Guida per gli sviluppatori

AWS Data Pipeline

Versione API 2012-10-29

AWS Data Pipeline: Guida per gli sviluppatori

Table of Contents

AWS Data Pipeline non è più disponibile per i nuovi clienti. I clienti esistenti di AWS Data Pipeline possono continuare a utilizzare il servizio normalmente. Ulteriori informazioni

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.

Che cos'è AWS Data Pipeline?

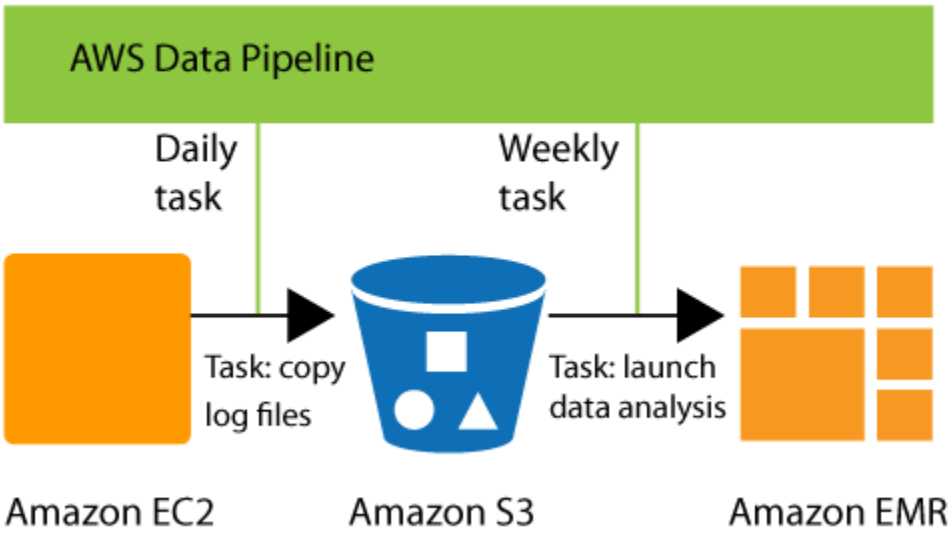
> **Note**
>
> AWS Data Pipeline il servizio è in modalità manutenzione e non sono previste nuove funzionalità o espansioni regionali. Per ulteriori informazioni e per scoprire come migrare i carichi di lavoro esistenti, consulta. Migrazione dei carichi di lavoro da AWS Data Pipeline

AWS Data Pipeline è un servizio web che puoi utilizzare per automatizzare lo spostamento e la trasformazione dei dati. Con AWS Data Pipeline, puoi definire flussi di lavoro basati sui dati, in modo che le attività possano dipendere dal completamento con successo delle attività precedenti. Definisci i parametri delle trasformazioni dei dati e AWS Data Pipeline applichi la logica che hai impostato.

I seguenti componenti AWS Data Pipeline collaborano per gestire i dati:

- Una pipeline definition (definizione di pipeline) specifica la logica di business della gestione dei dati. Per ulteriori informazioni, consulta Sintassi del file di definizione della pipeline.
- Una pipeline pianifica ed esegue le attività creando istanze Amazon EC2 per eseguire le attività lavorative definite. È possibile caricare la definizione di pipeline nella e quindi attivarla. È possibile modificare la definizione di pipeline e attivare di nuovo la pipeline affinché abbia effetto. Puoi disattivare la pipeline, modificare un'origine dati e quindi attivare la pipeline di nuovo. Quando la pipeline non è più necessaria, è possibile eliminarla.
- Task Runner analizza le attività e quindi le esegue. Ad esempio, Task Runner potrebbe copiare i file di registro su Amazon S3 e avviare cluster Amazon EMR. Task Runner viene installato e viene eseguito automaticamente sulle risorse create dalle definizioni della pipeline. È possibile scrivere un'applicazione task runner personalizzata oppure utilizzare l'applicazione Task Runner fornita da. AWS Data Pipeline Per ulteriori informazioni, consulta Runner delle attività.

Ad esempio, puoi AWS Data Pipeline archiviare i log del tuo server Web su Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) ogni giorno e poi eseguire un cluster Amazon EMR (Amazon EMR) settimanale su quei log per generare report sul traffico. AWS Data Pipeline pianifica le attività giornaliere per copiare i dati e l'attività settimanale per avviare il cluster Amazon EMR. AWS Data Pipeline assicura inoltre che Amazon EMR attenda il caricamento dei dati dell'ultimo giorno su Amazon S3 prima di iniziare l'analisi, anche in caso di ritardo imprevisto nel caricamento dei log.



Indice

Migrazione dei carichi di lavoro da AWS Data Pipeline

AWS ha lanciato il AWS Data Pipeline servizio nel 2012. A quel tempo, i clienti cercavano un servizio che li aiutasse a spostare in modo affidabile i dati tra diverse fonti di dati utilizzando una varietà di opzioni di elaborazione. Ora esistono altri servizi che offrono ai clienti un'esperienza migliore. Ad esempio, puoi utilizzarlo per eseguire e AWS Glue orchestrare le applicazioni Apache Spark, Step Functions AWS per aiutare a orchestrare i AWS componenti del servizio o Amazon Managed Workflows for Apache Airflow (Amazon MWAA) per aiutare a gestire l'orchestrazione del flusso di lavoro per Apache Airflow.

Questo argomento spiega come migrare da opzioni alternative. AWS Data Pipeline L'opzione scelta dipende dal carico di lavoro corrente su. AWS Data Pipeline Puoi migrare i casi d'uso tipici AWS Data Pipeline verso AWS Step Functions o Amazon MWAA. AWS Glue

Migrazione dei carichi di lavoro in AWS Glue

AWS Glue è un servizio di integrazione dati serverless che semplifica agli utenti analitici il rilevamento, la preparazione, lo spostamento e l'integrazione di dati da più origini. Include strumenti per la creazione, l'esecuzione di lavori e l'orchestrazione dei flussi di lavoro. Con AWS Glue, puoi scoprire e connetterti a più di 70 diverse fonti di dati e gestire i tuoi dati in un catalogo di dati centralizzato. Puoi creare, eseguire e monitorare visivamente pipeline di estrazione, trasformazione e caricamento (ETL) per caricare dati nei data lake. Inoltre, puoi eseguire ricerche e query immediatamente nei dati catalogati utilizzando Amazon Athena, Amazon EMR e Amazon Redshift Spectrum.

Ti consigliamo di migrare il AWS Data Pipeline carico di lavoro a quando: AWS Glue

- Stai cercando un servizio di integrazione dei dati senza server che supporti varie fonti di dati, interfacce di creazione tra cui editor visivi e notebook e funzionalità avanzate di gestione dei dati come la qualità dei dati e il rilevamento dei dati sensibili.
- Il carico di lavoro può essere migrato verso AWS Glue flussi di lavoro, job (in Python o Apache Spark) e crawler (ad esempio, la pipeline esistente è costruita su Apache Spark).
- È necessaria un'unica piattaforma in grado di gestire tutti gli aspetti della pipeline di dati, tra cui l'acquisizione, l'elaborazione, il trasferimento, i test di integrità e i controlli di qualità.
- La tua pipeline esistente è stata creata da un modello predefinito sulla AWS Data Pipeline console, ad esempio l'esportazione di una tabella DynamoDB in Amazon S3, e stai cercando un modello con lo stesso scopo.
- Il tuo carico di lavoro non dipende da una specifica applicazione dell'ecosistema Hadoop come Apache Hive.
- Il tuo carico di lavoro non richiede l'orchestrazione di server locali.

AWS addebita una tariffa oraria, fatturata al secondo, per i crawler (rilevamento dei dati) e i job ETL (elaborazione e caricamento dei dati). AWS Glue Studio è un motore di orchestrazione integrato per AWS Glue le risorse e viene offerto senza costi aggiuntivi. Scopri di più sui prezzi nella AWS Glue sezione Prezzi.

Migrazione dei carichi di lavoro a Step Functions AWS

AWS Step Functions è un servizio di orchestrazione serverless che consente di creare flussi di lavoro per le applicazioni aziendali critiche. Con Step Functions utilizzi un editor visivo per creare flussi di lavoro e integrarli direttamente con oltre 11.000 azioni per oltre 250 AWS servizi, come AWS

Lambda, Amazon EMR, DynamoDB e altri. Puoi usare Step Functions per orchestrare le pipeline di elaborazione dei dati, gestire gli errori e lavorare con i limiti di throttling sui servizi sottostanti. AWS È possibile creare flussi di lavoro che elaborano e pubblicano modelli di machine learning, orchestrano microservizi e AWS controllano servizi, ad esempio per creare flussi di lavoro di estrazione, trasformazione e AWS Glue caricamento (ETL). È possibile anche creare flussi di lavoro automatizzati e di lunga durata per applicazioni che richiedono l'interazione umana.

Analogamente AWS Data Pipeline, AWS Step Functions è un servizio completamente gestito fornito da AWS. Non ti verrà richiesto di gestire l'infrastruttura, applicare patch worker, gestire gli aggiornamenti delle versioni del sistema operativo o simili.

Ti consigliamo di migrare il AWS Data Pipeline carico di lavoro a AWS Step Functions quando:

- Stai cercando un servizio di orchestrazione del flusso di lavoro senza server e ad alta disponibilità.
- Stai cercando una soluzione conveniente che addebiti una granularità dell'esecuzione di una singola attività.
- I tuoi carichi di lavoro orchestrano attività per molti altri AWS servizi, come Amazon EMR, Lambda o DynamoDB. AWS Glue
- Stai cercando una soluzione low-code dotata di un drag-and-drop visual designer per la creazione di flussi di lavoro e che non richieda l'apprendimento di nuovi concetti di programmazione.
- Stai cercando un servizio che fornisca integrazioni con oltre 250 altri AWS servizi che coprano oltre 11.000 azioni out-of-the-box, oltre a consentire integrazioni con attività e servizi non personalizzati.AWS

AWS Data Pipeline Sia Step Functions che Step Functions utilizzano il formato JSON per definire i flussi di lavoro. Ciò consente di archiviare i flussi di lavoro nel controllo del codice sorgente, gestire le versioni, controllare l'accesso e automatizzare con CI/CD. Step Functions utilizza una sintassi chiamata Amazon State Language che è completamente basata su JSON e consente una transizione senza interruzioni tra le rappresentazioni testuali e visive del flusso di lavoro.

Con Step Functions, puoi scegliere la stessa versione di Amazon EMR in cui utilizzi attualmente. AWS Data Pipeline

Per la migrazione delle attività sulle risorse AWS Data Pipeline gestite, puoi utilizzare l'integrazione dei servizi AWS SDK su Step Functions per automatizzare il provisioning e la pulizia delle risorse.

Per la migrazione delle attività su server locali, istanze EC2 gestite dall'utente o un cluster EMR gestito dall'utente, puoi installare un agente SSM sull'istanza. È possibile avviare il comando tramite

AWS Systems Manager Run Command di Step Functions. Puoi anche avviare la macchina a stati dalla pianificazione definita in Amazon EventBridge.

AWS Step Functions ha due tipi di flussi di lavoro: flussi di lavoro standard e flussi di lavoro rapidi. Per i flussi di lavoro standard, l'addebito viene calcolato in base al numero di transizioni di stato necessarie per eseguire l'applicazione. Per Express Workflows, i costi vengono addebitati in base al numero di richieste per il flusso di lavoro e alla sua durata. Scopri di più sui prezzi in AWS Step Functions Pricing.

Migrazione dei carichi di lavoro su Amazon MWAA

Amazon MWAA (Managed Workflows for Apache Airflow) è un servizio di orchestrazione gestito per Apache Airflow che semplifica la configurazione e la gestione di pipeline di dati nel cloud su larga scala. end-to-end Apache Airflow è uno strumento open source utilizzato per creare, pianificare e monitorare in modo programmatico sequenze di processi e attività denominate «flussi di lavoro». Con Amazon MWAA, puoi usare i linguaggi di programmazione Airflow e Python per creare flussi di lavoro senza dover gestire l'infrastruttura sottostante per scalabilità, disponibilità e sicurezza. Amazon MWAA ridimensiona automaticamente la capacità di esecuzione del flusso di lavoro per soddisfare le tue esigenze ed è integrato con i servizi AWS di sicurezza per aiutarti a fornire un accesso rapido e sicuro ai tuoi dati.

Analogamente AWS Data Pipeline, Amazon MWAA è un servizio completamente gestito fornito da. AWS Sebbene sia necessario apprendere diversi nuovi concetti specifici relativi a questi servizi, non è necessario gestire l'infrastruttura, applicare patch worker, gestire gli aggiornamenti delle versioni del sistema operativo o simili.

Ti consigliamo di migrare i AWS Data Pipeline carichi di lavoro su Amazon MWAA quando:

- Stai cercando un servizio gestito e ad alta disponibilità per orchestrare i flussi di lavoro scritti in Python.
- Desideri passare a una tecnologia open source completamente gestita e ampiamente adottata, Apache Airflow, per la massima portabilità.
- È necessaria un'unica piattaforma in grado di gestire tutti gli aspetti della pipeline di dati, tra cui l'acquisizione, l'elaborazione, il trasferimento, i test di integrità e i controlli di qualità.
- Stai cercando un servizio progettato per l'orchestrazione della pipeline di dati con funzionalità come un'interfaccia utente avanzata per l'osservabilità, i riavvii per i flussi di lavoro non riusciti, i backfill e i nuovi tentativi di esecuzione delle attività.

- Stai cercando un servizio che includa più di 800 operatori e sensori predefiniti, che coprano e non coprano servizi. AWS AWS

I flussi di lavoro Amazon MWAA sono definiti come Directed Acyclic Graphs (DAGs) utilizzando Python, quindi puoi trattarli anche come codice sorgente. Il framework Python estensibile di Airflow ti consente di creare flussi di lavoro che si connettono praticamente con qualsiasi tecnologia. È dotato di una ricca interfaccia utente per la visualizzazione e il monitoraggio dei flussi di lavoro e può essere facilmente integrato con i sistemi di controllo delle versioni per automatizzare il processo. CI/CD

Con Amazon MWAA, puoi scegliere la stessa versione di Amazon EMR in cui utilizzi attualmente. AWS Data Pipeline

AWS addebita in base al tempo di funzionamento dell'ambiente Airflow e qualsiasi ulteriore scalabilità automatica per fornire maggiore capacità ai dipendenti o ai server Web. Scopri di più sui prezzi in Amazon Managed Workflows for Apache Airflow Pricing.

Mappatura dei concetti

La tabella seguente contiene la mappatura dei concetti principali utilizzati dai servizi. Aiuterà le persone che hanno familiarità con Data Pipeline a comprendere la terminologia Step Functions e MWAA.

Data Pipeline	Aderenza	Step Functions	Amazon MWAA
Pipelines	Flussi di lavoro	Flussi di lavoro	Grafi acrilici diretti
Definizione della pipeline JSON	Definizione del flusso di lavoro o progetti basati su Python	Amazon State Language JSON	Basato su Python
Attività	Jobs	Stati e attività	Attività (operatori e sensori)
Istanze	Job viene eseguito	Esecuzioni	DAG funziona
Tentativi	Tentativi di nuovo tentativo	Catcher e retrier	Tentativi

Data Pipeline	Aderenza	Step Functions	Amazon MWAA
Pianificazione della pipeline	Pianifica i trigger	EventBridge Attività dello scheduler	Cron, orari, sensibili ai dati
Espressioni e funzioni della pipeline	Libreria Blueprint	Step Functions, funzioni intrinseche e Lambda AWS	Framework Python estensibile

Esempi

Nelle sezioni seguenti sono elencati esempi pubblici a cui è possibile fare riferimento per migrare da un servizio AWS Data Pipeline all'altro. È possibile utilizzarli come esempi e creare la propria pipeline sui singoli servizi aggiornandola e testandola in base al proprio caso d'uso.

AWS Glue esempi

L'elenco seguente contiene implementazioni di esempio per i casi AWS Data Pipeline d'uso più comuni con. AWS Glue

- Esecuzione di job Spark
- Copia di dati da JDBC ad Amazon S3 (incluso Amazon Redshift)
- Copia di dati da Amazon S3 a JDBC (incluso Amazon Redshift)
- Copia di dati da Amazon S3 a DynamoDB
- Spostamento di dati da e verso Amazon Redshift
- Accesso interregionale tra più account alle tabelle Dynamodb

AWS Esempi di Step Functions

L'elenco seguente contiene implementazioni di esempio per i AWS Data Pipeline casi d'uso più comuni con Step Functions AWS .

- Gestione di un job in Amazon EMR
- Esecuzione di un processo di elaborazione dati su Amazon EMR Serverless
- Lavori in esecuzione Hive/Pig/Hadoop

- Interrogazione di set di dati di grandi dimensioni (Amazon Athena, Amazon S3,) AWS Glue
- Esecuzione di flussi di lavoro ETL con Amazon Redshift
- AWS Glue Orchestrazione dei crawler

Guarda tutorial aggiuntivi ed esempi di progetti per l'utilizzo di AWS Step Functions.

Esempi di Amazon MWAA

L'elenco seguente contiene implementazioni di esempio per i casi AWS Data Pipeline d'uso più comuni con Amazon MWAA.

- Esecuzione di un job Amazon EMR
- Creazione di un plug-in personalizzato per Apache Hive e Hadoop
- Copia dei dati da Amazon S3 a Redshift
- Esecuzione di uno script Shell su un'istanza EC2 remota
- Orchestrazione di flussi di lavoro ibridi (on-premise)

Consulta tutorial ed esempi di progetti aggiuntivi per l'uso di Amazon MWAA.

Servizi correlati

AWS Data Pipeline funziona con i seguenti servizi per archiviare dati.

- Amazon DynamoDB: fornisce un database NoSQL completamente gestito con prestazioni veloci a basso costo. Per ulteriori informazioni, consulta Amazon DynamoDB Developer Guide.
- Amazon RDS: fornisce un database relazionale completamente gestito che si adatta a set di dati di grandi dimensioni. Per ulteriori informazioni, consulta la Amazon Relational Database Service Developer Guide.
- Amazon Redshift: fornisce un data warehouse veloce, completamente gestito e su scala petabyte che semplifica ed economica l'analisi di grandi quantità di dati. Per ulteriori informazioni, consulta la Amazon Redshift Database Developer Guide.
- Amazon S3: fornisce uno storage di oggetti sicuro, durevole e altamente scalabile. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon Simple Storage Service.

AWS Data Pipeline funziona con i seguenti servizi di elaborazione per trasformare i dati.

- Amazon EC2: fornisce una capacità di elaborazione ridimensionabile, letteralmente server nei data center di Amazon, che puoi utilizzare per creare e ospitare i tuoi sistemi software. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon EC2.
- Amazon EMR: semplifica, velocizza ed economicamente vantaggiosa la distribuzione e l'elaborazione di grandi quantità di dati su server Amazon EC2, utilizzando un framework come Apache Hadoop o Apache Spark. Per ulteriori informazioni, consulta la Amazon EMR Developer Guide.

Accedere AWS Data Pipeline

È possibile creare, accedere e gestire le pipeline utilizzando una qualsiasi delle seguenti interfacce:

- Console di gestione AWS— Fornisce un'interfaccia web che è possibile utilizzare per accedere AWS Data Pipeline.
- AWS Command Line Interface (AWS CLI) — Fornisce comandi per un'ampia gamma di servizi AWS AWS Data Pipeline, inclusi ed è supportato su Windows, macOS e Linux. Per ulteriori informazioni sull'installazione di AWS CLI, consulta AWS Command Line Interface. Per un elenco di comandi per AWS Data Pipeline, consulta datapipeline.
- AWS SDKs: fornisce informazioni specifiche per la lingua APIs e si occupa di molti dettagli di connessione, come il calcolo delle firme, la gestione dei nuovi tentativi di richiesta e la gestione degli errori. Per ulteriori informazioni, consulta AWS SDKs.
- API di interrogazione: fornisce chiamate APIs di basso livello utilizzando richieste HTTPS. L'API di interrogazione è il modo più diretto per accedere al AWS Data Pipeline, ma richiede che la propria applicazione gestisca dettagli di basso livello, come la generazione di un hash per la firma della richiesta e la gestione degli errori. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di AWS Data Pipeline.

Prezzi

I prezzi di Amazon Web Services sono calcolati in base all'uso effettivo. In effetti AWS Data Pipeline, paghi per la tua pipeline in base alla frequenza con cui è programmata l'esecuzione delle tue attività e dei prerequisiti e al luogo in cui vengono eseguite. Per ulteriori informazioni, consultare AWS Data Pipeline Prezzi.

Se l'account AWS è inferiore a 12 mesi, hai diritto a utilizzare il piano gratuito. Il piano gratuito include tre precondizioni a bassa frequenza e cinque attività a bassa frequenza al mese senza alcun costo aggiuntivo. Per ulteriori informazioni, consulta Piano gratuito di AWS.

