiert werden.

`WAITING_ON_DEPENDENCIES`

Die Komponente überprüft, ob die standardmäßigen und vom Benutzer konfigurierten Vorbedingungen erfüllt sind, bevor sie ihre Arbeit ausführt.

Lokalisieren von Fehlerprotokollen

In diesem Abschnitt wird beschrieben, wie Sie die verschiedenen AWS Data Pipeline-Protokolle lokalisieren, die zur Ermittlung der Ursachen bestimmter Fehler und Probleme verwendet werden können.

Pipeline-Protokolle

Es wird empfohlen, Pipelines so zu konfigurieren, dass Protokolldateien an einem persistenten Speicherort erstellt werden, wie im folgenden Beispiel, in dem Sie das `pipelineLogUri` Feld im `Default` Objekt einer Pipeline verwenden, damit alle Pipeline-Komponenten standardmäßig einen Amazon S3-Protokollspeicherort verwenden (Sie können dies überschreiben, indem Sie einen Protokollspeicherort in einer bestimmten Pipeline-Komponente konfigurieren).

> **Note**
>
> Task Runner speichert seine Protokolle standardmäßig an einem anderen Ort, der möglicherweise nicht verfügbar ist, wenn die Pipeline abgeschlossen ist und die Instanz, auf der Task Runner ausgeführt wird, beendet wird. Weitere Informationen finden Sie unter Überprüfung der Task-Runner-Protokollierung.

Um den Protokollspeicherort über die AWS Data Pipeline-CLI in der JSON-Datei einer Pipeline festlegen, fügen Sie am Anfang der Datei folgenden Code ein:

```
{ "objects": [
{
  "id":"Default",
  "pipelineLogUri":"s3://mys3bucket/error_logs"
},
...
```

Nachdem Sie ein Pipeline-Protokollverzeichnis konfiguriert haben, erstellt Task Runner eine Kopie der Protokolle in Ihrem Verzeichnis mit derselben Formatierung und denselben Dateinamen, die im vorherigen Abschnitt über Task Runner-Protokolle beschrieben wurden.

Hadoop Job- und Amazon EMR-Schrittprotokolle

Bei jeder Hadoop-basierten Aktivität wieHadoopActivity,HiveActivity, oder PigActivity Sie können die Hadoop-Jobprotokolle an der im Laufzeitslot zurückgegebenen Position einsehen. hadoopJobLog EmrActivityverfügt über eigene Protokollierungsfunktionen und diese Protokolle werden an dem von Amazon EMR ausgewählten Ort gespeichert und vom Runtime-Slot zurückgegeben. emrStepLog Weitere Informationen finden Sie unter Logdateien anzeigen im Amazon EMR Developer Guide.

Beheben typischer Probleme

In diesem Thema werden die Symptome verschiedener AWS Data Pipeline-Probleme und die empfohlenen Schritte zu deren Behebung beschrieben.

Inhalt

Pipeline bleibt im Status PENDING

Wenn eine Pipeline dauerhaft im Status PENDING bleibt, weist dies darauf hin, dass sie noch nicht aktiviert wurde oder die Aktivierung aufgrund eines Fehlers in der Pipeline-Definition fehlgeschlagen

ist. Vergewissern Sie sich, dass beim Übertragen der Pipeline über die AWS Data Pipeline-CLI oder beim Versuch, die Pipeline in der AWS Data Pipeline-Konsole zu speichern oder zu aktivieren, keine Fehler aufgetreten sind. Überprüfen Sie außerdem, ob die Definition der Pipeline gültig ist.

So zeigen Sie die Pipeline-Definition unter Verwendung der Befehlszeile an:

```
aws datapipeline --get-pipeline-definition --pipeline-id df-EXAMPLE_PIPELINE_ID
```

Überzeugen Sie sich, dass die Definition vollständig ist und keine Syntaxfehler enthält. Achten Sie insbesondere darauf, dass keine schließenden Klammern, erforderlichen Kommas oder Verweise fehlen. Wir empfehlen, einen Texteditor zu verwenden, der die Syntax von JSON-Dateien visuell darstellen und validieren kann.

Pipeline-Komponente bleibt im Status WAITING_FOR_RUNNER

Wenn sich die Pipeline im Status SCHEDULED befindet und einzelne oder mehrere Aufgaben dauerhaft im Status WAITING_FOR_RUNNER bleiben, stellen Sie sicher, dass das Feld "runsOn" oder "workerGroup" dieser Aufgaben einen gültigen Wert enthält. Falls beide Werte leer sind oder fehlen, kann die betreffende Aufgabe nicht gestartet werden, da es keine Zuordnung zwischen ihr und dem Worker zur Durchführung der Aufgaben gibt. In diese Situation haben Sie zwar durchzuführende Aufgaben definiert, aber nicht festgelegt, auf welchem Computer dies geschehen soll. Stellen Sie gegebenenfalls sicher, dass der der Pipeline-Komponente zugewiesene WorkerGroup-Wert genau den gleichen Namen und die gleiche Groß- und Kleinschreibung hat wie der WorkerGroup-Wert, den Sie für Task Runner konfiguriert haben.

> **Note**
>
> Wenn Sie einen runsOn-Wert angeben und workerGroup vorhanden ist, wird workerGroup ignoriert.

Eine weitere mögliche Ursache für dieses Problem besteht darin, dass der Endpunkt und der Zugriffsschlüssel, die Task Runner zur Verfügung gestellt werden, nicht mit der AWS Data Pipeline Konsole oder dem Computer identisch sind, auf dem die AWS Data Pipeline CLI-Tools installiert sind. Möglicherweise haben Sie neue Pipelines ohne sichtbare Fehler erstellt, aber Task Runner fragt aufgrund der unterschiedlichen Anmeldeinformationen den falschen Standort ab oder fragt den richtigen Standort mit unzureichenden Berechtigungen ab, um die in der Pipeline-Definition angegebene Arbeit zu identifizieren und auszuführen.

Pipeline-Komponente bleibt im Status WAITING_ON_DEPENDENCIES

Wenn sich die Pipeline im Status `SCHEDULED` befindet und einzelne oder mehrere Aufgaben dauerhaft im Status `WAITING_ON_DEPENDENCIES` bleiben, vergewissern Sie sich, dass die Vorbedingungen erfüllt sind. Wenn die Vorbedingungen des ersten Objekts in der Logikkette nicht erfüllt sind, kann keines der Objekte, die vom ersten Objekt abhängig sind, den Status `WAITING_ON_DEPENDENCIES` verlassen.

Sehen Sie sich als Beispiel den folgenden Auszug aus einer Pipeline-Definition an. In diesem Fall hat das `InputData` Objekt die Vorbedingung „Bereit“, die angibt, dass die Daten vorhanden sein müssen, bevor das InputData Objekt vollständig ist. Wenn die Daten nicht existieren, verbleibt das InputData Objekt im `WAITING_ON_DEPENDENCIES` Status und wartet darauf, dass die im Pfadfeld angegebenen Daten verfügbar sind. Alle Objekte, die davon abhängen, bleiben InputData ebenfalls in einem `WAITING_ON_DEPENDENCIES` Zustand und warten darauf, dass das InputData Objekt den `FINISHED` Zustand erreicht.

```
{
    "id": "InputData",
    "type": "S3DataNode",
    "filePath": "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
    "schedule":{"ref":"MySchedule"},
    "precondition": "Ready"
},
{
    "id": "Ready",
    "type": "Exists"
...
```

Überprüfen Sie außerdem, ob die Objekte über die Berechtigungen zum Zugriff auf die Daten verfügen. Wenn im vorherigen Beispiel die Informationen im Feld Anmeldeinformationen nicht über Berechtigungen für den Zugriff auf die im Pfadfeld angegebenen Daten verfügten, würde das InputData Objekt in einem `WAITING_ON_DEPENDENCIES` Zustand stecken bleiben, weil es nicht auf die im Pfadfeld angegebenen Daten zugreifen kann, selbst wenn diese Daten existieren.

Es ist auch möglich, dass einer Ressource, die mit Amazon S3 kommuniziert, keine öffentliche IP-Adresse zugeordnet ist. So muss beispielsweise eine `Ec2Resource` in einem öffentlichen Subnetz über eine öffentliche IP-Adresse verfügen.

Und schließlich können Ressourceninstanzen unter bestimmten Bedingungen den Status `WAITING_ON_DEPENDENCIES` viel früher erreichen als ihre zugeordneten Aktivitäten, die zur

Ausführung geplant sind. Dies kann den Eindruck erwecken, dass die Ressource oder Aktivität fehlschlägt.

Ausführung beginnt nicht zum geplanten Zeitpunkt

Vergewissern Sie sich, dass Sie den richtigen Zeitplantyp ausgewählt haben, der festlegt, ob die Aufgabe am Anfang des Zeitplanintervalls (Cron-Zeitplantyp) oder an dessen Ende (Zeitreihen-Zeitplantyp) ausgeführt wird.