Tipi di istanza supportati per attività di lavoro delle pipeline

Quando AWS Data Pipeline esegue una pipeline, compila i componenti della pipeline per creare un set di istanze Amazon EC2 utilizzabili. Ogni istanza contiene tutte le informazioni necessarie per l'esecuzione di un'attività specifica. Il set completo di istanze costituisce l'elenco delle attività della pipeline. AWS Data Pipeline passa le istanze ai runner delle attività per essere processate.

Le istanze EC2 sono disponibili in diverse configurazioni, note come tipi di istanze. Ogni tipo di istanza dispone di diverse capacità di CPU, input/output e storage. Oltre a specificare il tipo di istanza per un'attività, puoi scegliere diverse opzioni di acquisto. Non tutti i tipi di istanze sono disponibili in tutte le regioni AWS. Se un tipo di istanza non è disponibile, è possibile che la pipeline non riesca a effettuare il provisioning o che il provisioning si blocchi. Per informazioni sulla disponibilità delle istanze, consulta la pagina dei prezzi di Amazon EC2. Aprire il link per l'opzione di acquisto di istanze e filtrare in base alla Region (Regione) per vedere se un tipo di istanza è disponibile nella regione. Per ulteriori informazioni su questi tipi di istanze, famiglie e tipi di virtualizzazione, consulta Amazon EC2 Instances e Amazon Linux AMI Instance Type Matrix.

Le tabelle seguenti descrivono i tipi di istanza supportati. AWS Data Pipeline Puoi utilizzarle AWS Data Pipeline per avviare istanze Amazon EC2 in qualsiasi regione, comprese le regioni in cui non AWS Data Pipeline è supportata. Per informazioni sulle regioni in cui AWS Data Pipeline è supportato, consulta AWS Regions and Endpoints.

Indice

Istanze Amazon EC2 predefinite per regione AWS

Se non si specifica un tipo di istanza nella definizione della pipeline, AWS Data Pipeline avvia un'istanza per impostazione predefinita.

La tabella seguente elenca le istanze Amazon EC2 AWS Data Pipeline utilizzate di default nelle regioni in cui AWS Data Pipeline è supportata.

Nome della regione	Regione	Tipo di istanza
Stati Uniti orientali (Virginia settentrionale)	us-east-1	m1.small
Stati Uniti occidentali (Oregon)	us-west-2	m1.small
Asia Pacifico (Sydney)	ap-southeast-2	m1.small
Asia Pacifico (Tokyo)	ap-northeast-1	m1.small
UE (Irlanda)	eu-west-1	m1.small

La tabella seguente elenca le istanze Amazon EC2 che vengono AWS Data Pipeline avviate per impostazione predefinita nelle regioni in cui AWS Data Pipeline non è supportata.

Nome della regione	Regione	Tipo di istanza
Stati Uniti orientali (Ohio)	us-east-2	t2.small
Stati Uniti occidentali (California settentrionale)	us-west-1	m1.small
Asia Pacifico (Mumbai)	ap-south-1	t2.small
Asia Pacifico (Singapore)	ap-southeast-1	m1.small
Asia Pacifico (Seoul)	ap-northeast-2	t2.small
Canada (Centrale)	ca-central-1	t2.small
UE (Francoforte)	eu-central-1	t2.small
UE (Londra)	eu-west-2	t2.small
UE (Parigi)	eu-west-3	t2.small
Sud America (San Paolo)	sa-east-1	m1.small

Istanze Amazon EC2 aggiuntive supportate

Oltre alle istanze predefinite che vengono create se non si specifica un tipo di istanza nella definizione della pipeline, vengono supportate le seguenti istanze.

La tabella seguente elenca le istanze Amazon EC2 che AWS Data Pipeline supportano e possono creare, se specificato.

Classe istanza	Tipi di istanza
Uso generale	t2.nano | t2.micro | t2.small | t2.medium | t2.large
Calcolo ottimizzato	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
Memoria ottimizzata	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
Storage ottimizzato	i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge | hs1.8xlarge | g2.2xlarge | g2.8xlarge | d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge

Istanze Amazon EC2 supportate per cluster Amazon EMR

Questa tabella elenca le istanze Amazon EC2 che AWS Data Pipeline supportano e possono creare per i cluster Amazon EMR, se specificato. Per ulteriori informazioni, consulta Tipi di istanza supportati nella Guida alla gestione di Amazon EMR.

Classe istanza	Tipi di istanza
Uso generale	m1.small | m1.medium | m1.large | m1.xlarge | m3.xlarge | m3.2xlarge
Calcolo ottimizzato	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
Memoria ottimizzata	m2.xlarge | m2.2xlarge | m2.4xlarge | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | cr1.8xlarge | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16large | m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge | r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge
Storage ottimizzato	h1.4xlarge | hs1.2xlarge | hs1.4xlarge| hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4large | i2.8xlarge | d2.xlarge | d2.2xlarge| d2.4xlarge | d2.8xlarge
Accelerazione informatica	g2.2xlarge | cg1.4xlarge

AWS Data Pipeline Concetti

Prima di iniziare, leggi i concetti e i componenti chiave di AWS Data Pipeline.

Indice

Definizione della pipeline

Una definizione di pipeline è il modo in cui comunichi la tua logica aziendale. AWS Data Pipeline Contiene le seguenti informazioni:

- Nomi, percorsi e formati delle origini dati
- Attività per la trasformazione dei dati
- La pianificazione per tali attività
- Risorse che eseguono attività e precondizioni
- Le precondizioni devono essere soddisfatte prima che le attività possano essere programmate
- Metodi per avvisarti con aggiornamenti di stato durante l'esecuzione della pipeline

Dalla definizione della pipeline, AWS Data Pipeline determina le attività, le pianifica e le assegna ai task runner. Se un'attività non viene completata correttamente, AWS Data Pipeline riprova l'attività in base alle istruzioni fornite e, se necessario, la riassegna a un altro task runner. Se l'operazione ha esito negativo ripetutamente, è possibile configurare la pipeline per la notifica.

Ad esempio, nella definizione della pipeline, puoi specificare che i file di log generati dalla tua applicazione vengano archiviati ogni mese nel 2013 in un bucket Amazon S3. AWS Data Pipeline creerebbe quindi 12 attività, ciascuna delle quali copia più di un mese di dati, indipendentemente dal fatto che il mese contenga 30, 31, 28 o 29 giorni.

Puoi creare una definizione di pipeline nei seguenti modi:

- Graficamente, utilizzando la console AWS Data Pipeline
- Testualmente, scrivendo un file in formato JSON utilizzato dall'interfaccia a riga di comando
- A livello di codice, chiamando il servizio Web con uno degli AWS SDKs o l'API AWS Data Pipeline

Una definizione di pipeline può contenere i seguenti tipi di componenti.

Componenti della pipeline

Nodi di dati

La posizione dei dati di input per un'attività o il percorso in cui i dati di output vengono archiviati.

Attività

Una definizione di lavoro da eseguire in base a una pianificazione utilizzando una risorsa di calcolo e, in genere, i nodi di dati di input e di output.

Precondizioni

Un'istruzione condizionale che deve essere true prima di eseguire un'operazione.

Resources

La risorsa di calcolo che esegue il lavoro definito da una pipeline.

Azioni

Operazione che viene attivata quando vengono soddisfatte determinate condizioni, per esempio la non riuscita di un'attività.

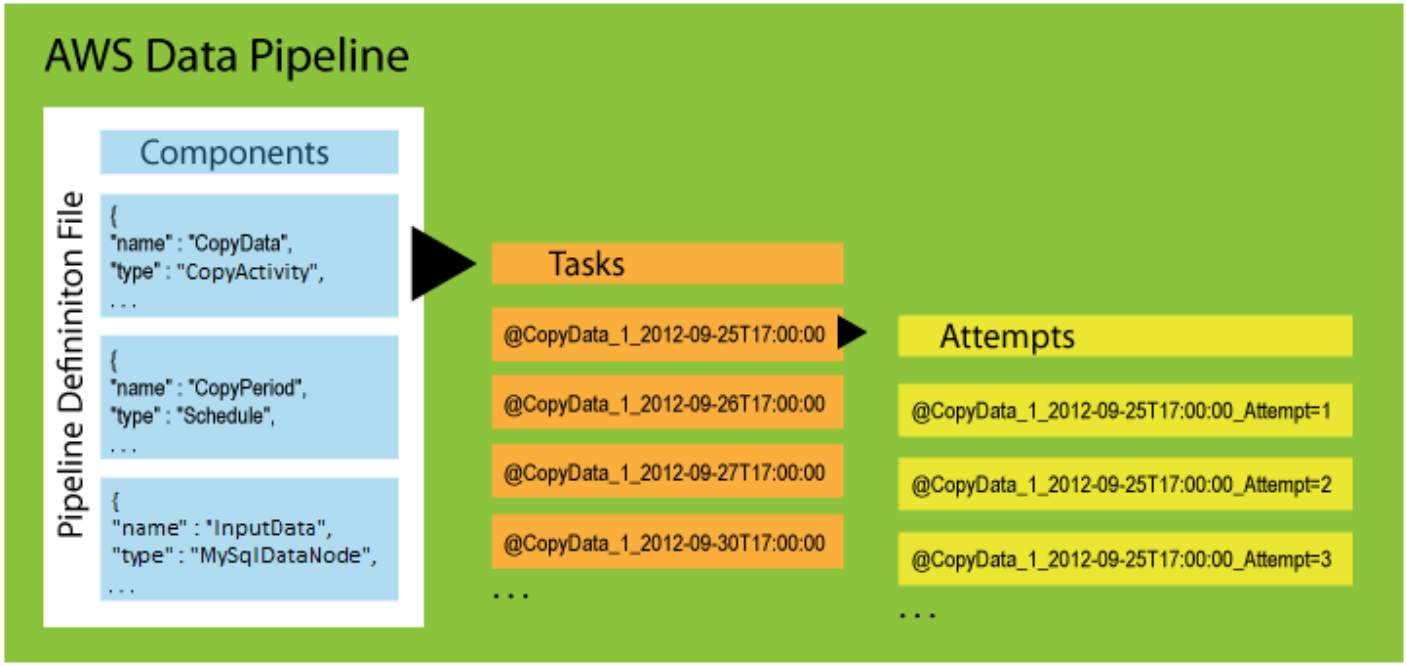
Per ulteriori informazioni, consulta Sintassi del file di definizione della pipeline.

Componenti, istanze e tentativi della pipeline

Esistono tre tipi di elementi associati a una pipeline pianificata:

- Componenti della pipeline: i componenti della pipeline rappresentano la logica di business della pipeline e sono rappresentati dalle diverse sezioni di una definizione di pipeline. I componenti della pipeline specificano le origini dati, le attività, la pianificazione e le precondizioni del flusso di lavoro. È possibile ereditare le proprietà da componenti padre. Relazioni tra i componenti vengono definite per riferimento. I componenti della pipeline definiscono le regole di gestione dei dati.
- Istanze: quando AWS Data Pipeline esegue una pipeline, compila i componenti della pipeline per creare un set di istanze utilizzabili. Ogni istanza contiene tutte le informazioni necessarie per l'esecuzione di un'attività specifica. Il set completo di istanze è l'elenco delle cose da fare della pipeline. AWS Data Pipeline consegna le istanze ai task runner per l'elaborazione.
- Tentativi: per fornire una solida gestione dei dati, AWS Data Pipeline riprova un'operazione fallita. Continua a farlo finché l'attività non raggiunge il numero massimo di tentativi consentiti. Gli Attempt Objects monitorano i diversi tentativi, i risultati e i motivi di errore, ove applicabile. Essenzialmente, si tratta dell'istanza con un contatore. AWS Data Pipeline esegue nuovi tentativi utilizzando le stesse risorse dei tentativi precedenti, come i cluster Amazon EMR e le istanze EC2.

> **Note**
>
> Rieseguire le attività non riuscite è un aspetto importante di una strategia di tolleranza ai guasti e le definizioni di AWS Data Pipeline forniscono condizioni e soglie per controllare i nuovi tentativi. Tuttavia, troppi tentativi possono ritardare il rilevamento di un errore irreversibile perché AWS Data Pipeline non segnala il guasto finché non ha esaurito tutti i tentativi specificati dall'utente. I nuovi tentativi aggiuntivi possono comportare costi aggiuntivi se sono in esecuzione su risorse AWS. Di conseguenza, valuta attentamente quando è opportuno superare le impostazioni AWS Data Pipeline predefinite utilizzate per controllare i nuovi tentativi e le impostazioni correlate.

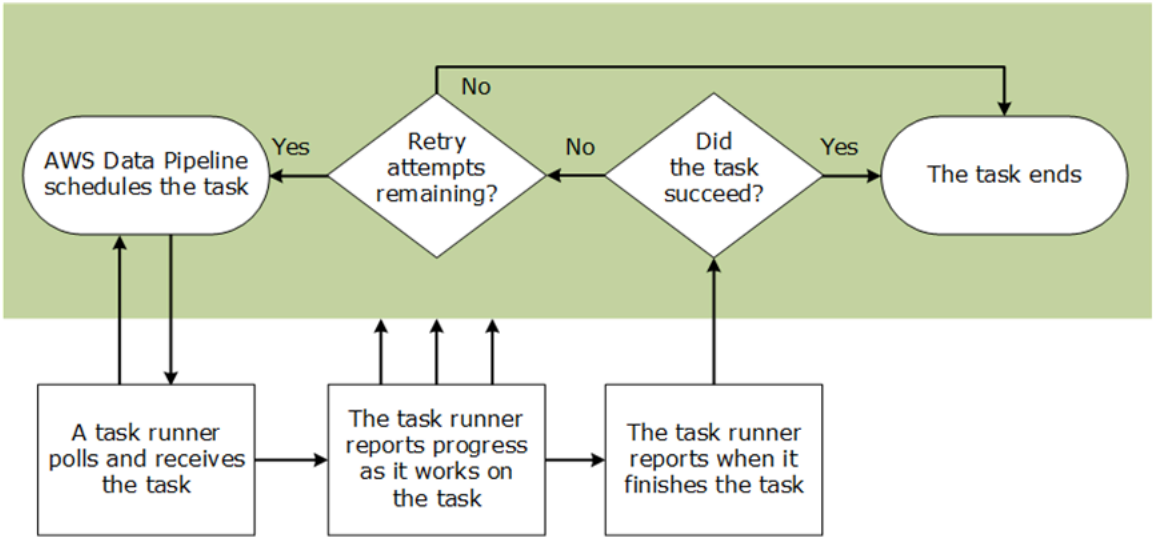


Runner delle attività

Un task runner è un'applicazione che analizza le attività e quindi AWS Data Pipeline le esegue.

Task Runner è un'implementazione predefinita di un task runner fornita da. AWS Data Pipeline Quando Task Runner è installato e configurato, esegue il polling AWS Data Pipeline delle attività associate alle pipeline attivate. Quando un'attività viene assegnata a Task Runner, la esegue e ne riporta lo stato a. AWS Data Pipeline

Il diagramma seguente illustra come AWS Data Pipeline e un task runner interagiscono per elaborare un'operazione pianificata. Un'attività è un'unità di lavoro discreta che il AWS Data Pipeline servizio condivide con un task runner. È diversa da una pipeline, che è una definizione generale di attività e risorse che normalmente produce diverse attività.



Esistono due modi per utilizzare Task Runner per elaborare la pipeline:

- AWS Data Pipeline installa Task Runner per voi su risorse avviate e gestite dal servizio web. AWS Data Pipeline
- Installa Task Runner su una risorsa di calcolo che gestisci, come un'istanza EC2 a esecuzione prolungata o un server locale.

Per ulteriori informazioni sull'utilizzo di Task Runner, consulta Lavorare con Task Runner.

Nodi di dati

In AWS Data Pipeline, un nodo di dati definisce la posizione e il tipo di dati che un'attività di pipeline utilizza come input o output. AWS Data Pipeline supporta i seguenti tipi di nodi di dati:

DBDataNodo Dynamo

Una tabella DynamoDB che contiene dati HiveActivity da utilizzare o da utilizzare. EmrActivity

SqlDataNode

Una tabella SQL e una query di database che rappresentano i dati per un'attività di pipeline da utilizzare.

> **Note**
>
> In precedenza, MySqlDataNode veniva utilizzato. Usa SqlDataNode invece.

RedshiftDataNode

Una tabella Amazon Redshift che contiene dati RedshiftCopyActivity da utilizzare.

S3 DataNode

Una posizione Amazon S3 che contiene uno o più file da utilizzare per un'attività di pipeline.

Database

AWS Data Pipeline supporta i seguenti tipi di database:

JdbcDatabase

Un database JDBC.

RdsDatabase

Un database Amazon RDS.

RedshiftDatabase

Un database Amazon Redshift.

Attività

In AWS Data Pipeline, un'attività è un componente della pipeline che definisce il lavoro da eseguire. AWS Data Pipeline fornisce diverse attività preconfezionate che si adattano a scenari comuni, come lo spostamento dei dati da una posizione all'altra, l'esecuzione di query Hive e così via. Le attività sono estendibili, perciò è possibile eseguire script personalizzati per supportare infinite combinazioni.

AWS Data Pipeline supporta i seguenti tipi di attività:

CopyActivity

Copia i dati da una posizione a un'altra.

EmrActivity

Esegue un cluster Amazon EMR.

HiveActivity

Esegue una query Hive su un cluster Amazon EMR.

HiveCopyActivity

Esegue una query Hive su un cluster Amazon EMR con supporto per il filtraggio avanzato dei dati e supporto per e. S3 DataNode DBDataNodo Dynamo

PigActivity

Esegue uno script Pig su un cluster Amazon EMR.

RedshiftCopyActivity

Copia i dati da e verso le tabelle Amazon Redshift.

ShellCommandActivity

Esegue un comando UNIX/Linux shell personalizzato come attività.

SqlActivity

Esegue una query SQL su un database.

Alcune attività hanno un supporto speciale per la gestione temporanea dei dati e delle tabelle di database. Per ulteriori informazioni, consulta Dati e tabelle in gestione temporanea con attività della pipeline.

Precondizioni

In AWS Data Pipeline, una precondizione è un componente della pipeline contenente istruzioni condizionali che devono essere vere prima che un'attività possa essere eseguita. Ad esempio, una condizione preliminare può verificare se i dati di origine sono presenti prima che un'attività di pipeline tenti di copiarli. AWS Data Pipeline fornisce diverse precondizioni preconfezionate che soddisfano scenari comuni, ad esempio l'esistenza di una tabella di database, la presenza di una chiave Amazon S3 e così via. Tuttavia, le precondizioni sono estendibili e consentono di eseguire script personalizzati a supporto di infinite combinazioni.

Sono disponibili due tipi di precondizioni: le precondizioni gestite dal sistema e le precondizioni gestite dall'utente. Le precondizioni gestite dal sistema vengono gestite dal servizio AWS Data Pipeline Web per conto dell'utente e non richiedono risorse di calcolo. Le precondizioni gestite dall'utente vengono eseguite solo sulla risorsa di calcolo che specifichi utilizzando i campi `runsOn` o `workerGroup`. La risorsa `workerGroup` deriva dall'attività che utilizza la precondizione.

Precondizioni gestite dal sistema

La dinamo esiste DBData

Verifica se i dati esistono in una tabella DynamoDB specifica.

La dinamo DBTable esiste

Verifica se esiste una tabella DynamoDB.

S3 KeyExists

Verifica se esiste una chiave Amazon S3.

S3 PrefixNotEmpty

Verifica se un prefisso Amazon S3 è vuoto.

Precondizioni gestite dall'utente

Exists

Verifica se esiste un nodo di dati.

ShellCommandPrecondition

Esegue un comando Unix/Linux shell personalizzato come condizione preliminare.

Resources

In AWS Data Pipeline, una risorsa è la risorsa computazionale che esegue il lavoro specificato da un'attività di pipeline. AWS Data Pipeline supporta i seguenti tipi di risorse:

Ec2Resource

Un'istanza EC2 che esegue il lavoro definito da un'attività di pipeline.

EmrCluster

Un cluster Amazon EMR che esegue il lavoro definito da un'attività di pipeline, ad esempio. EmrActivity

Le risorse possono essere eseguite nella stessa regione con i set di dati attivi e anche una regione diversa da AWS Data Pipeline. Per ulteriori informazioni, consulta Utilizzo di una pipeline con risorse in più regioni.

Limiti delle risorse

AWS Data Pipeline si adatta a un numero enorme di attività simultanee ed è possibile configurarlo per creare automaticamente le risorse necessarie per gestire carichi di lavoro di grandi dimensioni. Queste risorse create automaticamente sono sotto il controllo dell'utente e vengono conteggiate ai fini dei limiti delle risorse dell'account AWS. Ad esempio, se configuri per creare automaticamente un cluster Amazon EMR AWS Data Pipeline a 20 nodi per elaborare i dati e il tuo account AWS ha un limite di istanze EC2 impostato su 20, potresti inavvertitamente esaurire le risorse di backfill disponibili. Di conseguenza, è necessario considerare queste limitazioni in termini di risorse nel progetto oppure aumentare i limiti dell'account in base alle necessità. Per ulteriori informazioni sulle restrizioni dei servizi, consulta Restrizioni dei servizi AWS nella Guida di riferimento generale di AWS.