Überprüfen Sie außerdem, ob Sie die Daten in Ihren Zeitplanobjekten korrekt angegeben haben startDateTime und ob die endDateTime Werte und im UTC-Format vorliegen, wie im folgenden Beispiel:

```
{
    "id": "MySchedule",
    "startDateTime": "2012-11-12T19:30:00",
    "endDateTime":"2012-11-12T20:30:00",
    "period": "1 Hour",
    "type": "Schedule"
},
```

Pipeline-Komponenten werden in der falschen Reihenfolge ausgeführt

Möglicherweise stellen Sie fest, dass die Pipeline-Komponenten nicht in der durch die Start- und Endzeiten festgelegten Reihenfolge oder nicht in der erwarteten Abfolge ausgeführt werden. Sie müssen wissen, dass Pipeline-Komponenten gleichzeitig gestartet werden können, wenn ihre Vorbedingungen zum Ausführungszeitpunkt erfüllt sind. Anders ausgedrückt: Die Pipeline-Komponenten werden nicht standardmäßig der Reihe nach ausgeführt. Wenn Sie eine bestimmte Ausführungsreihenfolge verwenden möchten, müssen Sie diese mithilfe von Vorbedingungen und `dependsOn`-Feldern festlegen.

Stellen Sie außerdem sicher, dass das `dependsOn`-Feld einen Verweis auf die richtigen Vorbedingungs-Pipeline-Komponenten enthält und dass alle erforderlichen Zeiger zwischen den Komponenten vorhanden sind, um die gewünschte Reihenfolge zu erreichen.

EMR-Cluster schlägt mit Fehlermeldung fehl: The security token included in the request is invalid

Überprüfen Sie Ihre IAM-Rollen, Richtlinien und Vertrauensbeziehungen wie unter beschriebenIAM-Rollen für AWS Data Pipeline.

Unzureichende Berechtigungen für den Zugriff auf Ressourcen

Berechtigungen, die Sie für IAM-Rollen festlegen, bestimmen, ob Sie auf Ihre EMR-Cluster und EC2-Instances zugreifen AWS Data Pipeline können, um Ihre Pipelines auszuführen. Darüber hinaus bietet IAM das Konzept von Vertrauensbeziehungen, das die Schaffung von Ressourcen in Ihrem Namen ermöglicht. Wenn Sie beispielsweise eine Pipeline erstellen, in der eine EC2-Instance für die Ausführung eines Befehls zum Verschieben von Daten verwendet wird, kann AWS Data Pipeline diese EC2-Instance für Sie bereitstellen. Wenn Sie auf Probleme stoßen, insbesondere solche, die Ressourcen betreffen, auf die Sie manuell zugreifen können, auf die Sie jedoch AWS Data Pipeline nicht zugreifen können, überprüfen Sie Ihre IAM-Rollen, Richtlinien und Vertrauensbeziehungen, wie unter beschriebenIAM-Rollen für AWS Data Pipeline.

Statuscode: 400 Fehlercode: PipelineNotFoundException

Dieser Fehler bedeutet, dass Ihre IAM-Standardrollen möglicherweise nicht über die erforderlichen Berechtigungen verfügen, um ordnungsgemäß AWS Data Pipeline zu funktionieren. Weitere Informationen finden Sie unter IAM-Rollen für AWS Data Pipeline.

Pipeline-Erstellung führt zu einem Sicherheits-Token-Fehler

Wenn Sie versuchen, eine Pipeline zu erstellen, wird die folgende Fehlermeldung angezeigt:

Failed to create pipeline with 'pipeline_name'. Fehler: UnrecognizedClientException - Das in der Anfrage enthaltene Sicherheitstoken ist ungültig.

Pipeline-Details werden nicht in der Konsole angezeigt

Der Pipeline-Filter in der AWS Data Pipeline-Konsole wird auf das geplante Startdatum einer Pipeline angewendet, unabhängig davon, wann die Pipeline übermittelt wurde. Es ist möglich, eine neue Pipeline mit einem geplanten Anfangsdatum zu übermitteln, das in der Vergangenheit liegt. Diese Pipeline wird dann vom Standardfilter gefiltert und nicht angezeigt. Um die Details der Pipeline anzuzeigen, ändern Sie den Datumsfilter so, dass das geplante Startdatum der Pipeline innerhalb des Datumsbereichs des Filters liegt.

Error in remote runner Status Code: 404, AWS Service: Amazon S3

Dieser Fehler bedeutet, dass Task Runner nicht auf Ihre Dateien in Amazon S3 zugreifen konnte. Vergewissern Sie sich, dass folgende Bedingungen erfüllt sind:

- Die Anmeldeinformationen wurden richtig festgelegt.
- Der Amazon S3-Bucket, auf den Sie zugreifen möchten, ist vorhanden
- Sie sind berechtigt, auf den Amazon S3-Bucket zuzugreifen

Access Denied - Not Authorized to Perform Function datapipeline:

In den Task Runner-Protokollen wird möglicherweise ein Fehler angezeigt, der dem folgenden ähnelt:

- ERROR Status Code: 403
- AWS-Service: DataPipeline
- AWS-Fehlercode: AccessDenied
- AWS-Fehlermeldung: Benutzer: arn:aws:sts: :xxxxxxxxxxxx:Federated-User/I-xxxxxxxx ist nicht autorisiert, Folgendes auszuführen: datapipeline:. PollForTask

> **Note**
>
> In dieser Fehlermeldung PollForTask kann es durch Namen anderer AWS Data Pipeline Berechtigungen ersetzt werden.

Diese Fehlermeldung weist darauf hin, dass die von Ihnen angegebene IAM-Rolle zusätzliche Berechtigungen benötigt, um mit AWS Data Pipeline ihr interagieren zu können. Stellen Sie sicher, dass Ihre IAM-Rollenrichtlinie die folgenden Zeilen enthält, wobei PollForTask diese durch den Namen der Berechtigung ersetzt werden, die Sie hinzufügen möchten (verwenden Sie *, um alle Berechtigungen zu gewähren). Weitere Informationen darüber, wie Sie eine neue IAM-Rolle erstellen und eine Richtlinie darauf anwenden, finden Sie unter Verwaltung von IAM-Richtlinien im Leitfaden Using IAM.

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

Ältere Amazon EMR-AMIs erzeugen möglicherweise falsche Daten für große CSV-Dateien

Auf Amazon EMR AWS Data Pipeline verwendet AMIs vor 3.9 (3.8 und niedriger) eine benutzerdefinierte Version InputFormat zum Lesen und Schreiben von CSV-Dateien zur Verwendung mit MapReduce Jobs. Dies wird verwendet, wenn der Service Tabellen von und zu Amazon S3 bereitstellt. Dabei InputFormat wurde ein Problem entdeckt, bei dem das Lesen von Datensätzen aus großen CSV-Dateien dazu führen kann, dass Tabellen erstellt werden, die nicht korrekt kopiert werden. Dieses Problem wurde in späteren Amazon EMR-Versionen behoben. Bitte verwenden Sie Amazon EMR AMI 3.9 oder eine Amazon EMR-Version 4.0.0 oder höher.

Erhöhen der AWS Data Pipeline-Limits

Es kann gelegentlich vorkommen, dass bestimmte AWS Data Pipeline-Systemlimits überschritten werden. So können beispielsweise höchstens 20 Pipelines mit jeweils 50 Objekten erstellt werden. Falls Sie mehr Pipelines benötigen, können Sie mehrere Pipelines zusammenführen. Sie erhalten dann weniger Pipelines mit mehr Objekten in jeder. Weitere Informationen zu den Limits für AWS Data Pipeline finden Sie unter Limits für AWS Data Pipeline. Sollte diese Abhilfemaßnahme nicht zum Erfolg führen, können Sie mit dem folgenden Formular eine Kapazitätserhöhung anfordern: Erhöhen des Pipeline-Limits.

Limits für AWS Data Pipeline

Um sicherzustellen, dass für alle Benutzer ausreichend Kapazität verfügbar ist, legt AWS Data Pipeline Einschränkungen in Bezug auf die Ressourcen fest, die Sie zuweisen können, und die Rate, mit der Sie diese zuweisen können.

Inhalt

Kontolimits

Die folgenden Grenzwerte gelten für ein einzelnes AWS-Konto. Wenn Sie zusätzliche Kapazität benötigen, können Sie das Antragsformular für das Amazon Web Services Support Center verwenden, um Ihre Kapazität zu erhöhen.