> **Note**
>
> Il limite è un'istanza per l'oggetto componente `Ec2Resource`.

Piattaforme supportate

Le pipeline possono avviare le tue risorse nelle seguenti piattaforme:

EC2-Classic

Le risorse vengono eseguite in una rete semplice, singola condivisa con altri clienti.

EC2-VPC

Le risorse vengono eseguite in un cloud privato virtuale (VPC, Virtual Private Cloud), logicamente limitato all'account AWS.

L'account AWS è in grado di avviare risorse in entrambe le piattaforme oppure solo in EC2-VPC, in base alle regioni. Per ulteriori informazioni, consulta Supported Platforms nella Amazon EC2 User Guide.

Se l'account AWS supporta solo EC2-VPC, è necessario creare un VPC di default in ciascuna regione AWS. Per impostazione predefinita, è necessario avviare le proprie risorse in una sottorete di

default del VPC predefinito. In alternativa, è possibile creare un VPC non predefinito e specificare una delle relative sottoreti quando si configurano le proprie risorse, quindi è necessario lanciare le proprie risorse nella sottorete specificata del VPC non predefinito.

Quando avvii un'istanza in un VPC, devi specificare un gruppo di sicurezza creato in modo specifico per quel VPC. Non è possibile specificare un gruppo di sicurezza creato per un EC2-Classic quando si avvia un'istanza in un VPC. Inoltre, è necessario utilizzare l'ID del gruppo di sicurezza e non il nome del gruppo di sicurezza per identificare un gruppo di sicurezza per un VPC.

Istanze Spot Amazon EC2 con cluster Amazon EMR e AWS Data Pipeline

Le pipeline possono utilizzare le istanze Spot di Amazon EC2 per i nodi di attività nelle risorse del cluster Amazon EMR. Per impostazione predefinita, le pipeline utilizzano le istanze on demand. Le istanze Spot consentono di usare le istanze EC2 inutilizzate ed eseguirle. Il modello di tariffazione delle istanze Spot è complementare a quello di istanze riservate e on demand, che possono potenzialmente offrire opzioni più convenienti per acquistare capacità di elaborazione, a seconda dell'applicazione per cui vengono impiegate. Per ulteriori informazioni, consulta la pagina di prodotto Istanze Spot di Amazon EC2.

Quando utilizzi istanze Spot, AWS Data Pipeline invia il prezzo massimo dell'istanza Spot ad Amazon EMR al momento dell'avvio del cluster. Alloca automaticamente il lavoro del cluster al numero di nodi di attività dell'istanza Spot che definisci utilizzando il campo. `taskInstanceCount` AWS Data Pipeline limita le istanze Spot per i nodi di attività per garantire che i nodi principali su richiesta siano disponibili per eseguire la pipeline.

È possibile modificare un'istanza di risorse di pipeline non riuscita o completata per aggiungere le istanze Spot. Quando la pipeline lancia di nuovo il cluster, utilizza le istanze Spot per i nodi di task.

Considerazioni sulle istanze Spot

Quando utilizzi le istanze Spot con AWS Data Pipeline, valgono le seguenti considerazioni:

- Le tue istanze Spot possono terminare quando il prezzo dell'istanza Spot supera il prezzo massimo per l'istanza o per motivi di capacità di Amazon EC2. Tuttavia, non perderai i tuoi dati perché AWS Data Pipeline utilizza cluster con nodi principali che sono sempre istanze on demand e non soggetti a chiusura.
- Le istanze Spot possono richiedere più tempo per l'avvio in quanto soddisfano la capacità in modo asincrono. Pertanto, una pipeline di un'istanza Spot può essere eseguita più lentamente rispetto a una pipeline equivalente di un'istanza on demand.

- Il cluster potrebbe non essere eseguito se non si ricevono le istanze Spot, ad esempio nel caso in cui il prezzo massimo sia troppo basso.

Azioni

AWS Data Pipeline le azioni sono le azioni eseguite da un componente della pipeline quando si verificano determinati eventi, ad esempio attività riuscite, fallite o tardive. Il campo di eventi di un'attività si riferisce a un'operazione, ad esempio un riferimento a `snsalarm` nel campo `onLateAction` di `EmrActivity`.

AWS Data Pipeline si affida alle notifiche di Amazon SNS come metodo principale per indicare lo stato delle pipeline e dei relativi componenti in modo automatico. Per ulteriori dettagli, consulta la pagina Amazon SNS. Oltre alle notifiche SNS, puoi utilizzare la AWS Data Pipeline console e la CLI per ottenere informazioni sullo stato della pipeline.

AWS Data Pipeline supporta le seguenti azioni:

SnsAlarm

Azione che invia una notifica SNS a un argomento basato sugli eventi `onSuccess`, `OnFail` e `onLateAction`.

Interruzione

Un'azione che attiva l'annullamento di un'attività in sospeso o non terminata, una risorsa o un nodo di dati. Non è possibile terminare azioni che includono `onSuccess`, `OnFail` o `onLateAction`.

Monitoraggio proattivo delle pipeline

Il modo migliore per rilevare problemi è monitorare le pipeline in modo proattivo sin dall'inizio. Puoi configurare i componenti della pipeline per informarti di determinate situazioni o eventi, ad esempio quando un componente della pipeline si guasta o non si avvia entro l'ora di inizio pianificata. AWS Data Pipeline semplifica la configurazione delle notifiche fornendo campi evento sui componenti della pipeline che puoi associare alle notifiche di Amazon SNS, `onSuccess` come`OnFail`, e. `onLateAction`

Configurazione per AWS Data Pipeline

Prima di utilizzarlo AWS Data Pipeline per la prima volta, completa le seguenti attività.

Processi

- Registrati per AWS
- Crea ruoli IAM per AWS Data Pipeline risorse e pipeline
- Consenti ai responsabili IAM (utenti e gruppi) di eseguire le azioni necessarie
- Concessione dell'accesso programmatico

Dopo aver completato queste attività, puoi iniziare a utilizzare AWS Data Pipeline. Per un tutorial di base, vedere Guida introduttiva con AWS Data Pipeline.

Registrati per AWS

Quando ti iscrivi ad Amazon Web Services (AWS), il tuo account AWS viene automaticamente registrato per tutti i servizi in AWS, inclusi AWS Data Pipeline. Ti vengono addebitati solo i servizi che utilizzi. Per ulteriori informazioni sui tassi di AWS Data Pipeline utilizzo, consulta AWS Data Pipeline.

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi Console di gestione AWScome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Accedere come utente root nella Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita il Centro identità IAM.

   Per istruzioni, consulta Abilitazione del AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. Nel Centro identità IAM, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere come utente del Centro identità IAM, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente del Centro identità IAM.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegnazione dell'accesso ad altri utenti

1. Nel Centro identità IAM, crea un set di autorizzazioni conforme alla best practice per l'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Crea ruoli IAM per AWS Data Pipeline risorse e pipeline

AWS Data Pipeline richiede ruoli IAM che determinano le autorizzazioni per eseguire azioni e accedere alle AWS risorse. Il ruolo pipeline determina le autorizzazioni di cui AWS Data Pipeline dispone e un ruolo di risorsa determina le autorizzazioni di cui dispongono le applicazioni in esecuzione su risorse della pipeline, come le istanze EC2. Questi ruoli vengono specificati quando si crea una pipeline. Anche se non specificate un ruolo personalizzato e utilizzate i ruoli `DataPipelineDefaultRole` predefiniti`DataPipelineDefaultResourceRole`, dovete prima creare i ruoli e allegare le politiche di autorizzazione. Per ulteriori informazioni, consulta Ruoli IAM per AWS Data Pipeline.

Consenti ai responsabili IAM (utenti e gruppi) di eseguire le azioni necessarie

Per lavorare con una pipeline, devi consentire a un responsabile IAM (un utente o un gruppo) del tuo account di eseguire AWS Data Pipeline le azioni e le azioni richieste per altri servizi, come definito dalla tua pipeline.

Per semplificare le autorizzazioni, è possibile allegare la policy AWSDataPipeline_FullAccessgestita ai principali IAM. Questa policy gestita consente al responsabile di eseguire tutte le azioni richieste dall'utente e l'`iam:PassRole`azione sui ruoli predefiniti utilizzati AWS Data Pipeline quando non è specificato un ruolo personalizzato.

Ti consigliamo vivamente di valutare attentamente questa politica gestita e di limitare le autorizzazioni solo a quelle richieste dagli utenti. Se necessario, utilizza questa policy come punto di partenza, quindi rimuovi le autorizzazioni per creare una policy di autorizzazioni in linea più restrittiva da collegare ai principali IAM. Per ulteriori informazioni ed esempi di politiche di autorizzazione, consulta Politiche di esempio per AWS Data Pipeline

Una dichiarazione politica simile all'esempio seguente deve essere inclusa in una policy allegata a qualsiasi principale IAM che utilizza la pipeline. Questa dichiarazione consente al principale IAM di eseguire l'`PassRole`azione sui ruoli utilizzati da una pipeline. Se non utilizzi ruoli predefiniti, sostituisci *`MyPipelineRole`* e *`MyResourceRole`* con i ruoli personalizzati che crei.

JSON

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

La procedura seguente illustra come creare un gruppo IAM, allegare la policy AWSDataPipeline_FullAccessgestita al gruppo e quindi aggiungere utenti al gruppo. È possibile utilizzare questa procedura per qualsiasi politica in linea

Per creare un gruppo di utenti **`DataPipelineDevelopers`** e allegare la politica AWSDataPipeline_FullAccess

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Groups (Gruppi), Create New group (Crea nuovo gruppo).
3. Inserisci un nome di gruppo, ad esempio**`DataPipelineDevelopers`**, e quindi scegli Passaggio successivo.

4. Immettete **`AWSDataPipeline_FullAccess`** Filtro, quindi selezionatelo dall'elenco.
5. Scegliere Next Step (Fase successiva), quindi scegliere Create Group (Crea gruppo).
6. Per aggiungere utenti al gruppo:
   a. Seleziona il gruppo che hai creato dall'elenco dei gruppi.
   b. Scegli Azioni di gruppo, Aggiungi utenti al gruppo.
   c. Seleziona gli utenti che desideri aggiungere dall'elenco, quindi scegli Aggiungi utenti al gruppo.

Concessione dell'accesso programmatico

Gli utenti necessitano dell'accesso programmatico se desiderano interagire con utenti AWS esterni a. Console di gestione AWS Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l’accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell’accesso programmatico?	Per	Come
IAM	(Consigliato) Utilizza le credenziali della console come credenziali temporanee per firmare le richieste programmatiche a,, o. AWS CLI AWS SDKs AWS APIs	Segui le istruzioni per l’interfaccia che desideri utilizzare. • Per la AWS CLI, consulta Login for AWS local development nella Guida per l'AWS Command Line Interface utente. • Per AWS SDKs, consulta Login for AWS local development nella AWS SDKs and Tools Reference Guide.
Identità della forza lavoro	Utilizza credenziali temporanee per firmare le richieste	Segui le istruzioni per l’interfaccia che desideri utilizzare.

Quale utente necessita dell'accesso programmatico?	Per	Come
(Utenti gestiti nel centro identità IAM)	programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	• Per la AWS CLI, vedere Configurazione dell'uso AWS IAM Identity Center nella AWS CLI Guida per l'utente.AWS Command Line Interface • Per AWS SDKs gli strumenti e AWS APIs, consulta l'autenticazione di IAM Identity Center nella Guida di riferimento AWS SDKs and Tools.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	Seguendo le istruzioni riportate in Utilizzo delle credenziali temporanee con le AWS risorse nella Guida per l'utente IAM.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare richieste programmatiche a AWS CLI,, AWS SDKs o. AWS APIs	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli strumenti AWS SDKs e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli strumenti e agli AWS SDKs strumenti. • Per AWS APIs, consulta la sezione Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Guida introduttiva con AWS Data Pipeline

AWS Data Pipeline ti aiuta a sequenziare, pianificare, eseguire e gestire carichi di lavoro ricorrenti di elaborazione dati in modo affidabile ed economico. Questo servizio semplifica la progettazione di attività extract-transform-load (ETL) utilizzando dati strutturati e non strutturati, sia in locale che nel cloud, in base alla logica aziendale.

Per utilizzarlo AWS Data Pipeline, crei una definizione di pipeline che specifica la logica di business per l'elaborazione dei dati. Una tipica definizione di pipeline è costituita da attività che definiscono il lavoro da eseguire e da nodi di dati che definiscono la posizione e il tipo di dati di input e output.

In questo tutorial, si esegue lo script di un comando shell che conta il numero di richieste GET nei log del server Web Apache. Questa pipeline viene eseguita ogni 15 minuti per un'ora e scrive l'output su Amazon S3 a ogni iterazione.

Prerequisiti

Prima di iniziare, completa le attività in Configurazione per AWS Data Pipeline.

Oggetti della pipeline

La pipeline utilizza i seguenti oggetti:

ShellCommandActivity

Legge i file di log di input e conta il numero di errori.

S3 DataNode (input)

Bucket S3 che contiene il file di log di input.

S3 DataNode (output)

Bucket S3 per l'output.

Ec2Resource

La risorsa di calcolo AWS Data Pipeline utilizzata per eseguire l'attività.

Tieni presente che se disponi di una grande quantità di dati dei file di registro, puoi configurare la pipeline per utilizzare un cluster EMR per elaborare i file anziché EC2 un'istanza.

Schedule

Stabilisce che l'attività venga eseguita ogni 15 minuti per un'ora.

Processi

Per creare la pipeline

Il modo più rapido per iniziare AWS Data Pipeline è utilizzare una definizione di pipeline chiamata modello.

Per creare la pipeline

1. Apri la AWS Data Pipeline console all'indirizzo. https://console.aws.amazon.com/datapipeline/
2. Nella barra di navigazione, selezionare una regione. È possibile selezionare qualsiasi regione disponibile, indipendentemente dalla posizione. Molte risorse AWS sono specifiche per una regione, ma AWS Data Pipeline consentono di utilizzare risorse che si trovano in una regione diversa rispetto alla pipeline.
3. La prima schermata che vedi dipende dal fatto che tu abbia creato una pipeline nella regione corrente.
    a. Se non hai creato una pipeline in questa regione, la console visualizza una schermata introduttiva. Scegli Inizia subito.
    b. Se hai già creato una pipeline in questa regione, la console visualizza una pagina che elenca le pipeline per la regione. Scegli Crea nuova pipeline.
4. In Nome, inserisci un nome per la pipeline.
5. (Facoltativo) In Descrizione, inserisci una descrizione per la pipeline.
6. Per Source, seleziona Crea usando un modello, quindi seleziona il seguente modello: Guida introduttiva all'uso ShellCommandActivity.
7. Nella sezione Parameters (Parametri) che si è aperta quando è stato selezionato il modello, lasciare i valori predefiniti nella S3 input folder (cartella di input S3) e nel Shell command to run (Comando Shell da eseguire). Fare clic sull'icona della cartella accanto a S3 output folder (Cartella di output S3), selezionare uno dei bucket o delle cartelle, quindi fare clic su Select (Seleziona).

8. In Schedule (Pianificazione), lasciare i valori predefiniti. Quando si attiva la pipeline, viene eseguito l'avvio della pipeline che si ripete ogni 15 minuti per un'ora.

   Se si preferisce, è possibile selezionare Run once on pipeline activation (Esegui una volta all'attivazione della pipeline).

9. In Pipeline Configuration, lascia la registrazione abilitata. Scegli l'icona della cartella nella posizione S3 per i log, seleziona uno dei bucket o delle cartelle, quindi scegli Seleziona.

   Se preferisci, puoi invece disabilitare la registrazione.

10. In Security/Access, lascia i ruoli IAM impostati su Predefiniti.

11. Fai clic su Activate (Attiva).

    Se preferisci, puoi scegliere Modifica in Architect per modificare questa pipeline. Ad esempio, puoi aggiungere condizioni preliminari.

Monitorare la pipeline in esecuzione

Dopo aver attivato la pipeline, visualizzare la pagina Execution details (Dettagli esecuzione), dove è possibile monitorare l'avanzamento della pipeline.

Per monitorare l'avanzamento della pipeline

1. Fare clic su Update (Aggiorna) o premere F5 per aggiornare lo stato visualizzato.

   > **Tip**
   >
   > Se non vi sono esecuzioni elencate, verificare che Start (in UTC) (Inizio (in UTC)) e End (in UTC) (Fine (in UTC)) coprano l'inizio e la fine pianificati della pipeline, quindi selezionare Update (Aggiorna).

2. Quando lo stato di ogni oggetto nella pipeline è `FINISHED`, significa che la tua pipeline ha completato correttamente le attività pianificate.

3. Se la pipeline non viene completata correttamente, verifica se vi sono problemi con le impostazioni della pipeline. Per ulteriori informazioni sulla risoluzione di problemi con istanze della pipeline non eseguite o non completate, consulta Risoluzione dei problemi più comuni.

Visualizzazione dell'output

Apri la console Amazon S3 e accedi al tuo bucket. Se si esegue la pipeline ogni 15 minuti per un'ora, verranno visualizzate quattro sottocartelle con time-stamp. Ogni sottocartella contiene l'output in un file denominato `output.txt`. Poiché ogni volta lo script è stato eseguito sullo stesso file di input, i file di output sono identici.

Per eliminare la pipeline

Per evitare di incorrere in addebiti, elimina la pipeline. L'eliminazione della pipeline comporta l'eliminazione della definizione della pipeline e di tutti gli oggetti associati.

Per eliminare la pipeline

1. Nella pagina Elenca tubazioni, seleziona la pipeline.
2. Fai clic su Azioni, quindi scegli Elimina.
3. Quando viene richiesta la conferma, seleziona Elimina.

Se hai finito con l'output di questo tutorial, elimina le cartelle di output dal tuo bucket Amazon S3.

Uso delle pipeline

È possibile amministrare, creare e modificare le pipeline utilizzando l'interfaccia a riga di comando (CLI) o l'SDK. AWS Le seguenti sezioni introducono concetti fondamentali AWS Data Pipeline e illustrano come lavorare con le pipeline.

> **Important**
>
> Prima di iniziare, consulta Configurazione per AWS Data Pipeline.

Indice

Creazione di una pipeline

AWS Data Pipeline offre diversi modi per creare pipeline:

- Utilizza AWS Command Line Interface (CLI) con un modello fornito per comodità. Per ulteriori informazioni, consulta Crea una pipeline dai modelli di Data Pipeline utilizzando la CLI.
- Utilizza AWS Command Line Interface (CLI) con un file di definizione della pipeline in formato JSON.

- Utilizzare un SDK AWS con un'API specifica per il linguaggio. Per ulteriori informazioni, consulta Lavorare con l'API.

Crea una pipeline dai modelli di Data Pipeline utilizzando la CLI

Data Pipeline fornisce diverse definizioni di pipeline preconfigurate, note come modelli. Puoi utilizzare i modelli per iniziare rapidamente. AWS Data Pipeline Questi modelli sono disponibili in un bucket pubblico presso la sede Amazon S3:. `s3://datapipeline-us-east-1/templates/` Questi modelli predefiniti vengono creati per raggiungere casi d'uso specifici e possono essere utilizzati per creare pipeline. È possibile utilizzare `aws s3 ls --recursive "s3://datapipeline-us-east-1/templates/"` per elencare tutti i modelli disponibili.

Crea una pipeline da un modello utilizzando la CLI

Supponiamo di voler creare una pipeline che esporti una tabella DynamoDB in Amazon S3. Il modello da utilizzare in questo caso è disponibile all'indirizzo:. `s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json`

Per scaricare il modello JSON e creare una pipeline utilizzando la CLI

1. Scarica il modello utilizzando la `aws s3 cp` CLI o curl. Esempio:

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

2. Apporta le modifiche necessarie al modello scaricato. Ad esempio, per utilizzare l'ultima versione di EMR, modificare il `releaseLabel` campo nell'`EmrClusterForBackup`oggetto, modificare i tipi di istanza master e core e modificare i valori predefiniti dei parametri nel modello.
3. Crea una pipeline utilizzando la `create-pipeline` CLI. Esempio:

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

4. Annota l'ID della pipeline creata.
5. Utilizzare `put-pipeline-definition` per caricare la definizione. Fornisci i valori dei parametri di cui desideri sovrascrivere i valori predefiniti utilizzando l'`--parameter-values`opzione.

Per ulteriori informazioni sui modelli, consulta Choose a template (Scegli un modello).

Choose a template (Scegli un modello)

I seguenti modelli sono disponibili per il download dal bucket Amazon S3:. `s3://datapipeline-us-east-1/templates/`

Modelli

- Guida introduttiva a utilizzare ShellCommandActivity
- Esegui il comando AWS CLI
- Esporta la tabella DynamoDB in S3
- Importazione dei dati di backup DynamoDB da S3
- Esegui job su un cluster Amazon EMR
- Copia completa di Amazon RDS MySQL Table su Amazon S3
- Copia incrementale della tabella Amazon RDS MySQL su Amazon S3
- Carica i dati S3 nella tabella Amazon RDS MySQL
- Copia completa della tabella Amazon RDS MySQL su Amazon Redshift
- Copia incrementale di una tabella Amazon RDS MySQL su Amazon Redshift
- Caricare dati da Amazon S3 in Amazon Redshift

Guida introduttiva a utilizzare ShellCommandActivity

Il ShellCommandActivity modello Getting Started using esegue uno script di comandi di shell per contare il numero di richieste GET in un file di registro. L'output viene scritto in una posizione Amazon S3 con data e ora su ogni esecuzione pianificata della pipeline.