Attribut	Limit	Anpassbar
Anzahl Pipelines	100	Ja
Anzahl Objekte pro Pipeline	100	Ja
Anzahl aktiver Instances pro Objekt	5	Ja
Anzahl Felder pro Objekt	50	Nein
Anzahl der UTF8-Bytes pro Feldname oder Kennung	256	Nein
Anzahl der UTF8-Bytes pro Feld	10,240	Nein

Attribut	Limit	Anpassbar
Anzahl der UTF8-Bytes pro Objekt	15.360 (einschl. Feldnamen)	Nein
Erstellungsrate einer Instance von einem Objekt	1 pro 5 Minuten	Nein
Neuversuche einer Pipeline-Aktivität	5 pro Aufgabe	Nein
Minimale Verzögerung zwischen Neuversuchen	2 Minuten	Nein
Minimales Planungsintervall	15 Minuten	Nein
Maximale Anzahl Aggregationen zu einem Objekt	32	Nein
Maximale Anzahl EC2-Instances pro Ec2Resource-Objekt	1	Nein

Limits für Webservice-Aufrufe

AWS Data Pipeline begrenzt die Rate, mit der Sie die Webservice-API aufrufen können. Diese Beschränkungen gelten auch fürAWS Data Pipeline Agenten, die die Webservice-API in Ihrem Namen aufrufen, z. B. die Konsole, CLI und Task Runner.

Die folgenden Grenzwerte gelten für ein einzelnes AWS-Konto. Die Gesamtnutzung des Kontos, einschließlich der Nutzung durch -Benutzer, kann diese Grenzwerte also nicht überschreiten.

Mit der Burst-Rate können Sie Webservice-Aufrufe in inaktiven Zeiträumen einsparen und sie alle in einem kurzen Zeitraum aufbrauchen. CreatePipeline Hat beispielsweise eine reguläre Rate von

einem Anruf alle fünf Sekunden. Wenn Sie den Service 30 Sekunden nicht aufrufen, haben Sie 6 Aufrufe gespart. Sie können dann den Webservice sechsmal in einer Sekunde aufrufen. Da dieser Wert unter dem Burst-Limit liegt und Ihre durchschnittlichen Aufrufe auf dem regulären Ratenlimit belässt, werden die Aufrufe nicht gedrosselt.

Wenn Sie das Raten- und das Burst-Limit überschreiten, schlägt der Webservice-Aufruf fehl und gibt eine Drosselungsausnahme zurück. Die Standardimplementierung eines Workers, Task Runner, wiederholt automatisch API-Aufrufe, die mit einer Drosselungsausnahme fehlschlagen. Task Runner verfügt über einen Backoff, sodass nachfolgende Versuche, die API aufzurufen, in immer längeren Intervallen erfolgen. Wenn Sie einen Worker schreiben, empfehlen wir, dass Sie eine ähnliche Logik für wiederholte Versuche implementieren.

Diese Grenzwerte werden auf ein einzelnes AWS-Konto angewendet.

API	Reguläres Ratenlimit	Burst-Limit
ActivatePipeline	1 Aufruf pro Sekunde	100 Aufrufe
CreatePipeline	1 Aufruf pro Sekunde	100 Aufrufe
DeletePipeline	1 Aufruf pro Sekunde	100 Aufrufe
DescribeObjects	2 Aufrufe pro Sekunde	100 Aufrufe
DescribePipelines	1 Aufruf pro Sekunde	100 Aufrufe
GetPipelineDefinition	1 Aufruf pro Sekunde	100 Aufrufe
PollForTask	2 Aufrufe pro Sekunde	100 Aufrufe
ListPipelines	1 Aufruf pro Sekunde	100 Aufrufe
PutPipelineDefinition	1 Aufruf pro Sekunde	100 Aufrufe
QueryObjects	2 Aufrufe pro Sekunde	100 Aufrufe
ReportTaskProgress	10 Aufrufe pro Sekunde	100 Aufrufe
SetTaskStatus	10 Aufrufe pro Sekunde	100 Aufrufe
SetStatus	1 Aufruf pro Sekunde	100 Aufrufe

API	Reguläres Ratenlimit	Burst-Limit
ReportTaskRunnerHeartbeat	1 Aufruf pro Sekunde	100 Aufrufe
ValidatePipelineDefinition	1 Aufruf pro Sekunde	100 Aufrufe

Überlegungen zur Skalierung

AWS Data Pipeline kann skaliert werden, um eine große Anzahl von gleichzeitigen Aufgaben durchführen zu können. Sie können das System so konfigurieren, dass es automatisch die Ressourcen erstellt, die für die Verarbeitung großer Workloads erforderlich sind. Diese automatisch erstellten Ressourcen sind von Ihnen steuerbar und werden für die Ressourcenlimits für Ihr AWS-Konto berücksichtigt. Wenn Sie beispielsweise die automatische Erstellung eines Amazon EMR-Clusters mit 20 Knoten zur Verarbeitung von Daten konfigurierenAWS Data Pipeline und für IhrAWS Konto ein EC2-Instance-Limit von 20 festgelegt ist, können Sie versehentlich Ihre verfügbaren Backfill-Ressourcen erschöpfen. Daher sollten Sie diese Ressourceneinschränkungen bei Ihrem Design berücksichtigen oder Ihre Kontolimits entsprechend erweitern.