Il modello utilizza i seguenti oggetti della pipeline:

- ShellCommandActivity
- S3 InputNode
- S3 OutputNode
- Ec2Resource

Esegui il comando AWS CLI

Questo modello esegue un AWS CLI comando specificato dall'utente a intervalli pianificati.

Esporta la tabella DynamoDB in S3

Il modello Esporta tabella DynamoDB in S3 pianifica un cluster Amazon EMR per esportare dati da una tabella DynamoDB a un bucket Amazon S3. Questo modello utilizza un cluster Amazon EMR, che è dimensionato proporzionalmente al valore del throughput disponibile per la tabella DynamoDB. Sebbene sia possibile aumentare il numero di una tabella, ciò potrebbe comportare costi aggiuntivi IOPs durante l'importazione e l'esportazione. In precedenza, l'esportazione utilizzava un, HiveActivity ma ora utilizza la modalità nativa. MapReduce

Il modello utilizza i seguenti oggetti della pipeline:

- EmrActivity
- EmrCluster
- DBDataNodo Dynamo
- S3 DataNode

Importazione dei dati di backup DynamoDB da S3

Il modello Importa dati di backup DynamoDB da S3 pianifica un cluster Amazon EMR per caricare un backup DynamoDB creato in precedenza in Amazon S3 su una tabella DynamoDB. Gli elementi esistenti nella tabella DynamoDB vengono aggiornati con quelli dei dati di backup e nuovi elementi vengono aggiunti alla tabella. Questo modello utilizza un cluster Amazon EMR, che è dimensionato proporzionalmente al valore del throughput disponibile per la tabella DynamoDB. Sebbene sia possibile aumentare il numero di una tabella, ciò potrebbe comportare costi aggiuntivi IOPs durante l'importazione e l'esportazione. In precedenza, l'importazione utilizzava un metodo nativo, HiveActivity ma ora viene utilizzato il formato nativo. MapReduce

Il modello utilizza i seguenti oggetti della pipeline:

- EmrActivity
- EmrCluster
- DBDataNodo Dynamo
- S3 DataNode

- S3 PrefixNotEmpty

Esegui job su un cluster Amazon EMR

Il modello Run Job on an Elastic MapReduce Cluster avvia un cluster Amazon EMR in base ai parametri forniti e avvia l'esecuzione delle fasi in base alla pianificazione specificata. Una volta completato il processo, il cluster EMR viene terminato. Le operazioni di bootstrap opzionali possono essere specificate per installare software aggiuntivo o modificare la configurazione di applicazioni nel cluster.

Il modello utilizza i seguenti oggetti della pipeline:

- EmrActivity
- EmrCluster

Copia completa di Amazon RDS MySQL Table su Amazon S3

Il modello Full Copy of RDS MySQL Table to S3 copia un'intera tabella Amazon RDS MySQL e archivia l'output in una posizione Amazon S3. L'output viene archiviato come file CSV in una sottocartella con data e ora nella posizione Amazon S3 specificata.

Il modello utilizza i seguenti oggetti della pipeline:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Copia incrementale della tabella Amazon RDS MySQL su Amazon S3

Il modello Copia incrementale di RDS MySQL Table to S3 esegue una copia incrementale dei dati da una tabella Amazon RDS MySQL e archivia l'output in una posizione Amazon S3. La tabella Amazon RDS MySQL deve avere una colonna Last Modified.

Questo modello copia le modifiche apportate alla tabella tra intervalli pianificati a partire dalla data di inizio pianificata. Il tipo di pianificazione è una serie temporale, quindi se una copia è stata pianificata per una determinata ora, AWS Data Pipeline copia le righe della tabella che hanno

un timestamp dell'ultima modifica che rientra nell'ora. Le eliminazioni fisiche fatte alla tabella non vengono copiate. L'output viene scritto in una sottocartella con data e ora nella posizione Amazon S3 a ogni esecuzione pianificata.

Il modello utilizza i seguenti oggetti della pipeline:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Carica i dati S3 nella tabella Amazon RDS MySQL

Il modello Load S3 Data into RDS MySQL Table pianifica un'istanza Amazon EC2 per copiare il file CSV dal percorso del file Amazon S3 specificato di seguito in una tabella Amazon RDS MySQL. Il file CSV non deve avere una riga di intestazione. Il modello aggiorna le voci esistenti nella tabella Amazon RDS MySQL con quelle nei dati Amazon S3 e aggiunge nuove voci dai dati Amazon S3 alla tabella Amazon RDS MySQL. È possibile caricare i dati in una tabella esistente o fornire una query SQL per creare una nuova tabella.

Il modello utilizza i seguenti oggetti della pipeline:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Modelli da Amazon RDS ad Amazon Redshift

I due modelli seguenti copiano le tabelle da Amazon RDS MySQL ad Amazon Redshift utilizzando uno script di traduzione, che crea una tabella Amazon Redshift utilizzando lo schema della tabella di origine con le seguenti avvertenze:

- Se non viene specificata una chiave di distribuzione, la prima chiave primaria della tabella Amazon RDS viene impostata come chiave di distribuzione.
- Non puoi saltare una colonna presente in una tabella Amazon RDS MySQL quando esegui una copia su Amazon Redshift.

- (Facoltativo) Puoi fornire una mappatura del tipo di dati delle colonne da Amazon RDS MySQL ad Amazon Redshift come uno dei parametri del modello. Se viene specificato, lo script lo utilizza per creare la tabella Amazon Redshift.

Se viene utilizzata la modalità di inserimento di `Overwrite_Existing` Amazon Redshift:

- Se non viene fornita una chiave di distribuzione, viene utilizzata una chiave primaria nella tabella Amazon RDS MySQL.
- Se ci sono chiavi primarie composite nella tabella, la prima viene usata come chiave di distribuzione se la chiave di distribuzione non viene fornita. Solo la prima chiave composita viene impostata come chiave primaria nella tabella Amazon Redshift.
- Se non viene fornita una chiave di distribuzione e non esiste una chiave primaria nella tabella Amazon RDS MySQL, l'operazione di copia non riesce.

Per ulteriori informazioni su Amazon Redshift, consulta i seguenti argomenti:

- Cluster Amazon Redshift
- Amazon Redshift COPY
- Stili di distribuzione ed esempi DISTKEY
- Chiavi ordinamento

La tabella seguente spiega come lo script traduce i tipi di dati:

Traduzioni dei tipi di dati tra MySQL e Amazon Redshift

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
TINYINT, TINYINT (dimensioni)	SMALLINT	MySQL: -da 128 a 127. Il numero massimo di cifre può essere specificato tra parentesi. Amazon Redshift:. INT2 Intero a due byte firmato
TINYINT UNSIGNED,	SMALLINT	MySQL: da 0 a 255 UNSIGNED. Il numero

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
TINYINT (dimensione) UNSIGNED		massimo di cifre può essere specificato tra parentesi. Amazon Redshift:. INT2 Intero a due byte firmato
SMALLINT, SMALLINT (dimensioni)	SMALLINT	MySQL: - da 32768 a 32767 normale. Il numero massimo di cifre può essere specificato tra parentesi. Amazon Redshift:. INT2 Intero a due byte firmato
SMALLINT UNSIGNED, SMALLINT(dimensione) UNSIGNED,	INTEGER	MySQL: da 0 a 65535 UNSIGNED*. Il numero massimo di cifre può essere specificato tra parentesi Amazon Redshift:. INT4 Intero a quattro byte firmato
MEDIUMINT, MEDIUMINT(dimensione)	INTEGER	MySQL: da 388608 a 8388607. Il numero massimo di cifre può essere specificato tra parentesi Amazon Redshift:. INT4 Intero a quattro byte firmato
MEDIUMINT UNSIGNED, MEDIUMINT(dimensione) UNSIGNED	INTEGER	MySQL: da 0 a 16777215. Il numero massimo di cifre può essere specificato tra parentesi Amazon Redshift:. INT4 Intero a quattro byte firmato

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
INT, INT(dimensione)	INTEGER	MySQL: da 147483648 a 2147483647 Amazon Redshift:. INT4 Intero a quattro byte firmato
INT UNSIGNED, INT(dimensione) UNSIGNED	BIGINT	MySQL: da 0 a 4294967295 Amazon Redshift:. INT8 Intero a otto byte firmato
BIGINT BIGINT (dimensione)	BIGINT	Amazon Redshift:. INT8 Intero a otto byte firmato
BIGINT UNSIGNED BIGINT(dimensione) UNSIGNED	VARCHAR(20*4)	MySQL: da 0 a 18446744073709551615 Amazon Redshift: nessun equivalente nativo, quindi utilizza un array di caratteri.
FLOAT FLOAT (dimensioni, d) FLOAT (dimensioni, d) UNSIGNED	REAL	Il numero massimo di cifre può essere specificato nel parametro della dimensione. Il numero di cifre alla destra del punto decimale viene specificato nel parametro d. Amazon Redshift: FLOAT4

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
DOUBLE (dimensioni, d)	DOUBLE PRECISION	Il numero massimo di cifre può essere specificato nel parametro della dimensione. Il numero di cifre alla destra del punto decimale viene specifica to nel parametro d. Amazon Redshift: FLOAT8
DECIMAL (dimensioni, d)	DECIMAL (dimensioni, d)	Un DOUBLE memorizzato come stringa, consentendo una virgola decimale fissa. Il numero massimo di cifre può essere specificato nel parametro della dimensione. Il numero di cifre alla destra del punto decimale viene specifica to nel parametro d. Amazon Redshift: nessun equivalente nativo.

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
CHAR(dimensione)	VARCHAR (dimensione* 4)	Contiene una stringa di lunghezza fissa, che può contenere lettere, numeri e caratteri speciali. La dimensione fissa viene specificata come parametro tra parentesi. Può contenere fino a 255 caratteri. Riempito a destra con spazi. Amazon Redshift: il tipo di dati CHAR non supporta caratteri multibyte, quindi viene utilizzato VARCHAR. Il numero massimo di byte per carattere è 4 in base a, il che limita la tabella dei caratteri a RFC3629U+10FFFF.
VARCHAR(dimensione)	VARCHAR (dimensione* 4)	Può contenere fino a 255 caratteri. VARCHAR non supporta i seguenti punti di codice UTF-8 non validi: 0xD800- 0xDFFF, (sequenze di byte: ED A0 80- ED BF BF), 0x FDD0 - 0xFDEF, 0xFFFE e 0xFFFF, (sequenze di byte: EF B7 90- EF B7 AF, EF BF BE e EF BF BF)
TINYTEXT	VARCHAR(255*4)	Contiene una stringa con un massimo di 255 caratteri

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
TEXT	VARCHAR(max)	Contiene una stringa con un massimo di 65.535 caratteri.
MEDIUMTEXT	VARCHAR(max)	Da 0 a 16.777.215 char
LONGTEXT	VARCHAR(max)	Da 0 a 4.294.967.295 char
BOOLEAN BOOL TINYINT(1)	BOOLEAN	MySQL: questi tipi sono sinonimi di TINYINT (1). Il valore zero è considerato falso. I valori diversi da zero sono considerati veri.
BINARY[(M)]	varchar(255)	M è da 0 a 255 byte, FIXED
VARBINARY(M)	VARCHAR(max)	Da 0 a 65.535 byte
TINYBLOB	VARCHAR(255)	Da 0 a 255 byte
BLOB	VARCHAR(max)	Da 0 a 65.535 byte
MEDIUMBLOB	VARCHAR(max)	Da 0 a 16.777.215 byte
LONGBLOB	VARCHAR(max)	Da 0 a 4.294.967.295 byte
ENUM	VARCHAR(255*2)	Il limite non è alla lunghezza della stringa di enumerazione letterale, bensì sulla definizione della tabella per il numero di valori di enumerazione.
SET	VARCHAR(255*2)	Come enum.
DATE	DATE	(YYYY-MM-DD) da "1000-01-01" a "9999-12-31"

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
TIME	VARCHAR(10*4)	(hh:mm:ss) da "-838:59:59" a "838:59:59"
DATETIME	TIMESTAMP	(YYYY-MM-DD hh:mm:ss) da 1000-01-01 00:00:00" a "9999-12-31 23:59:59"
TIMESTAMP	TIMESTAMP	(YYYYMMDDhhmmss) da 19700101000000 a 2037+
ANNO	VARCHAR(4*4)	(YYYY) Da 1900 a 2155
colonna SERIAL	generazione di ID/Questo attributo non è necessario per un data warehouse OLAP poiché questa colonna è copiata. La parola chiave SERIAL non viene aggiunta durante la traduzione.	SERIAL è un'entità denominat a SEQUENCE. Esiste indipendentemente dal resto della tabella. colonna GENERATED BY DEFAULT equivalente a: Nome CREATE SEQUENCE; tabella CREATE TABLE (colonna INTEGER NOT NULL DEFAULT nextval (name));

Tipi di dati MySQL	Tipo di dati Amazon Redshift	Note
colonna BIGINT UNSIGNED NOT NULL AUTO_INCR EMENT UNIQUE	generazione di ID/Questo attributo non è necessario per un data warehouse OLAP poiché questa colonna è copiata. Quindi, la parola chiave SERIAL non viene aggiunta durante la traduzione.	SERIAL è un'entità denominat a SEQUENCE. Esiste indipendentemente dal resto della tabella. colonna GENERATED BY DEFAULT equivalente a: Nome CREATE SEQUENCE; tabella CREATE TABLE (colonna INTEGER NOT NULL DEFAULT nextval (name));
ZEROFILL	La parola chiave ZEROFILL non viene aggiunta durante la traduzione.	INT UNSIGNED ZEROFILL NOT NULL ZEROFILL riempie il valore visualizzato del campo con zeri fino alla larghezza di visualizzazione specificata nella definizione della colonna. I valori superiori alla larghezza di visualizzazione non sono troncati. Si noti che l'utilizzo di ZEROFILL implica anche UNSIGNED.

Copia completa della tabella Amazon RDS MySQL su Amazon Redshift

La copia completa della tabella Amazon RDS MySQL nel modello Amazon Redshift copia l'intera tabella Amazon RDS MySQL in una tabella Amazon Redshift inserendo i dati in una cartella Amazon S3. La cartella di staging di Amazon S3 deve trovarsi nella stessa regione del cluster Amazon Redshift. Una tabella Amazon Redshift viene creata con lo stesso schema della tabella Amazon RDS

MySQL di origine se non esiste già. Fornisci eventuali sostituzioni del tipo di dati delle colonne da Amazon RDS MySQL ad Amazon Redshift che desideri applicare durante la creazione di tabelle Amazon Redshift.

Il modello utilizza i seguenti oggetti della pipeline:

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

Copia incrementale di una tabella Amazon RDS MySQL su Amazon Redshift

La copia incrementale della tabella Amazon RDS MySQL nel modello Amazon Redshift copia i dati da una tabella Amazon RDS MySQL a una tabella Amazon Redshift posizionando i dati in una cartella Amazon S3.

La cartella di staging di Amazon S3 deve trovarsi nella stessa regione del cluster Amazon Redshift.

AWS Data Pipeline utilizza uno script di traduzione per creare una tabella Amazon Redshift con lo stesso schema della tabella Amazon RDS MySQL di origine, se non esiste già. È necessario fornire tutte le sostituzioni dei tipi di dati delle colonne da Amazon RDS MySQL ad Amazon Redshift che desideri applicare durante la creazione di tabelle Amazon Redshift.

Questo modello copia le modifiche apportate alla tabella Amazon RDS MySQL tra intervalli pianificati, a partire dall'ora di inizio pianificata. Le eliminazioni fisiche nella tabella Amazon RDS MySQL non vengono copiate. È necessario specificare il nome della colonna che memorizza l'ultimo valore temporale modificato.

Quando utilizzi il modello predefinito per creare pipeline per la copia incrementale di Amazon RDS, viene creata un'attività con il nome `RDSToS3CopyActivity` predefinito. È possibile assegnarle un nome diverso.

Il modello utilizza i seguenti oggetti della pipeline:

- CopyActivity

- RedshiftCopyActivity
- S3 DataNode
- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

Caricare dati da Amazon S3 in Amazon Redshift

Il modello Load data from S3 to Redshift copia i dati da una cartella Amazon S3 in una tabella Amazon Redshift. È possibile caricare i dati in una tabella esistente o fornire una query SQL per creare una tabella.

I dati vengono copiati in base alle opzioni di Amazon `COPY` Redshift. La tabella Amazon Redshift deve avere lo stesso schema dei dati in Amazon S3. Per `COPY` le opzioni, consulta COPY nella Amazon Redshift Database Developer Guide.

Il modello utilizza i seguenti oggetti della pipeline:

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

Creazione di una pipeline utilizzando modelli parametrizzati

È possibile utilizzare un modello parametrizzato per personalizzare una definizione di pipeline. In questo modo è possibile creare una definizione di pipeline comune offrendo diversi parametri quando si aggiunge la definizione di pipeline a una nuova pipeline.

Indice

- Definire i valori di parametro
- Invio della definizione della pipeline

Aggiungi MyVariables alla definizione della pipeline

Quando create il file di definizione della pipeline, specificate le variabili utilizzando la seguente sintassi: # {my}. *Variable* È necessario che la variabile abbia il prefisso my. Ad esempio, il seguente file di definizione della pipeline include le seguenti variabili:*myShellCmd*, *myS3InputLoc* e. pipeline-definition.json *myS3OutputLoc*

> **Note**
>
> Una definizione di pipeline dispone di un limite massimo di 50 parametri.

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

Definire gli oggetti parametrici

È possibile creare un file separato con gli oggetti dei parametri che definisca le variabili della definizione della pipeline. Ad esempio, il seguente file JSON contiene oggetti parametrici per e *myS3OutputLoc* variabili tratte dalla definizione di pipeline di esempio riportata sopra. `parameters.json` *myShellCmd* *myS3InputLoc*

```
{
  "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**
>
> È possibile aggiungere questi oggetti direttamente al file di definizione della pipeline invece di utilizzare un file separato.

La tabella seguente descrive gli attributi per gli oggetti dei parametri.

Attributi dei parametri

Attributo	Tipo	Description
`id`	Stringa	Identificatore univoco del parametro. Per mascherare il valore mentre è digitato o visualizzato, aggiungere un asterisco (*) come prefisso. Ad esempio, `*myVariable` —. Da notare che questo, inoltre, crittografa il valore prima di

Attributo	Tipo	Description
		essere memorizzato da AWS Data Pipeline.
description	Stringa	Descrizione del parametro.
tipo	Stringa, numero intero, doppio o AWS::S3::ObjectKey	Il tipo di parametro che definisce l'intervallo consentito o di valori di input e regole di convalida. L'impostazione predefinita è Stringa.
facoltativo	Booleano	Indica se il parametro è obbligatorio o facoltativo. Il valore predefinito è `false`.
allowedValues	Elenco di stringhe	Enumera tutti i valori consentiti per il parametro.
default	Stringa	Il valore predefinito per il parametro. Se si specifica un valore per questo parametro utilizzando i valori dei parametri, sostituisce il valore di default.
isArray	Booleano	Indica se il parametro è un array.

Definire i valori di parametro

È possibile creare un file separato per definire le variabili utilizzando i valori dei parametri. Ad esempio, il seguente file JSON contiene il valore della *`myS3OutputLoc`* variabile della definizione di pipeline di esempio riportata sopra. `file://values.json`

```
{
  "values":
    {
```

```
      "myS3OutputLoc": "myOutputLocation"
    }
}
```

Invio della definizione della pipeline

Quando si invia la definizione di pipeline, è possibile specificare i parametri, gli oggetti dei parametri e i valori dei parametri. Ad esempio, è possibile utilizzare il put-pipeline-definition AWS CLI comando come segue:

```
$ 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**
>
> Una definizione di pipeline dispone di un limite massimo di 50 parametri. La dimensione del file per `parameter-values-uri` dispone di un limite massimo di 15 KB.

Visualizzazione delle pipeline

È possibile visualizzare le pipeline utilizzando l'interfaccia a riga di comando (CLI).

Per visualizzare le tue pipeline utilizzando il AWS CLI

- Utilizzare il comando list-pipelines per elencare le pipeline:

  ```
  aws datapipeline list-pipelines
  ```

Interpretazione dei codici sullo stato della pipeline

I livelli di stato visualizzati nella AWS Data Pipeline console e nella CLI indicano le condizioni di una pipeline e dei suoi componenti. Lo stato della pipeline è semplicemente una panoramica di una pipeline; per visualizzare ulteriori informazioni, visualizzare lo stato dei singoli componenti della pipeline.