Wenn Sie zusätzliche Kapazität benötigen, können Sie das Antragsformular für das Amazon Web Services Support Center verwenden, um Ihre Kapazität zu erhöhen.

AWS Data Pipeline-Ressourcen

Im Folgenden finden Sie Ressourcen für die Verwendung von AWS Data Pipeline.

- AWS Data PipelineProduktinformationen — Die primäre Website für Informationen zuAWS Data Pipeline.
- AWS Data PipelineTechnische FAQ — Enthält die 20 häufigsten Fragen, die Entwickler zu diesem Produkt stellen.
- Versionshinweise — Geben Sie einen allgemeinen Überblick über die aktuelle Version. Im Einzelnen werden neue Funktionen, Korrekturen und bekannte Probleme vorgestellt.
- AWS-Lösungsforen für Entwickler — Ein Community-basiertes für Entwickler, um technische Fragen zu Amazon Web Services zu erörtern.

- Kurse und Workshops — Links zu rollenbasierten und speziellen Kursen sowie Übungen im Selbststudium zur Verbesserung IhrerAWS -Kompetenzen und Erweiterung Ihrer praktischen Erfahrung.
- AWSDeveloper Center — Entdecken Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen fürAWS -Entwickler.
- AWSDeveloper Tools — Links zu Entwickler-Tools, SDKs, IDE-Toolkits und Befehlszeilen-Tools für die Entwicklung und Verwaltung vonAWS -Anwendungen.
- Ressourcenzentrum für die ersten Schritte — Hier erfahren SieAWS-Konto, wie Sie einrichten, derAWS Community beitreten und Ihre erste Anwendung starten.
- Praktische step-by-step Tutorials zum Starten Ihrer ersten Anwendung aufAWS.
- AWS-Whitepaper — Links zu einer umfangreichen Liste technischerAWS -Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit. Diese Whitepaper wurden vonAWS -Lösungsarchitekten und anderen technischen Experten verfasst.
- AWS Support-Center – Hub für die Erstellung und Verwaltung Ihrer AWS Support-Fälle. Stellt darüber hinaus Links zu weiteren nützlichen Ressourcen bereit, beispielsweise Foren, häufig gestellten technischen Fragen, Status der Service-Integrität und AWS Trusted Advisor.
- AWS Support— Die primäre Website für Informationen zuAWS Support, einem Support-Channel one-on-one, der Sie bei der Erstellung und Ausführung von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS-Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.

- Nutzungsbedingungen für die AWS-Website – Detaillierte Informationen zu unseren Copyright- und Markenbestimmungen, Ihrem Konto, den Lizenzen und anderen Themen.

Dokumentverlauf

Diese Dokumentation ist mit der Version 2012-10-29 von verknüpft. AWS Data Pipeline

Änderung	Beschreibung	Veröffentlichungsdatum
Dokumentation für die Ausführung bestimmter Verfahren mit der AWS CLI hinzugefügt. Die AWS Data Pipeline konsolenbezogenen Verfahren wurden entfernt.	Weitere Informationen finden Sie unter Klonen Ihrer Pipeline, Anzeigen von Pipeline-Protokollen und Erstellen Sie mit der CLI eine Pipeline aus Data Pipeline-Vorlagen.	26. Mai 2023
Weitere Inhalte und Beispiele für die Migration von AWS Data Pipeline zu anderen alternativen Diensten wurden hinzugefügt.	Das Thema für die Migration AWS Data Pipeline zu AWS Step Functions oder Amazon MWAA wurde aktualisiert und enthält weitere Informationen zu den einzelnen Alternativen, Konzeptzuordnungen zwischen den Services und Beispiele. AWS Glue Weitere Informationen finden Sie unter Migrieren von Workloads von AWS Data Pipeline.	31. März 2023
Informationen zur AWS Data Pipeline Unterstützung von IMDSv2 wurden hinzugefügt.	AWS Data Pipelineunterstützt IMDSv2 für Amazon EMR- und Amazon EC2-Ressourcen. Weitere Informationen finden Sie unter Datenschutz in AWS Data Pipeline, EmrCluster und Ec2Resource.	16. Dezember 2022
Es wurde ein Thema für die Migration von AWS Data Pipeline zu anderen alternativen Diensten hinzugefügt.	Es gibt jetzt andere AWS Dienste, die Kunden ein besseres Datenintegrationserlebnis bieten. Sie können typische Anwendungsfälle entweder AWS Data Pipeline AWS Glue zu AWS Step Functions oder Amazon MWAA migrieren. Weitere Informationen	16. Dezember 2022