Una pipeline ha uno stato `SCHEDULED` se è pronto (la definizione di pipeline ha passato la convalida), al momento è in esecuzione il lavoro, oppure l'esecuzione del lavoro è terminata. Una pipeline con lo

stato PENDING se non è attivato o non è in grado di eseguire il lavoro (ad esempio, la convalida della definizione della pipeline non è riuscita).

Una pipeline è considerata inattiva se il suo stato è PENDING, INACTIVE o FINISHED. Alle pipeline inattive verrà applicato un costo (per ulteriori informazioni, consulta la pagina Prezzi).

Codici di stato

ACTIVATING

Il componente o la risorsa è in fase di avvio, ad esempio un'istanza EC2.

CANCELED

Il componente è stato annullato da un utente o AWS Data Pipeline prima che potesse essere eseguito. Ciò può avvenire automaticamente quando si verifica un errore in un altro componente o risorsa da cui dipende questo componente.

CASCADE_FAILED

Il componente o la risorsa è stato annullato a causa di un errore a cascata dovuto a una delle sue dipendenze, ma probabilmente non era il componente all'origine dell'errore.

DEACTIVATING

La pipeline viene disattivata.

FAILED

Il componente o la risorsa ha riscontrato un errore e ha smesso di funzionare. Quando un componente o una risorsa si guasta, possono verificarsi annullamenti ed errori che si ripercuotono su altri componenti che dipendono da esso.

FINISHED

Il componente ha completato il lavoro assegnato.

INACTIVE

La pipeline è stata disattivata.

PAUSED

Il componente è stato messo in pausa e attualmente non sta funzionando.

PENDING

La pipeline è pronta per essere attivata per la prima volta.

`RUNNING`

La risorsa è in esecuzione e pronta per ricevere lavoro.

`SCHEDULED`

L'esecuzione della risorsa è pianificata.

`SHUTTING_DOWN`

La risorsa si spegne dopo aver completato con successo il suo lavoro.

`SKIPPED`

Il componente ha saltato gli intervalli di esecuzione dopo l'attivazione della pipeline utilizzando un timestamp successivo alla pianificazione corrente.

`TIMEDOUT`

La risorsa ha superato la `terminateAfter` soglia ed è stata interrotta da. AWS Data Pipeline Dopo che la risorsa ha raggiunto questo stato, AWS Data Pipeline ignora i `actionOnResourceFailure` `retryTimeout` valori e e per quella risorsa. `retryDelay` Questo stato si applica solo alle risorse.

`VALIDATING`

La definizione della pipeline viene convalidata da. AWS Data Pipeline

`WAITING_FOR_RUNNER`

Il componente è in attesa che il suo client di lavoro recuperi un elemento di lavoro. La relazione tra componente e cliente-lavoratore è controllata dai `workerGroup` campi `runsOn` o dai campi definiti da quel componente.

`WAITING_ON_DEPENDENCIES`

Il componente sta verificando che le precondizioni predefinite e configurate dall'utente siano soddisfatte prima di eseguire il suo lavoro.

Interpretazione dello stato di pipeline e componenti

Ogni pipeline e componente all'interno di quella pipeline restituisce uno stato di integrità di `HEALTHY`, `ERROR`, `'-'`, `No Completed Executions` o `No Health Information Available`. Una pipeline dispone di un solo stato di integrità dopo che un componente della pipeline ha completato la prima esecuzione o se le precondizioni del componente hanno dato esito negativo. Lo stato di

integrità per i componenti aggregati in uno stato di integrità della pipeline negli stati di errore è visibile prima, al momento della visualizzazione dei dettagli di esecuzione.

Stati di integrità delle pipeline

`HEALTHY`

Lo stato di integrità aggregato di tutti i componenti è `HEALTHY`. Questo significa che almeno un componente deve essere completato con successo. È possibile fare clic sullo stato `HEALTHY` per vedere l'istanza dei componenti della pipeline più recente completata con successo nella pagina di Execution Details (Dettagli di esecuzione).

`ERROR`

Almeno un componente della pipeline dispone di un stato di integrità di `ERROR`. È possibile fare clic sullo stato `ERROR` per vedere l'istanza dei componenti della pipeline fallita più di recente nella pagina di Execution Details (Dettagli di esecuzione).

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

Nessun stato di integrità è stato segnalato per questa pipeline.

> **Note**
>
> Mentre i componenti aggiornano lo stato di integrità quasi immediatamente, possono essere necessari fino a cinque minuti per aggiornare lo stato di integrità di una pipeline.

Stati di integrità dei componenti

`HEALTHY`

Un componente (`Activity` o `DataNode`) ha uno stato di integrità di `HEALTHY` nel caso in cui una esecuzione viene completata con successo, contrassegnata con stato `FINISHED` o `MARK_FINISHED`. È possibile fare clic sul nome del componente o sullo stato `HEALTHY` per vedere le istanze dei componenti della pipeline più recenti completati con successo nella pagina di Execution Details (Dettagli di esecuzione).

`ERROR`

Si è verificato un errore a livello di componente o una delle precondizioni non è riuscita. Gli stati `FAILED`, `TIMEOUT` oppure `CANCELED` attivano questo errore. È possibile fare clic sul nome del

componente o sullo stato `ERROR` per vedere le istanze dei componenti della pipeline più recenti non riusciti nella pagina di Execution Details (Dettagli di esecuzione).

`No Completed Executions` o `No Health Information Available`

Nessun stato di integrità è stato segnalato per questo componente.

Visualizzazione delle definizioni di pipeline

Utilizza l'interfaccia a riga di comando (CLI) per visualizzare la definizione della pipeline. La CLI stampa un file di definizione della pipeline, in formato JSON. Per ulteriori informazioni sulla sintassi e l'utilizzo di file di definizione della pipeline, vedere Sintassi del file di definizione della pipeline.

Quando si utilizza la CLI, è consigliabile recuperare la definizione della pipeline prima di inviare le modifiche, poiché è possibile che un altro utente o processo abbia modificato la definizione della pipeline dopo l'ultima volta che l'hai utilizzata. Scaricando una copia della definizione corrente e utilizzandola come base per le modifiche, è possibile avere la certezza che si sta utilizzando la definizione di pipeline più recente. È inoltre consigliabile recuperare di nuovo la definizione di pipeline dopo la modifica, in modo da assicurarsi che l'aggiornamento sia riuscito.

Quando si utilizza la CLI, è possibile ottenere due diverse versioni della pipeline. La versione `active` è la pipeline attualmente in esecuzione. La versione `latest` è una copia creata quando si modifica una pipeline in esecuzione. Quando si carica la pipeline modificata, diventa la versione `active` e quella precedente `active` non è più disponibile.

Per ottenere una definizione della pipeline utilizzando il AWS CLI

Per ottenere la definizione completa della pipeline, utilizzare il get-pipeline-definitioncomando. La definizione di pipeline è stampata per l'output standard (stdout).

L'esempio seguente riceve la definizione di pipeline per la pipeline specificata.

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

Per recuperare una versione specifica di una pipeline, utilizzare l'opzione `--version`. L'esempio seguente recupera la versione `active` della pipeline specificata.

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

Visualizzazione dei dettagli dell'istanza della pipeline

È possibile monitorare l'avanzamento della pipeline. Per ulteriori informazioni sullo stato delle istanze, consulta Interpretazione dei dettagli sullo stato della pipeline. Per ulteriori informazioni sulla risoluzione di problemi con istanze della pipeline non eseguite o non completate, consulta Risoluzione dei problemi più comuni.

Per monitorare lo stato di avanzamento di una pipeline utilizzando il AWS CLI

Per recuperare i dettagli dell'istanza della pipeline, ad esempio uno storico delle volte in cui la pipeline è stata eseguita, utilizzare il comando list-runs. Questo comando consente di filtrare l'elenco di esecuzioni restituite in base al loro stato corrente o agli intervalli di data in cui sono state avviate. Il filtraggio dei risultati è utile perché, a seconda dell'età e della pianificazione della pipeline, la cronologia delle esecuzioni può essere di grandi dimensioni.

L'esempio seguente recupera informazioni per tutte le esecuzioni.

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

L'esempio seguente recupera informazioni per tutte le esecuzioni completate.

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

L'esempio seguente recupera informazioni per tutte le esecuzioni avviate nell'intervallo di tempo specificato.

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

Visualizza log pipeline

La registrazione a livello di pipeline è supportata durante la creazione della pipeline specificando una posizione Amazon S3 nella console o con `pipelineLogUri` un oggetto predefinito in SDK/CLI. La struttura della directory per ogni pipeline all'interno di quella URI è la seguente:

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

Per la pipeline, df-00123456ABC7DEF8HIJK, la struttura della directory è simile a:

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

Per ShellCommandActivity, i log per stderr e stdout associati a queste attività sono memorizzati nella directory per ogni tentativo.

Per le risorse, ad esempio, EmrCluster, dove viene impostato un valore emrLogUri, tale valore ha la priorità. Altrimenti, le risorse (compresi i TaskRunner log di tali risorse) seguono la struttura di registrazione della pipeline sopra descritta.

Per visualizzare i log di una determinata pipeline, esegui:

1. Recupera il ObjectId chiamando per query-objects ottenere l'ID esatto dell'oggetto. Esempio:

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

   query-objectsè una CLI impaginata e può restituire un token di impaginazione se ci sono più esecuzioni per quel dato. pipeline-id È possibile utilizzare il token per eseguire tutti i tentativi fino a trovare l'oggetto previsto. Ad esempio, un risultato restituito ObjectId sarebbe simile a:@TableBackupActivity_2023-05-020T18:05:18_Attempt=1.
2. Utilizzando il ObjectId, recupera la posizione del registro utilizzando:

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

Messaggio di errore relativo a un'attività non riuscita

Per visualizzare il messaggio di errore, è necessario innanzitutto ObjectId utilizzarloquery-objects.

Dopo aver recuperato l'errore ObjectId, usa la describe-objects CLI per ottenere il messaggio di errore effettivo.

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

Annulla, riesegui o contrassegna un oggetto come finito

Utilizzate la `set-status` CLI per annullare un oggetto in esecuzione, eseguire nuovamente un oggetto fallito o contrassegnare un oggetto in esecuzione come Finito.

Innanzitutto, ottieni l'ID dell'oggetto utilizzando la `query-objects` CLI. Esempio:

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

Utilizzate la `set-status` CLI per modificare lo stato dell'oggetto desiderato. Esempio:

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

Modifica della pipeline

Per modificare un aspetto di una delle pipeline, è possibile aggiornare la definizione di pipeline. Dopo aver modificato una pipeline in esecuzione, è necessario riattivare la pipeline affinché le modifiche diventino effettive. Inoltre, è possibile eseguire nuovamente uno o più componenti della pipeline.

Indice

Limitazioni

Mentre la pipeline è nello `PENDING` stato e non è attivata, non è possibile apportarvi alcuna modifica. Dopo aver attivato una pipeline, è possibile modificare la pipeline con le seguenti limitazioni. Le modifiche apportate si applicano a nuove esecuzioni degli oggetti della pipeline dopo averli salvati, quindi attivare la pipeline di nuovo.

- Non è possibile eliminare un oggetto

- Non è possibile modificare il periodo di pianificazione di un oggetto esistente
- Non è possibile aggiungere, eliminare o modificare campi di riferimento in un oggetto esistente
- Non è possibile fare riferimento a un oggetto esistente in un campo di output di un nuovo oggetto
- Non è possibile modificare la data di inizio pianificata di un oggetto (invece, attivare la pipeline con data e ora specifiche)

Modifica di una tubazione utilizzando il AWS CLI

È possibile modificare una pipeline tramite gli strumenti a riga di comando.

Innanzitutto, scaricate una copia della definizione corrente della pipeline utilizzando il get-pipeline-definitioncomando. In questo modo, si ha la certezza di modificare la definizione più recente di pipeline. L'esempio seguente stampa la definizione di pipeline in un output standard (stdout).

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

Salvare la definizione di pipeline su un file e modificarla in base alle esigenze. Aggiorna la definizione della pipeline usando il put-pipeline-definitioncomando. L'esempio seguente carica il file di definizione pipeline aggiornato.

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

È possibile recuperare la definizione di pipeline utilizzando il comando `get-pipeline-definition` per assicurarsi che l'aggiornamento sia stato eseguito correttamente. Per attivare la pipeline, utilizzare il seguente comando activate-pipeline:

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

Se si preferisce, attivare la pipeline a partire da una determinata data e ora, utilizzando l'opzione `--start-timestamp` come segue:

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

Per eseguire nuovamente uno o più componenti della pipeline, utilizzare il comando set-status.

Clonazione della pipeline

La clonazione crea una copia di una pipeline e consente di specificare un nome per la nuova pipeline. È possibile clonare una pipeline che si trova in qualsiasi stato, anche se presenta errori; tuttavia, la nuova pipeline rimane nello stato di `PENDING` finché non verrà attivata manualmente. Per la nuova pipeline, l'operazione di clonazione utilizza la versione più recente della definizione originale di pipeline anziché la versione attiva. Nell'operazione di clonazione, la pianificazione completa della pipeline originale non viene copiata nella nuova pipeline, solo l'impostazione del periodo.

Per clonare una pipeline utilizzando la CLI: AWS

1. Crea una nuova pipeline con un nuovo nome e un ID univoco. Annota l'ID della pipeline restituito.
2. Utilizza la `get-pipeline-definition` CLI per ottenere la definizione della pipeline esistente da clonare e scriverla in un file temporaneo. Nota il percorso assoluto del file.
3. Utilizzate la `put-pipeline-definition` CLI per copiare la definizione della pipeline dalla pipeline esistente alla nuova pipeline.
4. Utilizza la `get-pipeline-definition` CLI per ottenere la definizione della nuova pipeline per verificare la definizione della pipeline.

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

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

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

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

Assegnazione di tag alla pipeline

I tag sono coppie chiave-valore che fanno distinzione tra minuscole e maiuscole e contengono una chiave e un valore facoltativo, entrambi definiti dall'utente. È possibile applicare fino a dieci tag a ogni pipeline. Le chiavi del tag devono essere univoche per ciascuna pipeline. Se aggiungi un tag con una chiave già associata alla pipeline, il valore del tag viene aggiornato.

L'applicazione di un tag a una pipeline propaga anche i tag alle relative risorse sottostanti (ad esempio, cluster Amazon EMR e istanze Amazon EC2). Tuttavia, non applica questi tag alle risorse in uno stato `FINISHED` o comunque terminato. È possibile utilizzare l'interfaccia a riga di comando per applicare tag a queste risorse, se necessario.

Quando il tag non è più necessario, è possibile eliminarlo dalla pipeline.

Applicare tag alla pipeline utilizzando AWS CLI

Per aggiungere i tag a una nuova pipeline, aggiungere l'opzione `--tags` al comando create-pipeline. Ad esempio, l'opzione seguente crea una pipeline con due tag, un tag `environment` con un valore di `production` e un tag `owner` con un valore di `sales`.

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

Per aggiungere i tag a una pipeline esistente, utilizzare il comando add-tags come segue:

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

Per eliminare i tag da una pipeline esistente, utilizzare il comando remove-tags come segue:

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

Disattivazione pipeline

La disattivazione di una pipeline in esecuzione sospende la pipeline. Per riprendere l'esecuzione della pipeline, è possibile attivare la pipeline. In questo modo è possibile apportare modifiche. Ad esempio, se si scrivono dati su un database per cui è prevista la manutenzione, è possibile disattivare la pipeline, attenderne il completamento, quindi attivare la pipeline.

Quando si disattiva una pipeline, è possibile specificare il risultato per l'esecuzione di attività. Per impostazione predefinita, queste attività vengono annullate immediatamente. In alternativa, è possibile fare in modo che AWS Data Pipeline attenda il completamento delle attività prima di disattivare la pipeline.

Quando si attiva una pipeline disattivata, è possibile specificare quando riprende. Utilizzando l'API AWS CLI o l'API, per impostazione predefinita la pipeline riprende dall'ultima esecuzione completata oppure puoi specificare la data e l'ora in cui riprendere la pipeline.

Disattiva la tua pipeline utilizzando il AWS CLI

Per disattivare la pipeline, utilizza il seguente comando deactivate-pipeline:

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

Per disattivare la pipeline solo dopo che tutte le attività in esecuzione siano terminate, aggiungere l'opzione `--no-cancel-active` come segue:

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

Quando si è pronti, è possibile riprendere l'esecuzione della pipeline nel punto in cui era stata interrotta utilizzando il seguente comando attivare-pipeline:

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

Per avviare la pipeline a partire da una data e ora specifiche, aggiungere l'opzione `--start-timestamp`, come segue:

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

Eliminazione della pipeline

Quando non è più necessario avere una pipeline, ad esempio una pipeline create durante il testing delle applicazioni, è necessario eliminarla per rimuoverla dall'uso attivo. L'eliminazione di una pipeline la inserisce in uno stato di eliminazione. Quando la pipeline è nello stato di eliminazione, la definizione di pipeline e la cronologia delle esecuzioni vengono rimosse. Pertanto, non è più possibile eseguire operazioni sulla pipeline, inclusa la descrizione.

> **Important**
>
> Non è possibile ripristinare una pipeline eliminata, per cui assicurarsi di non aver bisogno della pipeline in futuro prima di eliminarla.

Per eliminare una pipeline utilizzando il AWS CLI

Per eliminare la pipeline, utilizzare il comando delete-pipeline. Il comando seguente elimina la pipeline specificata.

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

Dati e tabelle in gestione temporanea con attività della pipeline

AWS Data Pipeline può inserire i dati di input e output nelle pipeline per semplificare l'utilizzo di determinate attività, come `ShellCommandActivity` e. `HiveActivity`

La gestione temporanea dei dati consente di copiare i dati dal nodo di dati di input alla risorsa che esegue l'attività e, in modo analogo, dalla risorsa al nodo di dati di output.

I dati staged sulla risorsa Amazon EMR o Amazon EC2 sono disponibili utilizzando variabili speciali nei comandi della shell dell'attività o negli script Hive.

La gestione temporanea delle tabelle è simile a quella dei dati, tranne che i dati in gestione temporanea assumono la forma di tabelle di database, in modo specifico.

AWS Data Pipeline supporta i seguenti scenari di staging:

- Gestione temporanea dei dati con `ShellCommandActivity`
- La gestione temporanea della tabella con Hive e i nodi di dati supportati da tale gestione
- La gestione temporanea della tabella con Hive e i nodi di dati non supportati da tale gestione

> **Note**
>
> La gestione temporanea funziona solo quando il campo `stage` è impostato su `true` in un'attività, ad esempio `ShellCommandActivity`. Per ulteriori informazioni, consulta ShellCommandActivity.

Inoltre, i nodi di dati e le attività possono essere correlati in quattro modi:

Gestione temporanea dei dati in locale su una risorsa

I dati di input vengono automaticamente copiati nel file system locale della risorsa. I dati di output vengono automaticamente copiati dal file system locale della risorsa al nodo di dati di output. Ad esempio, quando si configurano gli input e gli output `ShellCommandActivity` con gestione temporanea = true, i dati di input sono disponibili come INPUTx_STAGING_DIR e i dati di output sono disponibili come OUTPUTx_STAGING_DIR, dove x è il numero di input o output.

Gestione temporanea delle definizioni di input e output per un'attività

Il formato di dati di input (nomi delle colonne e nomi delle tabelle) vengono automaticamente copiati nella risorsa dell'attività. Ad esempio, quando si configura `HiveActivity` con gestione temporanea = true. Il formato di dati specificato nell'input `S3DataNode` viene utilizzato per la definizione della tabella dalla tabella Hive.

Gestione temporanea non abilitata

Gli oggetti di input e output e i loro campi sono disponibili per l'attività, mentre i dati no. Ad esempio, `EmrActivity` per impostazione predefinita o quando si configurano altre attività con gestione temporanea = false. In questa configurazione, i campi di dati sono disponibili affinché l'attività possa farvi riferimento utilizzando la sintassi dell' AWS Data Pipeline espressione e ciò si verifica solo quando la dipendenza è soddisfatta. Questo serve solo come controllo delle dipendenze. Il codice nell'attività è responsabile della copia dei dati dall'input alla risorsa che esegue l'attività.

Relazione di dipendenza tra gli oggetti

Si è verificato un dipende-dalla relazione tra due oggetti, che comporta una situazione analoga a quando la gestione temporanea non è abilitata. In questo modo, i dati di un nodo o un'attività agiscono come precondizione per l'esecuzione di un'altra attività.

Gestione temporanea dei dati con ShellCommandActivity

Consideriamo uno scenario che utilizza `S3DataNode` oggetti `ShellCommandActivity` con come input e output dei dati. AWS Data Pipeline organizza automaticamente i nodi di dati per renderli accessibili al comando shell come se fossero cartelle di file locali utilizzando le variabili di ambiente `${INPUT1_STAGING_DIR}` e `${OUTPUT1_STAGING_DIR}` come mostrato nell'esempio seguente. La porzione numerica delle variabili denominate `INPUT1_STAGING_DIR` e `OUTPUT1_STAGING_DIR` cresce a seconda del numero di nodi di dati a cui fa riferimento l'attività.

> **Note**
>
> Questo scenario funziona solo se, come descritto, i dati di input e output sono oggetti `S3DataNode`. Inoltre, la gestione temporanea dei dati di output è consentita solo quando `directoryPath` è impostato sull'oggetto di output `S3DataNode`.

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

La gestione temporanea della tabella con Hive e i nodi di dati supportati da tale gestione

Consideriamo uno scenario che utilizza `S3DataNode` oggetti `HiveActivity` con come input e output dei dati. AWS Data Pipeline organizza automaticamente i nodi di dati per renderli accessibili allo script Hive come se fossero tabelle Hive utilizzando le variabili `${input1}` e `${output1}` come mostrato nell'esempio seguente per. `HiveActivity` La porzione numerica delle variabili denominate `input` e `output` cresce a seconda del numero di nodi di dati a cui fa riferimento l'attività.

> **Note**
>
> Questo scenario funziona solo se, come descritto, i dati di input e output sono oggetti `S3DataNode` o `MySqlDataNode`. La gestione temporanea delle tabelle non è supportata per `DynamoDBDataNode`.

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

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

La gestione temporanea della tabella con Hive e i nodi di dati non supportati da tale gestione

Considerare uno scenario utilizzando una `HiveActivity` con `DynamoDBDataNode` come input dei dati e un oggetto `S3DataNode` come output. Non è disponibile alcun data staging`DynamoDBDataNode`, quindi è necessario prima creare manualmente la tabella all'interno dello script hive, utilizzando il nome della variabile `#{input.tableName}` per fare riferimento alla tabella DynamoDB. Una nomenclatura simile si applica se la tabella DynamoDB è l'output, a meno che non si utilizzi una variabile. `#{output.tableName}` La gestione temporanea è disponibile per l'output dell'oggetto `S3DataNode` in questo esempio, pertanto è possibile fare riferimento al nodo dei dati di output come a `${output1}`.

> **Note**
>
> In questo esempio, la variabile del nome della tabella ha il prefisso del carattere # (hash) perché AWS Data Pipeline utilizza espressioni per accedere a or. `tableName` `directoryPath` Per ulteriori informazioni su come funziona la valutazione delle espressioni AWS Data Pipeline, vedere. Valutazione delle espressioni

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

Utilizzo di una pipeline con risorse in più regioni

Per impostazione predefinita, le `EmrCluster` risorse `Ec2Resource` e vengono eseguite nella stessa area AWS Data Pipeline, tuttavia AWS Data Pipeline supporta la capacità di orchestrare i flussi di dati tra più aree, ad esempio l'esecuzione di risorse in un'area che consolida i dati di input da un'altra regione. Consentendo alle risorse di essere eseguite in una determinata regione, si ha anche la

flessibilità necessaria per individuare le risorse con i propri dataset dipendenti e per massimizzare le prestazioni riducendo latenze ed evitando costi di trasferimento dei dati in più regioni. È possibile configurare le risorse in modo che vengano eseguite in un'area diversa rispetto all' AWS Data Pipeline utilizzo del `region` campo su and. `Ec2Resource` `EmrCluster`

Il seguente file JSON della pipeline di esempio mostra come eseguire una `EmrCluster` risorsa nella regione Europa (Irlanda), supponendo che nella stessa regione esista una grande quantità di dati su cui lavorare il cluster. In questo esempio, l'unica differenza rispetto a una tipica pipeline è che `EmrCluster` ha un valore campo `region` impostato su `eu-west-1`.

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

```
}
```

La tabella seguente elenca le regioni che è possibile scegliere e i codici di regione associati da utilizzare nel campo `region`.

> **Note**
>
> L'elenco seguente include le regioni in cui è AWS Data Pipeline possibile orchestrare i flussi di lavoro e avviare risorse Amazon EMR o Amazon EC2. AWS Data Pipeline potrebbe non essere supportato in queste regioni. Per informazioni sulle regioni in cui AWS Data Pipeline è supportato, consulta AWS Regions and Endpoints.

Nome della regione	Codice regione
Stati Uniti orientali (Virginia settentrionale)	us-east-1
Stati Uniti orientali (Ohio)	us-east-2
Stati Uniti occidentali (California settentrionale)	us-west-1
US West (Oregon)	us-west-2
Canada (Centrale)	ca-central-1
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2
Europa (Francoforte)	eu-central-1
Asia Pacifico (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Mumbai)	ap-south-1
Asia Pacifico (Tokyo)	ap-northeast-1
Asia Pacifico (Seoul)	ap-northeast-2

Nome della regione	Codice regione
Sud America (San Paolo)	sa-east-1

Guasti di una delle dipendenze e riesecuzioni

AWS Data Pipeline consente di configurare il comportamento degli oggetti della pipeline quando una dipendenza fallisce o viene annullata da un utente. È possibile accertarsi che i guasti cadano su altri oggetti della pipeline (consumatori), per evitare attese indefinite. Tutte le attività, i nodi di dati e le precondizioni dispongono di un campo denominato `failureAndRerunMode` con un valore di default di none. Per abilitare i guasti di una delle dipendenze, impostare il campo `failureAndRerunMode` su `cascade`.

Quando questo campo è abilitato, i guasti di una delle dipendenze si verificano se l'oggetto di una pipeline è bloccato nello stato `WAITING_ON_DEPENDENCIES` ed eventuali dipendenze hanno dato esito negativo senza alcun comando in sospeso. Durante un guasto di una delle dipendenze, si verificano i seguenti eventi:

- Quando un oggetto ha esito negativo, i suoi consumatori vengono impostati su `CASCADE_FAILED` e sia l'oggetto originale sia le precondizioni dei consumatori vengono impostate su `CANCELED`.
- Tutti gli oggetti già presenti `FINISHED`, `FAILED` oppure `CANCELED` vengono ignorati.

Il guasto di una delle sue dipendenze non funziona sulle dipendenze non riuscite di un oggetto (upstream), tranne per le precondizioni associate all'oggetto originale non riuscito. Gli oggetti di una pipeline interessati da un guasto di una delle dipendenze possono attivare nuovi tentativi o post-operazioni, come, ad esempio `onFail`.

Gli effetti dettagliati di un guasto di una delle dipendenze dipendono dal tipo di oggetto.

Attività

Un'attività cambia in `CASCADE_FAILED` se una delle dipendenze fallisce e, successivamente, innesca un guasto di una delle dipendenze nei consumatori dell'attività. Se una risorsa da cui dipende un'attività ha esito negativo, l'attività è `CANCELED` e tutti i suoi consumatori cambiano in `CASCADE_FAILED`.

Nodi di dati e condizioni preliminari

Se un nodo dati viene configurato come l'output di un'attività che ha esito negativo, il nodo di dati cambia nello stato `CASCADE_FAILED`. Il guasto di un nodo di dati si propaga a qualsiasi precondizione associata che cambia nello stato `CANCELED`.

Resources

Se gli oggetti che dipendono da una risorsa sono nello stato `FAILED` e la risorsa stessa è nello stato `WAITING_ON_DEPENDENCIES`, allora la risorsa passa allo stato `FINISHED`.

Riesecuzione di oggetti con errori a cascata

Per impostazione predefinita, rieseguendo qualsiasi attività o nodo di dati si esegue di nuovo solo la risorsa associata. Tuttavia, impostare il campo `failureAndRerunMode` su `cascade` in un oggetto pipeline consente una nuova esecuzione di un comando su un oggetto di destinazione da propagare a tutti i consumatori, nelle seguenti condizioni:

- I consumatori dell'oggetto di destinazione sono nello stato `CASCADE_FAILED`.
- Le dipendenze dell'oggetto di destinazione non hanno comandi per la riesecuzione in sospeso.
- Le dipendenze dell'oggetto di destinazione non sono nello stato `FAILED`, `CASCADE_FAILED` o `CANCELED`.

Se tenti di eseguire di nuovo un oggetto `CASCADE_FAILED` e una qualsiasi delle sue dipendenze è `FAILED`, `CASCADE_FAILED` oppure `CANCELED`, la nuova esecuzione fallirà e restituirà l'oggetto allo stato `CASCADE_FAILED`. Per rieseguire l'oggetto fallito senza errori, è necessario rintracciare l'errore fino alla catena di dipendenza per individuare l'origine dell'errore ed eseguire di nuovo l'oggetto. Quando viene inviato il comando di riesecuzione su una risorsa, si tenta anche di eseguire di nuovo tutti gli oggetti che dipendono da tale risorsa.

Errori in cascata e backfill

Se si abilita l'errore a cascata e si dispone di una pipeline che crea molti backfill, gli errori di runtime della pipeline possono causare la creazione e l'eliminazione di risorse in rapida successione senza eseguire operazioni utili. AWS Data Pipeline tenta di avvisarti di questa situazione con il seguente messaggio di avviso quando salvi una pipeline: *`Pipeline_object_name`* `has 'failureAndRerunMode' field set to 'cascade' and you are about to create`

`a backfill with scheduleStartTime start_time. This can result in rapid creation of pipeline objects in case of failures.` Ciò accade perché un errore a cascata può impostare rapidamente le attività a valle e chiudere i cluster EMR `CASCADE_FAILED` e le risorse EC2 che non sono più necessarie. È consigliabile testare le pipeline con brevi intervalli di tempo per limitare l'impatto di questa situazione.

Sintassi del file di definizione della pipeline

Le istruzioni in questa sezione riguardano l'utilizzo manuale dei file di definizione della pipeline utilizzando l'interfaccia a riga di AWS Data Pipeline comando (CLI). Si tratta di un'alternativa alla progettazione interattiva di una pipeline tramite la console. AWS Data Pipeline

È possibile creare manualmente i file di definizione della pipeline utilizzando qualsiasi editor di testo che supporti il salvataggio dei file utilizzando il formato di file UTF-8 e inviare i file utilizzando l'interfaccia a riga di comando. AWS Data Pipeline

AWS Data Pipeline supporta anche una varietà di espressioni e funzioni complesse all'interno delle definizioni delle pipeline. Per ulteriori informazioni, consulta Funzioni ed espressioni della pipeline.

Struttura dei file

Il primo passo nella creazione della pipeline consiste nel comporre oggetti di definizione della pipeline in un file di definizione della pipeline. L'esempio seguente illustra la struttura generale di un file di definizione della pipeline. Questo file definisce due oggetti, che sono delimitati da "{ and }" e separati da una virgola.

Nell'esempio seguente il primo oggetto definisce due coppie nome-valore, note come campi. Il secondo oggetto definisce tre campi.

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

```
  ]
}
```

Quando si crea un file di definizione della pipeline, è necessario selezionare i tipi di oggetti della pipeline di cui hai bisogno, aggiungerli al file di definizione della pipeline, quindi aggiungere i campi appropriati. Per ulteriori informazioni sugli oggetti della pipeline, consulta Riferimento all'oggetto pipeline.

Ad esempio, è possibile creare un oggetto di definizione della pipeline per un nodo di dati di input e un altro per il nodo di dati di output. Quindi crea un altro oggetto di definizione della pipeline per un'attività, ad esempio l'elaborazione dei dati di input utilizzando Amazon EMR.

Campi della pipeline

Dopo aver capito quali tipi di oggetti includere nel file di definizione della pipeline, aggiungere campi alla definizione di ogni oggetto della pipeline. I nomi dei campi vengono inclusi tra virgolette e separati dai valori di campo da uno spazio, una virgola e uno spazio, come mostrato nel seguente esempio.

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

Il valore del campo può essere una stringa di testo, un riferimento a un altro oggetto, una chiamata di funzione, un'espressione o un elenco ordinato di uno qualsiasi dei tipi precedenti. Per ulteriori informazioni sui tipi di dati che possono essere utilizzati per i valori dei campi, vedi Tipi di dati di esempio . Per ulteriori informazioni sulle funzioni da utilizzare per valutare i valori dei campi, consulta Valutazione delle espressioni.

I campi sono limitati a 2048 caratteri. Gli oggetti possono avere una dimensione pari a 20 KB, il che significa che non è possibile aggiungere molti campi di grandi dimensioni a un oggetto.

Ogni oggetto della pipeline deve contenere i campi riportati di seguito: `id` e `type`, come mostrato nel seguente esempio. Altri campi potrebbero essere richiesti in base al tipo di oggetto. Selezionare un valore `id` significativo per l'utente e univoco all'interno della definizione di pipeline. Il valore per `type` specifica il tipo dell'oggetto. Specificare uno dei tipi di oggetto di definizione della pipeline supportati tra quelli elencati nell'argomento Riferimento all'oggetto pipeline.

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

Per ulteriori informazioni sui campi obbligatori e facoltativi per ogni oggetto, consulta la documentazione per l'oggetto.

Per includere campi da un oggetto in un altro oggetto, utilizzare il campo `parent` con un riferimento all'oggetto. Ad esempio, l'oggetto "B" include i propri campi, "B1" e "B2", più i campi dell'oggetto "A", "A1" e "A2".

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

È possibile definire campi comuni in un oggetto con l'ID "Default". Questi campi vengono automaticamente inclusi in ogni oggetto nel file di definizione della pipeline che non imposta esplicitamente il campo `parent` di riferimento a un altro oggetto.

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

Campi definiti dall'utente

È possibile creare campi personalizzati o definiti dall'utente nei componenti della pipeline e fare riferimento a essi con le espressioni. L'esempio seguente mostra un campo personalizzato denominato `myCustomField` e `my_customFieldReference` aggiunto a un oggetto S3: DataNode

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

Un campo definito dall'utente deve avere un nome con prefisso con la parola "my" in tutte lettere minuscole, seguito da una lettera maiuscola o con il carattere di sottolineatura. Inoltre, un campo definito dall'utente può essere un valore di stringa, come l'esempio precedente `myCustomField` o un riferimento a un altro componente della pipeline, come l'esempio precedente `my_customFieldReference`.

> **Note**
>
> Nei campi definiti dall'utente, verifica AWS Data Pipeline solo i riferimenti validi ad altri componenti della pipeline, non i valori di stringa di campo personalizzati aggiunti.

Lavorare con l'API

> **Note**
>
> Se non stai scrivendo programmi che interagiscono con AWS Data Pipeline, non devi installare nessuno degli AWS SDKs. Puoi creare ed eseguire pipeline tramite la console o l'interfaccia a riga di comando. Per ulteriori informazioni, consulta Configurazione per AWS Data Pipeline

Il modo più semplice per scrivere applicazioni che interagiscono AWS Data Pipeline o per implementare un Task Runner personalizzato consiste nell'utilizzare uno degli AWS SDKs. AWS SDKs fornisce funzionalità che semplificano la chiamata al servizio Web APIs dall'ambiente di programmazione preferito. Per ulteriori informazioni, consulta Installazione del kit SDK AWS.

Installazione del kit SDK AWS

AWS SDKs Forniscono funzioni che racchiudono l'API e si occupano di molti dettagli della connessione, come il calcolo delle firme, la gestione dei nuovi tentativi di richiesta e la gestione degli errori. SDKs Inoltre contengono codice di esempio, tutorial e altre risorse per aiutarti a iniziare a scrivere applicazioni che chiamano. AWS La chiamata alle funzioni wrapper in un SDK può

semplificare notevolmente il processo di scrittura di un'applicazione. AWS Per ulteriori informazioni su come scaricare e utilizzare AWS SDKs, consulta Sample Code & Libraries.

AWS Data Pipeline il supporto è disponibile SDKs per le seguenti piattaforme:

- SDK AWS per Java
- AWS SDK per Node.js
- SDK AWS per PHP
- SDK AWS per Python (Boto)
- SDK AWS per Ruby
- SDK AWS per .NET

Effettuare una richiesta HTTP a AWS Data Pipeline

Per una descrizione completa degli oggetti programmatici di AWS Data Pipeline, consultate l'AWS Data Pipeline API Reference.

Se non utilizzi uno degli AWS SDKs, puoi eseguire AWS Data Pipeline operazioni su HTTP utilizzando il metodo di richiesta POST. Per utilizzare il metodo POST devi specificare l'operazione nell'intestazione della richiesta e fornire i dati per l'operazione in formato JSON nel corpo della richiesta.

Contenuti nell'intestazione HTTP

AWS Data Pipeline richiede le seguenti informazioni nell'intestazione di una richiesta HTTP:

- `host`L' AWS Data Pipeline endpoint.

  Per informazioni sugli endpoint consulta Regioni ed endpoint.
- `x-amz-date`È necessario fornire il timestamp nell'intestazione HTTP Date o nell'intestazione x-amz-date AWS. (Alcune librerie client HTTP non consentono di impostare l'intestazione Date). Quando è presente un' x-amz-dateintestazione, il sistema ignora qualsiasi intestazione Date durante l'autenticazione della richiesta.

  La data deve essere specificata in uno dei seguenti tre formati, come indicato nel protocollo HTTP/1.1 RFC:
  - Dom, 06 novembre 1994 08:49:37 GMT (RFC 822, aggiornato da RFC 1123)
  - Domenica, 06-Nov-94 08:49:37 GMT (RFC 850, reso obsoleto da RFC 1036)

  - Dom Nov 6 08:49:37 1994 (Formato asctime() ANSI C)
- `Authorization` Il set di parametri di autorizzazione usato da AWS per garantire la validità e l'autenticità della richiesta. Per ulteriori informazioni su come costruire l'intestazione, consulta Processo di firma Signature Versione 4.
- `x-amz-target` Il servizio di destinazione della richiesta e l'operazione per i dati, nel formato: `<<serviceName>>_<<API version>>.<<operationName>>`

  Ad esempio, `DataPipeline_20121129.ActivatePipeline`
- `content-type` Specifica JSON e la versione. Ad esempio, `Content-Type: application/x-amz-json-1.0`

Il seguente è un esempio di intestazione per una richiesta HTTP per l'attivazione di una 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
```

Contenuto del corpo HTTP

Il corpo di una richiesta HTTP contiene i dati per l'operazione specificata nell'intestazione di una richiesta HTTP. I dati devono essere formattati secondo lo schema di dati JSON per ogni API. AWS Data Pipeline Lo schema di dati AWS Data Pipeline JSON definisce i tipi di dati e parametri (come operatori di confronto e costanti di enumerazione) disponibili per ogni operazione.

Formattare il corpo di una richiesta HTTP

Utilizza il formato dati JSON per trasmettere simultaneamente i valori dei dati e la struttura corrispondente. Gli elementi possono essere annidati all'interno di altri elementi utilizzando la notazione parentesi. L'esempio seguente mostra una richiesta per mettere una definizione di pipeline composta da tre oggetti e relativi slot.