Änderung	Beschreibung	Veröffentlichungsdatum
	finden Sie unter Migrieren von Workloads von AWS Data Pipeline.	
Die Liste der unterstützten Amazon EC2- und Amazon EMR-Instances wurde aktualisiert. Die Liste der IDs der für die Instances verwendeten HVM(Hardware Virtual Machine)-AMIs wurde aktualisiert.	Die Liste der unterstützten Amazon EC2- und Amazon EMR-Instances wurde aktualisiert. Weitere Informationen finden Sie unter Unterstützte Instance-Typen für Pipeline-Aktivitäten. Die Liste der IDs der für die Instances verwendeten HVM(Hardware Virtual Machine)-AMIs wurde aktualisiert. Weitere Informationen finden Sie unter Syntax und bei der Suche nach `imageId`.	9. November 2018

Änderung	Beschreibung	Veröffentlichungsdatum
Konfiguration für das Anhängen von Amazon EBS-Volumes an Clusterknoten und für den Start eines Amazon EMR-Clusters in einem privaten Subnetz hinzugefügt.	Konfigurationsoptionen wurden zu einem `EMRcluster` -Objekt hinzugefügt. Sie können diese Optionen in Pipelines verwenden, die Amazon EMR-Cluster verwenden. Verwenden Sie die `TaskEbsConfiguration` Felder `coreEbsConfiguration` `masterEbsConfiguration` , und, um das Anhängen von Amazon EBS-Volumes an Core-, Master- und Task-Nodes im Amazon EMR-Cluster zu konfigurieren. Weitere Informationen finden Sie unter EBS-Volumes zu Cluster-Knoten hinzufügen. Verwenden Sie die `ServiceAccessSecurityGroupId` Felder `emrManagedMasterSecurityGroupId` `emrManagedSlaveSecurityGroupId` , und, um einen Amazon EMR-Cluster in einem privaten Subnetz zu konfigurieren. Weitere Informationen finden Sie unter Einen Amazon EMR-Cluster in einem privaten Subnetz konfigurieren. Weitere Informationen zur `EMRcluster` -Syntax finden Sie unter EmrCluster.	19. April 2018
Die Liste der unterstützten Amazon EC2- und Amazon EMR-Instances wurde hinzugefügt.	Es wurde die Liste der Instances hinzugefügt, die von AWS Data Pipeline standardmäßig erstellt werden, wenn Sie keinen Instance-Typ in der Pipeline-Definition angeben. Eine Liste der unterstützten Amazon EC2- und Amazon EMR-Instances wurde hinzugefügt. Weitere Informationen finden Sie unter Unterstützte Instance-Typen für Pipeline-Aktivitäten.	22. März 2018

Änderung	Beschreibung	Veröffentlichungsdatum
Unterstützung für On-Demand-Pipelines wurde hinzugefügt.	• Unterstützung für On-Demand-Pipelines hinzugefügt, die das Wiederholen einer Pipeline durch erneutes Aktivieren ermöglicht.	22. Februar 2016
Zusätzliche Unterstützung für RDS-Datenbanken hinzugefügt	• `rdsInstanceId` , `region` und `jdbcDriverJarUri` wurden RdsDatabase hinzugefügt. • `database` in SqlActivity wurde aktualisiert, um auch `RdsDatabase` zu unterstützen.	17. August 2015
Zusätzliche JDBC-Unterstützung	• `database` in SqlActivity wurde aktualisiert, um auch `JdbcDatabase` zu unterstützen. • `jdbcDriverJarUri` wurde JdbcDatabase hinzugefügt. • `initTimeout` wurde Ec2Resource und EmrCluster hinzugefügt. • `runAsUser` wurde Ec2Resource hinzugefügt.	7. Juli 2015
HadoopActivity, Availability Zone und Spot-Support	• Unterstützung der Übermittlung paralleler Arbeitsaufträge an Hadoop-Cluster wurde hinzugefügt. Weitere Informationen finden Sie unter HadoopActivity. • Möglichkeit zur Anforderung von Spot-Instances mit Ec2Resource und EmrCluster hinzugefügt. • Möglichkeit zum Starten von `EmrCluster` -Ressourcen in einer angegebenen Availability Zone wurden hinzugefügt.	1. Juni 2015
Deaktivieren von Pipelines	Unterstützung zum Deaktivieren aktiver Pipelines hinzugefügt. Weitere Informationen finden Sie unter Deaktivieren Ihrer Pipeline.	7. April 2015

Änderung	Beschreibung	Veröffentlichungsdatum
Vorlagen und Konsole wurden aktualisiert	Neue Vorlagen wurden hinzugefügt. Das Kapitel Erste Schritte wurde aktualisiert und verwendet nun die ShellCommandActivity Vorlage Erste Schritte mit. Weitere Informationen finden Sie unter Erstellen Sie mit der CLI eine Pipeline aus Data Pipeline-Vorlagen.	25. November 2014
VPC-Unterstützung	Unterstützung zum Starten von Ressourcen in einer virtuellen privaten Cloud (VPC) hinzugefügt.	12. März 2014
Regionsunterstützung	Unterstützung für mehrere Serviceregionen hinzugefügt. Außer in `us-east-1` wird AWS Data Pipeline in `eu-west-1` , `ap-northeast-1` , `ap-southeast-2` und `us-west-2` unterstützt.	20. Februar 2014
Amazon Redshift-Unterstützung	Unterstützung für Amazon Redshift wurde hinzugefügtAWS Data Pipeline, einschließlich einer neuen Konsolenvorlage (Copy to Redshift) und eines Tutorials zur Demonstration der Vorlage. Weitere Informationen dazu finden Sie unter Kopieren Sie Daten nach Amazon Redshift mit AWS Data Pipeline, RedshiftDataNode, RedshiftDatabase und RedshiftCopyActivity.	6. November 2013
PigActivity	HinzugefügtPigActivity, was native Unterstützung für Pig bietet. Weitere Informationen finden Sie unter PigActivity.	15. Oktober 2013
Neue Konsolenvorlage, neue Aktivität und neues Datenformat	Die neue CrossRegion DynamoDB Copy-Konsolenvorlage wurde hinzugefügt, einschließlich der neuen Vorlage HiveCopyActivity und DynamoDB. ExportDataFormat	21. August 2013
Cascading-Ausfälle und erneute Ausführungen	Informationen zu Cascading-Ausfällen und erneuten Ausführungen in AWS Data Pipeline hinzugefügt. Weitere Informationen finden Sie unter Cascading-Ausfälle und erneute Ausführungen.	8. August 2013

Änderung	Beschreibung	Veröffentlichungsdatum
Video zur Fehlerbehebung	Video zu grundlegenden Fehlerbehebungsmaßnahmen in AWS Data Pipeline hinzugefügt. Weitere Informationen finden Sie unter Fehlerbehebung.	17. Juli 2013
Bearbeiten von aktiven Pipelines	Zusätzliche Informationen zum Bearbeiten von aktiven Pipelines und erneuten Ausführen von Pipeline-Komponenten hinzugefügt. Weitere Informationen finden Sie unter Bearbeiten Ihrer Pipeline.	17. Juli 2013
Verwenden von Ressourcen in verschiedenen Regionen	Zusätzliche Informationen zum Verwenden von Ressourcen in verschiedenen Regionen hinzugefügt. Weitere Informationen finden Sie unter Verwenden einer Pipeline mit Ressourcen in mehreren Regionen.	17. Juni 2013
Status WAITING_ON_DEPENDENCIES	Status CHECKING_PRECONDITIONS in WAITING_ON_DEPENDENCIES geändert und Laufzeitfeld @waitingOn für Pipeline-Objekte hinzugefügt.	20. Mai 2013
DynamoDB DataFormat	DynamoDB-Vorlage DataFormat hinzugefügt.	23. April 2013
Video zur Verarbeitung von Webprotokollen, Unterstützung von Spot-Instances	Es wurden das Video „Verarbeiten von Webprotokollen mit AWS Data Pipeline, Amazon EMR und Hive“ und die Unterstützung von Amazon EC2 Spot Instances vorgestellt.	21. Februar 2013
	Erstveröffentlichung des AWS Data Pipeline-Entwicklerhandbuchs.	20. Dezember 2012

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.