```
{
```

```
 "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"}

      ]
    }
  ]
}
```

Gestire la risposta HTTP

Si elencano di seguito alcune intestazioni importanti nella risposta HTTP e il modo in cui vanno gestite nell'applicazione:

- HTTP/1.1: questa intestazione è seguita da un codice di stato. Un valore del codice di 200 indica un'operazione riuscita. Qualsiasi altro valore indica un errore.
- x-amzn- RequestId —Questa intestazione contiene un ID di richiesta che puoi utilizzare se devi risolvere una richiesta con. AWS Data Pipeline Un esempio di ID di richiesta è K2 07N97 Q9ASUAAJG. QH8 DNOU9 FNA2 GDLL8 OBVV4 KQNSO5 AEMVJF66
- x-amz-crc32 —AWS Data Pipeline calcola un CRC32 checksum del payload HTTP e lo restituisce nell'intestazione 32. x-amz-crc Ti consigliamo di calcolare il tuo CRC32 checksum sul lato client e di confrontarlo con l'intestazione x-amz-crc 32; se i checksum non corrispondono, potrebbe indicare che i dati sono stati danneggiati durante il transito. In questo caso, è necessario effettuare di nuovo la richiesta.

Gli utenti dell'SDK AWS non devono eseguire manualmente questa verifica, in quanto SDKs calcolano il checksum di ogni risposta da Amazon DynamoDB e riprovano automaticamente se viene rilevata una mancata corrispondenza.

Esempio di richiesta e risposta JSON AWS Data Pipeline

I seguenti esempi mostrano una richiesta per la creazione di una nuova pipeline. Quindi mostra la AWS Data Pipeline risposta, incluso l'identificatore della pipeline appena creata.

Richiesta HTTP POST

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

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

AWS Data Pipeline Risposta

```
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"}
```

Sicurezza in AWS Data Pipeline

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS Data Pipeline, consulta AWS Services in Scope by Compliance Program .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo AWS Data Pipeline. Negli argomenti seguenti viene illustrato come eseguire la configurazione AWS Data Pipeline per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a usare altri servizi AWS che ti aiutano a monitorare e proteggere AWS Data Pipeline le tue risorse.

Argomenti

Protezione dei dati in AWS Data Pipeline

Il modello di responsabilità AWS condivisa modello di di si applica alla protezione dei dati in AWS Data Pipeline. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. Questi contenuti comprendono la configurazione della protezione e le attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al AWS Modello di responsabilità condivisa e GDPR nel AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È consigliabile TLS 1.2 o versioni successive.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.
- AWS Data Pipeline supporta IMDSv2 le risorse Amazon EMR e Amazon EC2 . Per l'uso IMDSv2 con Amazon EMR, utilizza le versioni 5.23.1, 5.27.1 o 5.32 o successive o la versione 6.2 o successiva. Per ulteriori informazioni, consulta Configurazione delle richieste di servizi di metadati per le EC2 istanze Amazon e Utilizzo. IMDSv2

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori AWS Data Pipeline o Servizi AWS utilizzi la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere

utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Identity and Access Management per AWS Data Pipeline

Le credenziali di sicurezza identificano l'utente per i servizi su AWS e concedono le autorizzazione per utilizzare le risorse AWS, ad esempio le pipeline. Puoi utilizzare le funzionalità di AWS Data Pipeline and AWS Identity and Access Management (IAM) per consentire AWS Data Pipeline e ad altri utenti di accedere alle tue AWS Data Pipeline risorse senza condividere le tue credenziali di sicurezza.

Le organizzazioni possono condividere l'accesso alle pipeline, in modo che i singoli utenti in quella organizzazione siano in grado di svilupparle e mantenerle in modo collaborativo. Tuttavia, per esempio, può essere necessario eseguire le operazioni seguenti:

- Controlla quali utenti possono accedere a pipeline specifiche
- Proteggere una pipeline di produzione per evitare che venga modificata per errore
- Consentire a entità di controllo l'accesso in sola lettura alle pipeline, ma impedire loro di apportare modifiche

AWS Data Pipeline è integrato con AWS Identity and Access Management (IAM), che offre un'ampia gamma di funzionalità:

- Crea utenti e gruppi nel tuo Account AWS.
- Condividi facilmente AWS le tue risorse tra gli utenti del tuo Account AWS.
- Assegna credenziali di sicurezza uniche a ciascun utente.
- Controlla l'accesso di ogni utente a servizi e risorse.
- Ottieni una fattura unica per tutti gli utenti del tuo Account AWS.

Utilizzando IAM with AWS Data Pipeline, puoi controllare se gli utenti della tua organizzazione possono eseguire un'attività utilizzando azioni API specifiche e se possono utilizzare risorse AWS specifiche. Puoi utilizzare le policy IAM basate su tag di pipeline e gruppi di lavoro per condividere le tue pipeline con altri utenti e controllare il livello di accesso di cui dispongono.

Indice

Politiche IAM per AWS Data Pipeline

Per impostazione predefinita, le entità IAM non dispongono dell'autorizzazione per creare o modificare risorse AWS. Per consentire alle entità IAM di creare o modificare risorse ed eseguire attività, è necessario creare policy IAM che concedano alle entità IAM l'autorizzazione a utilizzare le risorse e le azioni API specifiche di cui avranno bisogno, e quindi collegare tali policy alle entità IAM che richiedono tali autorizzazioni.

Quando si collega una policy a un utente o a un gruppo di utenti, viene concessa o rifiutata agli utenti l'autorizzazione per l'esecuzione delle attività specificate sulle risorse specificate. Per informazioni generali sulle policy IAM, consulta Permissions and Policies nella guida per l'utente IAM. Per ulteriori informazioni sulla gestione e la creazione di policy IAM personalizzate, consulta la sezione relativa alla gestione delle policy IAM.

Indice

Sintassi delle policy

Una policy IAM è un documento JSON costituito da una o più dichiarazioni. Ogni dichiarazione è strutturata come segue:

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

```
      }
    }
  ]
}
```

Una dichiarazione di policy include i seguenti elementi:

- Effetto: l'elemento effect può essere `Allow` o `Deny`. Per impostazione predefinita, le entità IAM non sono autorizzate a utilizzare risorse e azioni API, quindi tutte le richieste vengono rifiutate. Un permesso esplicito sostituisce l'impostazione predefinita. Un rifiuto esplicito sovrascrive tutti i consensi.
- Action (Operazione): l'elemento action corrisponde all'operazione API specifica per la quale si concede o si nega l'autorizzazione. Per un elenco di azioni per AWS Data Pipeline, consulta Actions in the AWS Data Pipeline API Reference.
- Resource (Risorsa): la risorsa che viene modificata dall'operazione. L'unico valore valido qui è `"*"`.
- Condition: le condizioni sono facoltative. Possono essere utilizzate per controllare quando sarà in vigore una policy.

  AWS Data Pipeline implementa le chiavi di contesto a livello di AWS (vedi Available Keys for Conditions), oltre alle seguenti chiavi specifiche del servizio.

  - `datapipeline:PipelineCreator`— Per concedere l'accesso all'utente che ha creato la pipeline. Per un esempio, vedi Concedere al proprietario della pipeline l'accesso completo.
  - `datapipeline:Tag`— Per concedere l'accesso in base all'etichettatura della pipeline. Per ulteriori informazioni, consulta Controllo degli accessi alle pipeline tramite i tag.
  - `datapipeline:workerGroup`— Concedere l'accesso in base al nome del gruppo di lavoro. Per ulteriori informazioni, consulta Controllo degli accessi alle pipeline tramite i gruppi di lavoratori.

Controllo degli accessi alle pipeline tramite i tag

Puoi creare policy IAM che fanno riferimento ai tag della tua pipeline. In questo modo è possibile utilizzare l'applicazione di tag alle pipeline per eseguire quanto segue:

- Concedere l'accesso in sola lettura a una pipeline
- Concedi read/write l'accesso a una pipeline
- Bloccare l'accesso a una pipeline

Ad esempio, supponiamo che un responsabile disponga di due ambienti di pipeline, produzione e sviluppo, e di un gruppo IAM per ogni ambiente. Per le pipeline nell'ambiente di produzione, il manager concede l' read/write accesso agli utenti del gruppo IAM di produzione, ma concede l'accesso in sola lettura agli utenti del gruppo IAM di sviluppo. Per le pipeline nell'ambiente di sviluppo, il manager concede l' read/write accesso sia al gruppo IAM di produzione che a quello di sviluppo.

Per realizzare questo scenario, il manager etichetta le pipeline di produzione con il tag «environment=production» e allega la seguente policy al gruppo IAM di sviluppatori. La prima istruzione concede l'accesso in sola lettura a tutte le pipeline. La seconda istruzione concede read/write l'accesso alle pipeline che non hanno un tag «environment=production».

JSON

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

Inoltre, il manager applica la seguente politica al gruppo IAM di produzione. Questa istruzione concede l'accesso completo a tutte le pipeline.

JSON

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

Per ulteriori esempi, vedi Concessione agli utenti dell'accesso in sola lettura basato su un tag e Concessione agli utenti dell'accesso completo basato su un tag.

Controllo degli accessi alle pipeline tramite i gruppi di lavoratori

È possibile creare policy IAM che facciano riferimento ai nomi dei gruppi di lavoro.

Ad esempio, supponiamo che un responsabile disponga di due ambienti di pipeline, produzione e sviluppo, e di un gruppo IAM per ogni ambiente. Il responsabile ha tre server di database con runner di attività configurati, rispettivamente, per gli ambienti di produzione, pre-produzione e sviluppo. Il manager desidera assicurarsi che gli utenti del gruppo IAM di produzione possano creare pipeline che trasferiscano le attività alle risorse di produzione e che gli utenti del gruppo IAM di sviluppo possano creare pipeline che trasferiscano le attività sia alle risorse di pre-produzione che a quelle degli sviluppatori.

Per ottenere questo scenario, il responsabile installa runner di attività sulle risorse di produzione con le credenziali relative e imposta `workerGroup` su "prodresource". Inoltre, il responsabile installa il runner delle attività sulle risorse di sviluppo con le credenziali di sviluppo e imposta `workerGroup` su "pre-produzione" e "sviluppo". Il manager applica la seguente policy al gruppo IAM di sviluppatori per bloccare l'accesso alle risorse «prodresource». La prima istruzione concede l'accesso in sola lettura a tutte le pipeline. La seconda istruzione concede read/write l'accesso alle pipeline quando il nome del gruppo di lavoro ha il prefisso «dev» o «pre-prod».

Inoltre, il manager applica la seguente policy al gruppo IAM di produzione per concedere l'accesso alle risorse «prodresource». La prima istruzione concede l'accesso in sola lettura a tutte le pipeline.

La seconda istruzione concede read/write l'accesso quando il nome del gruppo di lavoro ha il prefisso «prod».

Politiche di esempio per AWS Data Pipeline

I seguenti esempi illustrano come concedere agli utenti accesso completo o limitato alle pipeline.

Indice

Esempio 1: Concedere agli utenti accesso in sola lettura in base a un tag

La seguente policy consente agli utenti di utilizzare le azioni AWS Data Pipeline API di sola lettura, ma solo con le pipeline che hanno il tag «environment=production».

L'azione ListPipelines API non supporta l'autorizzazione basata su tag.

JSON

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

```
        }
      }
    }
  ]
}
```

Esempio 2: Concedere agli utenti l'accesso completo in base a un tag

La seguente politica consente agli utenti di utilizzare tutte le azioni AWS Data Pipeline API, ad eccezione di ListPipelines, ma solo con, le pipeline con il tag «environment=test».

JSON

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

Esempio 3: Concedere al proprietario della pipeline l'accesso completo

La seguente politica consente agli utenti di utilizzare tutte le azioni AWS Data Pipeline API, ma solo con le proprie pipeline.

Esempio 4: concedere agli utenti l'accesso alla console AWS Data Pipeline

La policy seguente consente agli utenti di creare e gestire una pipeline utilizzando la console AWS Data Pipeline .

Questa politica include l'azione per `PassRole` le autorizzazioni per risorse specifiche legate a `role`ARN tali AWS Data Pipeline esigenze. Per ulteriori informazioni sull'autorizzazione basata sull'identità (IAM), consulta il post del blog Granting `PassRole` Permission to Launch EC2 Instances with IAM Roles (Permission). PassRole

JSON

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

Ruoli IAM per AWS Data Pipeline

AWS Data Pipeline utilizza AWS Identity and Access Management ruoli. Le politiche di autorizzazione associate ai ruoli IAM determinano quali azioni AWS Data Pipeline e applicazioni possono eseguire e a quali AWS risorse possono accedere. Per ulteriori informazioni, consulta Ruoli IAM nella Guida per l'utente IAM.

AWS Data Pipeline richiede due ruoli IAM:

- Il ruolo pipeline controlla AWS Data Pipeline l'accesso alle tue risorse AWS. Nelle definizioni degli oggetti della pipeline, il `role` campo specifica questo ruolo.
- Il ruolo dell'istanza EC2 controlla l'accesso delle applicazioni in esecuzione su istanze EC2, incluse le istanze EC2 nei cluster Amazon EMR, alle risorse. AWS Nelle definizioni degli oggetti della pipeline, il campo specifica questo ruolo. `resourceRole`

> **Important**
>
> Se hai creato una pipeline prima del 3 ottobre 2022 utilizzando la AWS Data Pipeline console con ruoli predefiniti, l'hai AWS Data Pipeline creata `DataPipelineDefaultRole` per te e ha allegato la politica `AWSDataPipelineRole` gestita al ruolo. A partire dal 3 ottobre 2022, la policy `AWSDataPipelineRole` gestita è obsoleta e il ruolo della pipeline deve essere specificato per una pipeline quando si utilizza la console.
> Ti consigliamo di esaminare le pipeline esistenti e determinare se è associata alla pipeline e se `DataPipelineDefaultRole` è associata a quel ruolo. `AWSDataPipelineRole` In tal caso, esamina l'accesso consentito da questa politica per assicurarti che sia appropriato per i tuoi requisiti di sicurezza. Aggiungi, aggiorna o sostituisci le politiche e le dichiarazioni

politiche allegate a questo ruolo, se necessario. In alternativa, puoi aggiornare una pipeline per utilizzare un ruolo creato con politiche di autorizzazione diverse.

Esempi di politiche di autorizzazione per i ruoli AWS Data Pipeline

A ogni ruolo sono associate una o più politiche di autorizzazione che determinano AWS le risorse a cui il ruolo può accedere e le azioni che il ruolo può eseguire. Questo argomento fornisce un esempio di politica di autorizzazione per il ruolo pipeline. Fornisce inoltre il contenuto di`AmazonEC2RoleforDataPipelineRole`, che è la politica gestita per il ruolo predefinito dell'istanza EC2,. `DataPipelineDefaultResourceRole`

Esempio di politica di autorizzazione dei ruoli Pipeline

La policy di esempio che segue ha lo scopo di consentire funzioni essenziali che AWS Data Pipeline richiedono l'esecuzione di una pipeline con risorse Amazon EC2 e Amazon EMR. Fornisce inoltre le autorizzazioni per accedere ad altre AWS risorse, come Amazon Simple Storage Service e Amazon Simple Notification Service, richieste da molte pipeline. Se gli oggetti definiti in una pipeline non richiedono le risorse di un AWS servizio, ti consigliamo vivamente di rimuovere le autorizzazioni per accedere a quel servizio. Ad esempio, se la pipeline non definisce un'azione DBDataNodo Dynamo o non utilizza l'SnsAlarmazione, si consiglia di rimuovere le istruzioni allow per tali azioni.

- *111122223333*Sostituiscilo con l'ID AWS del tuo account.
- Sostituisci *NameOfDataPipelineRole* con il nome del ruolo della pipeline (il ruolo a cui è associata questa policy).
- Sostituisci *NameOfDataPipelineResourceRole* con il nome del ruolo dell'istanza EC2.
- *us-west-1*Sostituiscilo con la regione appropriata per la tua applicazione.

Politica gestita predefinita per il ruolo dell'istanza EC2

Il contenuto di `AmazonEC2RoleforDataPipelineRole` è mostrato di seguito.
Questa è la politica gestita allegata al ruolo di risorsa predefinito per AWS Data Pipeline,`DataPipelineDefaultResourceRole`. Quando definisci un ruolo di risorsa per la tua pipeline, ti consigliamo di iniziare con questa politica di autorizzazioni e quindi di rimuovere le autorizzazioni per le azioni di AWS servizio che non sono richieste.

Viene mostrata la versione 3 della policy, che è la versione più recente al momento della stesura di questo documento. Visualizza la versione più recente della policy utilizzando la console IAM.

JSON

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

Creazione di ruoli IAM per AWS Data Pipeline e modifica delle autorizzazioni dei ruoli

Utilizza le seguenti procedure per creare ruoli per l' AWS Data Pipeline utilizzo della console IAM. Il processo consiste in due fasi. Innanzitutto, crei una politica di autorizzazioni da associare al ruolo. Successivamente, si crea il ruolo e si allega la politica. Dopo aver creato un ruolo, puoi modificare le autorizzazioni del ruolo allegando e scollegando le politiche di autorizzazione.

> **Note**
>
> Quando crei ruoli per l' AWS Data Pipeline utilizzo della console come descritto di seguito, IAM crea e allega le politiche di fiducia appropriate richieste dal ruolo.

Per creare una politica di autorizzazioni da utilizzare con un ruolo per AWS Data Pipeline

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegliere Policies (Policy) e Create Policy (Crea policy).
3. Scegli la scheda JSON.
4. Se state creando un ruolo di pipeline, copiate e incollate il contenuto dell'esempio di policy inEsempio di politica di autorizzazione dei ruoli Pipeline, modificandolo in base ai vostri requisiti di sicurezza. In alternativa, se stai creando un ruolo di istanza EC2 personalizzato, fai lo stesso per l'esempio in. Politica gestita predefinita per il ruolo dell'istanza EC2
5. Scegliere Esamina policy.
6. Inserisci un nome per la politica, ad esempio, **`MyDataPipelineRolePolicy`** e una descrizione opzionale, quindi scegli Crea politica.
7. Annota il nome della politica. Ne hai bisogno quando crei il tuo ruolo.

Per creare un ruolo IAM per AWS Data Pipeline

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli, quindi scegli Crea ruolo.
3. In Scegli un caso d'uso, scegli Data Pipeline.
4. In Seleziona il tuo caso d'uso, esegui una delle seguenti operazioni:
   - Scegliete `Data Pipeline` di creare un ruolo di pipeline.
   - Scegli `EC2 Role for Data Pipeline` di creare un ruolo di risorsa.
5. Scegli Successivo: autorizzazioni.
6. Se AWS Data Pipeline è elencato il criterio predefinito per, procedi con i seguenti passaggi per creare il ruolo, quindi modificalo in base alle istruzioni della procedura successiva. Altrimenti, inserisci il nome della politica che hai creato nella procedura precedente, quindi selezionala dall'elenco.
7. Scegliete Avanti: tag, inserite i tag da aggiungere al ruolo, quindi scegliete Avanti: revisione.
8. Inserisci un nome per il ruolo, ad esempio, **`MyDataPipelineRole`** e una Descrizione facoltativa, quindi scegli Crea ruolo.

Per allegare o scollegare una politica di autorizzazioni per un ruolo IAM per AWS Data Pipeline

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli
3. Nella casella di ricerca, inizia a digitare il nome del ruolo che desideri modificare, ad esempio DataPipelineDefaultRoleo, MyDataPipelineRolee quindi scegli il nome del ruolo dall'elenco.
4. Nella scheda Autorizzazioni, procedi come segue:

   - Per scollegare un criterio di autorizzazione, in Criteri di autorizzazione, scegli il pulsante di rimozione all'estrema destra della voce del criterio. Scegli Scollega quando ti viene richiesto di confermare.
   - Per allegare una politica creata in precedenza, scegli Allega politiche. Nella casella di ricerca, inizia a digitare il nome della politica che desideri modificare, selezionala dall'elenco, quindi scegli Allega politica.

Modifica dei ruoli per una pipeline esistente

Se desideri assegnare un ruolo di pipeline o un ruolo di risorsa diverso a una pipeline, puoi utilizzare l'editor di architettura nella console. AWS Data Pipeline

Per modificare i ruoli assegnati a una pipeline utilizzando la console

1. Apri la AWS Data Pipeline console all'indirizzo https://console.aws.amazon.com/datapipeline/.
2. Seleziona la pipeline dall'elenco, quindi scegli Azioni, Modifica.
3. Nel riquadro destro dell'editor dell'architetto, scegli Altri.
4. Dagli elenchi Ruolo risorsa e Ruolo, scegli i ruoli AWS Data Pipeline che desideri assegnare, quindi scegli Salva.

Registrazione e monitoraggio AWS Data Pipeline

AWS Data Pipeline è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in AWS Data Pipeline. CloudTrail acquisisce tutte le chiamate API AWS Data Pipeline come eventi. Le chiamate acquisite includono chiamate dalla AWS Data Pipeline console e chiamate di codice alle operazioni AWS Data Pipeline API. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. AWS Data Pipeline Se non configuri un percorso, puoi comunque

visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata AWS Data Pipeline, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

AWS Data Pipeline Informazioni in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in AWS Data Pipeline, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS . Per ulteriori informazioni, vedere Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi di AWS Data Pipeline, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le regioni AWS. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le AWS Data Pipeline azioni vengono registrate CloudTrail e documentate nel capitolo Azioni di riferimento dell'API AWS Data Pipeline. Ad esempio, le chiamate all'CreatePipelineazione generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali di ruolo root o IAM.

- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Comprensione delle AWS Data Pipeline voci dei file di registro

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra l'`CreatePipeline`operazione:


```
    {
"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
  ]
}
```

Risposta agli incidenti in AWS Data Pipeline

La risposta agli incidenti AWS Data Pipeline è una AWS responsabilità. AWS ha una politica e un programma formali e documentati che regolano la risposta agli incidenti.

I problemi operativi di AWS con un ampio impatto sono pubblicati in AWS Service Health Dashboard. Problemi operativi sono anche pubblicati su singoli account tramite il Personal Health Dashboard.

Convalida della conformità per AWS Data Pipeline

AWS Data Pipeline non rientra nell'ambito di alcun programma di conformità AWS. Per un elenco di servizi AWS nell'ambito di programmi di conformità specifici, consulta Servizi AWS coperti dal programma di compliance. Per informazioni, consulta Programmi per la conformità di AWS.

Resilienza in AWS Data Pipeline

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità è possibile progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Sicurezza dell'infrastruttura in AWS Data Pipeline

In quanto servizio gestito, AWS Data Pipeline è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper Amazon Web Services: Overview of Security Processes.

Utilizzi chiamate API AWS pubblicate per accedere AWS Data Pipeline attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versioni successive. È consigliabile TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa è possibile utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Configurazione e analisi delle vulnerabilità in AWS Data Pipeline

La configurazione e i controlli IT sono una responsabilità condivisa tra voi AWS e voi, nostri clienti. Per ulteriori informazioni, consulta il modello di responsabilità AWS condivisa.

Esercitazioni

I seguenti tutorial illustrano il processo di creazione e step-by-step utilizzo delle pipeline con. AWS Data Pipeline

Esercitazioni

- Elaborazione dei dati utilizzando Amazon EMR con Hadoop Streaming
- Copia dati CSV tra bucket Amazon S3 utilizzando AWS Data Pipeline
- Esportazione di dati MySQL su Amazon S3 utilizzando AWS Data Pipeline
- Copia i dati su Amazon Redshift utilizzando AWS Data Pipeline

Elaborazione dei dati utilizzando Amazon EMR con Hadoop Streaming

Puoi utilizzarlo AWS Data Pipeline per gestire i tuoi cluster Amazon EMR. Con AWS Data Pipeline puoi specificare i prerequisiti che devono essere soddisfatti prima dell'avvio del cluster (ad esempio, garantire che i dati odierni siano caricati su Amazon S3), una pianificazione per l'esecuzione ripetuta del cluster e la configurazione del cluster da utilizzare. Il seguente tutorial ti guiderà attraverso l'avvio di un semplice cluster.

In questo tutorial, crei una pipeline per un semplice cluster Amazon EMR per eseguire un job Hadoop Streaming preesistente fornito da Amazon EMR e inviare una notifica Amazon SNS dopo che l'attività è stata completata correttamente. Per questa attività utilizzi la risorsa del cluster Amazon EMR fornita AWS Data Pipeline da. L'applicazione di esempio viene chiamata WordCount e può essere eseguita anche manualmente dalla console Amazon EMR. Tieni presente che i cluster generati da te vengono visualizzati nella console Amazon EMR e fatturati AWS Data Pipeline sul tuo account AWS.

Oggetti della pipeline

La pipeline utilizza i seguenti oggetti:

EmrActivity

Definisce il lavoro da eseguire nella pipeline (eseguire un job Hadoop Streaming preesistente fornito da Amazon EMR).

EmrCluster

AWS Data Pipeline Utilizzo delle risorse per eseguire questa attività.

Un cluster è un insieme di EC2 istanze Amazon. AWS Data Pipeline avvia il cluster e quindi lo termina al termine dell'attività.

Schedule

Data di avvio, ora e durata di questa attività. È anche possibile specificare la data e l'ora di fine.

SnsAlarm

Invia una notifica Amazon SNS all'argomento specificato dopo che l'attività è stata completata correttamente.

Indice

Prima di iniziare

Assicurarsi di aver effettuato le operazioni seguenti.

- Completare le operazioni descritte in Configurazione per AWS Data Pipeline.
- (facoltativo) Imposta un VPC per il cluster e un gruppo di sicurezza per il VPC.
- Creare un argomento per l'invio di e-mail di notifica e annotare l'ARN (Amazon Resource Name) dell'argomento. Per ulteriori informazioni, consultare la pagina relativa alla Creazione di un argomento in Nozioni di base del servizio di notifiche di Amazon Simple.

Avvio di un cluster mediante la riga di comando

Se gestisci regolarmente un cluster Amazon EMR per analizzare log Web o eseguire analisi di dati scientifici, puoi utilizzarlo AWS Data Pipeline per gestire i tuoi cluster Amazon EMR. Con AWS Data Pipeline, puoi specificare le condizioni preliminari che devono essere soddisfatte prima dell'avvio del cluster (ad esempio, assicurandoti che i dati odierni vengano caricati su Amazon S3). Questo tutorial illustra come avviare un cluster che può fungere da modello per una semplice pipeline basata su Amazon EMR o come parte di una pipeline più complessa.

Prerequisiti

Prima di poter usare la CLI, devi completare le fasi seguenti:

1. Installa e configura un'interfaccia a riga di comando (CLI). Per ulteriori informazioni, consulta Accedere AWS Data Pipeline.
2. Assicurati che i ruoli IAM siano denominati DataPipelineDefaultRoleed DataPipelineDefaultResourceRoleesistano. La AWS Data Pipeline console crea questi ruoli automaticamente. Se non hai utilizzato la AWS Data Pipeline console almeno una volta, devi creare questi ruoli manualmente. Per ulteriori informazioni, consulta Ruoli IAM per AWS Data Pipeline.

Processi

- Creazione del file di definizione della pipeline
- Caricamento e attivazione della definizione della pipeline
- Monitorare le esecuzioni della pipeline

Creazione del file di definizione della pipeline

Il codice seguente è il file di definizione della pipeline per un semplice cluster Amazon EMR che esegue un job di streaming Hadoop esistente fornito da Amazon EMR. Questa applicazione di esempio viene chiamata WordCount ed è possibile eseguirla anche utilizzando la console Amazon EMR.

Copiare questo codice in un file di testo e salvarlo come `MyEmrPipelineDefinition.json`. È necessario sostituire la posizione del bucket Amazon S3 con il nome di un bucket Amazon S3 di tua proprietà. È inoltre necessario sostituire le date di inizio e fine. Per lanciare i cluster immediatamente, imposta `startDateTime` una data un giorno nel passato e `endDateTime` un giorno nelle future. AWS Data Pipeline inizia quindi a lanciare immediatamente i cluster «scaduti» nel tentativo di risolvere ciò che percepisce come un arretrato di lavoro. Questo backfilling significa che non è necessario attendere un'ora per vedere AWS Data Pipeline il lancio del primo cluster.

```
{
  "objects": [
    {
      "id": "Hourly",
      "type": "Schedule",
      "startDateTime": "2012-11-19T07:48:00",
```

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

Questa pipeline ha tre oggetti:

- `Hourly`, che rappresenta la pianificazione del lavoro. È possibile impostare una pianificazione come uno dei campi di un'attività. Quando si effettua questa operazione, l'attività viene eseguita in base alla pianificazione, oppure in questo caso, su base oraria.
- `MyCluster`, che rappresenta il set di EC2 istanze Amazon utilizzate per eseguire il cluster. Puoi specificare la dimensione e il numero di EC2 istanze da eseguire come cluster. Se non si specifica il numero di istanze, il cluster ne usa due, un nodo master e un nodo di task. È possibile specificare una sottorete in cui avviare il cluster. Puoi aggiungere configurazioni aggiuntive al cluster, ad esempio azioni di bootstrap per caricare software aggiuntivo sull'AMI fornita da Amazon EMR.
- `MyEmrActivity`, che rappresenta il calcolo da elaborare con il cluster. Amazon EMR supporta diversi tipi di cluster, tra cui streaming, Cascading e Scripted Hive. Il `runsOn` campo fa riferimento a MyCluster, utilizzandolo come specifica per le basi del cluster.

Caricamento e attivazione della definizione della pipeline

È necessario caricare la definizione della pipeline e attivare la pipeline. Nei seguenti comandi di esempio, sostituiteli *pipeline_name* con un'etichetta per la pipeline e *pipeline_file* con il percorso completo per il file di definizione della pipeline. `.json`

AWS CLI

Per creare la definizione della pipeline e attivare la pipeline, utilizzate il seguente comando create-pipeline. Annota l'ID della pipeline, poiché utilizzerai questo valore con la maggior parte dei comandi CLI.

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

Per caricare la definizione della pipeline, utilizzate il seguente comando. put-pipeline-definition

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

Se la pipeline viene convalidata correttamente, il `validationErrors` campo è vuoto. È necessario esaminare eventuali avvertenze.

Per attivare la pipeline, usa il seguente comando activate-pipeline.

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

È possibile verificare che la pipeline venga visualizzata nell'elenco delle pipeline utilizzando il seguente comando list-pipelines.

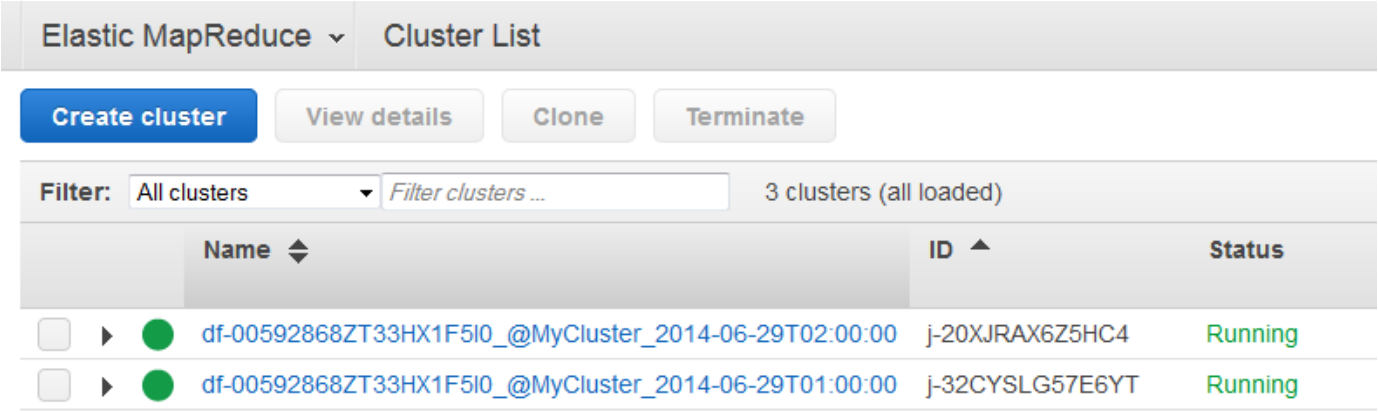
```
aws datapipeline list-pipelines
```

Monitorare le esecuzioni della pipeline

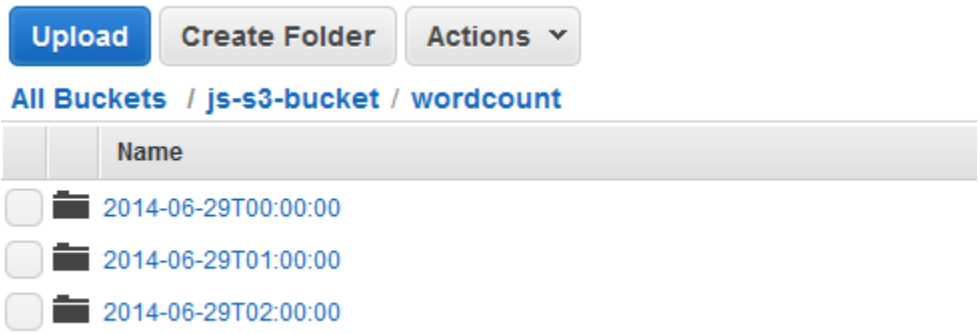
È possibile visualizzare i cluster avviati AWS Data Pipeline utilizzando la console Amazon EMR e visualizzare la cartella di output utilizzando la console Amazon S3.

Per verificare lo stato di avanzamento dei cluster lanciati da AWS Data Pipeline

1. Apri la console Amazon EMR.
2. I cluster generati da AWS Data Pipeline hanno un nome formattato come segue: _@ _. *<pipeline-identifier> <emr-cluster-name> <launch-time>*



3. Al termine di una delle esecuzioni, apri la console Amazon S3 e verifica che la cartella di output con data e ora esista e contenga i risultati previsti del cluster.



Copia dati CSV tra bucket Amazon S3 utilizzando AWS Data Pipeline

Dopo aver letto Che cos'è AWS Data Pipeline? e deciso cosa utilizzare per AWS Data Pipeline automatizzare lo spostamento e la trasformazione dei dati, è il momento di iniziare a creare pipeline di dati. Per aiutare a comprendere come funziona AWS Data Pipeline , esaminiamo un'attività semplice.

Questo tutorial illustra il processo di creazione di una pipeline di dati per copiare i dati da un bucket Amazon S3 a un altro e quindi inviare una notifica Amazon SNS dopo il completamento dell'attività di copia. Per questa attività di copia, utilizzi un' EC2 istanza gestita da AWS Data Pipeline .

Oggetti della pipeline

La pipeline utilizza i seguenti oggetti:

CopyActivity

L'attività svolta per AWS Data Pipeline questa pipeline (copia dei dati CSV da un bucket Amazon S3 a un altro).

> **Important**
>
> Esistono dei limiti quando si usa il formato di file CSV con `CopyActivity` e `S3DataNode`. Per ulteriori informazioni, consulta CopyActivity.

Schedule

Data di avvio, ora e ricorrenza di questa attività. È anche possibile specificare la data e l'ora di fine.

Ec2Resource

La risorsa (un' EC2 istanza) che AWS Data Pipeline viene utilizzata per eseguire questa attività.

S3 DataNode

I nodi di input e output (bucket Amazon S3) per questa pipeline.

SnsAlarm

L'azione AWS Data Pipeline deve essere eseguita quando vengono soddisfatte le condizioni specificate (inviare notifiche Amazon SNS a un argomento dopo che l'attività è stata completata correttamente).

Indice

Prima di iniziare

Assicurarsi di aver effettuato le operazioni seguenti.

- Completare le operazioni descritte in Configurazione per AWS Data Pipeline.
- (facoltativo) Impostare un VPC per l'istanza e un gruppo di sicurezza per il VPC.
- Crea un bucket Amazon S3 come fonte di dati.

  Per ulteriori informazioni, consulta Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.
- Carica i tuoi dati nel tuo bucket Amazon S3.

  Per ulteriori informazioni, consulta Aggiunta di un oggetto a un bucket nella Guida per l'utente di Amazon Simple Storage Service.
- Crea un altro bucket Amazon S3 come destinazione dati
- Creare un argomento per l'invio di e-mail di notifica e annotare l'ARN (Amazon Resource Name) dell'argomento. Per ulteriori informazioni, consultare la pagina relativa alla Creazione di un argomento in Nozioni di base del servizio di notifiche di Amazon Simple.
- (Facoltativo) Questo tutorial utilizza le policy del ruolo IAM di default create da AWS Data Pipeline. Se preferisci creare e configurare la tua politica di ruolo IAM e le tue relazioni di fiducia, segui le istruzioni descritte in. Ruoli IAM per AWS Data Pipeline

Copia dei dati CSV tramite la riga di comando

Puoi creare e utilizzare pipeline per copiare dati da un bucket Amazon S3 a un altro.

Prerequisiti

Prima di iniziare , devi completare le fasi seguenti:

1. Installa e configura un'interfaccia a riga di comando (CLI). Per ulteriori informazioni, consulta Accedere AWS Data Pipeline.
2. Assicurati che i ruoli IAM siano denominati DataPipelineDefaultRoleed DataPipelineDefaultResourceRoleesistano. La AWS Data Pipeline console crea questi ruoli automaticamente. Se non hai utilizzato la AWS Data Pipeline console almeno una volta, devi creare questi ruoli manualmente. Per ulteriori informazioni, consulta Ruoli IAM per AWS Data Pipeline.

Processi

- Definire una pipeline in formato JSON
- Caricamento e attivazione della definizione della pipeline

Definire una pipeline in formato JSON

Questo scenario di esempio mostra come utilizzare le definizioni di pipeline JSON e la AWS Data Pipeline CLI per pianificare la copia dei dati tra due bucket Amazon S3 a un intervallo di tempo specifico. Questo è il file JSON completo di definizione della pipeline seguito da una spiegazione per ciascuna delle sue sezioni.

> **Note**
>
> È consigliabile utilizzare un editor di testo che può aiutare a verificare la sintassi di file in formato JSON e nominare il file utilizzando l'estensione del file .json.

In questo esempio, per chiarezza, abbiamo saltato i campi facoltativi e abbiamo mostrato solo i campi obbligatori. Il file JSON completo della pipeline per questo esempio è:

```
{
  "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://amzn-s3-demo-bucket/source/inputfile.csv"
    },
    {
      "id": "S3Output",
      "type": "S3DataNode",
```

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

Schedule

La pipeline definisce una pianificazione con un data di inizio e di fine, insieme a un periodo per stabilire con quale frequenza viene eseguita l'attività in questa pipeline.

```
{
  "id": "MySchedule",
  "type": "Schedule",
  "startDateTime": "2013-08-18T00:00:00",
```

```
  "endDateTime": "2013-08-19T00:00:00",
  "period": "1 day"
},
```

Nodi di dati Amazon S3

Successivamente, il componente della DataNode pipeline di input S3 definisce una posizione per i file di input; in questo caso, una posizione del bucket Amazon S3. Il DataNode componente di input S3 è definito dai seguenti campi:

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

Id

Il nome definito dall'utente per il percorso di input (un'etichetta solo di riferimento).

Tipo

Il tipo di componente della pipeline, che è «DataNodeS3" per corrispondere alla posizione in cui risiedono i dati, in un bucket Amazon S3.

Schedule

Un riferimento al componente di pianificazione che abbiamo creato nelle righe precedenti del file JSON denominato «». MySchedule

Path

Il percorso ai dati associati al nodo di dati. La sintassi per un nodo di dati è determinata dal tipo. Ad esempio, la sintassi per un percorso Amazon S3 segue una sintassi diversa appropriata per una tabella di database.

Successivamente, il DataNode componente di output S3 definisce la posizione di destinazione dell'output per i dati. Segue lo stesso formato del DataNode componente S3 di input, ad eccezione del nome del componente e di un percorso diverso per indicare il file di destinazione.

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

Risorsa

Questa è una definizione della risorsa di calcolo che esegue l'operazione di copia. In questo esempio, AWS Data Pipeline dovrebbe creare automaticamente un' EC2 istanza per eseguire l'attività di copia e terminare la risorsa al termine dell'attività. I campi qui definiti controllano la creazione e la funzione dell' EC2 istanza che esegue il lavoro. La EC2 risorsa è definita dai seguenti campi:

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

Id

Il nome definito dall'utente per la pianificazione della pipeline, un'etichetta solo di riferimento.

Tipo

Il tipo di risorsa computazionale per eseguire il lavoro, in questo caso un' EC2 istanza. Sono disponibili altri tipi di risorse, ad esempio un EmrCluster tipo.

Schedule

La pianificazione su cui creare questa risorsa di calcolo.

instanceType

La dimensione dell' EC2 istanza da creare. Assicurati di impostare la dimensione appropriata dell' EC2istanza che meglio corrisponde al carico del lavoro che desideri eseguire AWS Data Pipeline.

In questo caso, impostiamo un'istanza m1.medium EC2. Per ulteriori informazioni sui diversi tipi di istanza e su quando utilizzarli, consulta l'argomento Amazon EC2 Instance Types all'indirizzo http://aws.amazon. com/ec2/instance-tipi/.

Ruolo

Il ruolo IAM dell'account che accede alle risorse, ad esempio l'accesso a un bucket Amazon S3 per recuperare i dati.

resourceRole

Il ruolo IAM dell'account che crea risorse, ad esempio la creazione e la configurazione di un' EC2istanza per tuo conto. Ruolo e ResourceRole 3 possono essere lo stesso ruolo, ma forniscono separatamente una maggiore granularità nella configurazione di sicurezza.

Attività

L'ultima sezione del file JSON è la definizione dell'attività che rappresenta il lavoro da eseguire. Questo esempio utilizza `CopyActivity` la copia dei dati da un file CSV in un file http://aws.amazon. com/ec2/instance-types/ bucket su un altro. Questo componente `CopyActivity` è definito dai campi seguenti:

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

Id

Il nome definito dall'utente per l'attività, un'etichetta solo di riferimento.

Tipo

Il tipo di attività da svolgere, ad esempio. MyCopyActivity

runsOn

La risorsa di calcolo che esegue il lavoro definito dall'attività. In questo esempio, forniamo un riferimento all' EC2istanza definita in precedenza. L'utilizzo del `runs0n` campo AWS Data Pipeline consente di creare l' EC2 istanza automaticamente. Il campo `runs0n` indica che la risorsa è disponibile nell'infrastruttura AWS, mentre il valore `workerGroup` indica che si desidera utilizzare le proprie risorse locali per eseguire il lavoro.

Input

Posizione dei dati da copiare.

Output

Dati del percorso di destinazione.

Schedule

La pianificazione su cui eseguire questa attività.

Caricamento e attivazione della definizione della pipeline

È necessario caricare la definizione della pipeline e attivare la pipeline. Nei seguenti comandi di esempio, sostituiteli *`pipeline_name`* con un'etichetta per la pipeline e *`pipeline_file`* con il percorso completo per il file di definizione della pipeline. `.json`

AWS CLI

Per creare la definizione della pipeline e attivare la pipeline, utilizzate il seguente comando create-pipeline. Annota l'ID della pipeline, poiché utilizzerai questo valore con la maggior parte dei comandi CLI.

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

Per caricare la definizione della pipeline, utilizzate il seguente comando. put-pipeline-definition

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

Se la pipeline viene convalidata correttamente, il `validationErrors` campo è vuoto. È necessario esaminare eventuali avvertenze.

Per attivare la pipeline, usa il seguente comando activate-pipeline.

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

È possibile verificare che la pipeline venga visualizzata nell'elenco delle pipeline utilizzando il seguente comando list-pipelines.

```
aws datapipeline list-pipelines
```

Esportazione di dati MySQL su Amazon S3 utilizzando AWS Data Pipeline

Questo tutorial illustra il processo di creazione di una pipeline di dati per copiare i dati (righe) da una tabella nel database MySQL a un file CSV (valori separati da virgole) in un bucket Amazon S3 e quindi inviare una notifica Amazon SNS dopo il completamento dell'attività di copia. Utilizzerai un'istanza fornita da per questa attività di copia. EC2 AWS Data Pipeline

Oggetti della pipeline

La pipeline utilizza i seguenti oggetti:

- CopyActivity
- Ec2Resource
- MySqlDataNode
- S3 DataNode
- SnsAlarm

Indice

- Copia dei dati su MySQL tramite la riga di comando

Prima di iniziare

Assicurarsi di aver effettuato le operazioni seguenti.

- Completare le operazioni descritte in Configurazione per AWS Data Pipeline.
- (facoltativo) Impostare un VPC per l'istanza e un gruppo di sicurezza per il VPC.
- Crea un bucket Amazon S3 come output di dati.

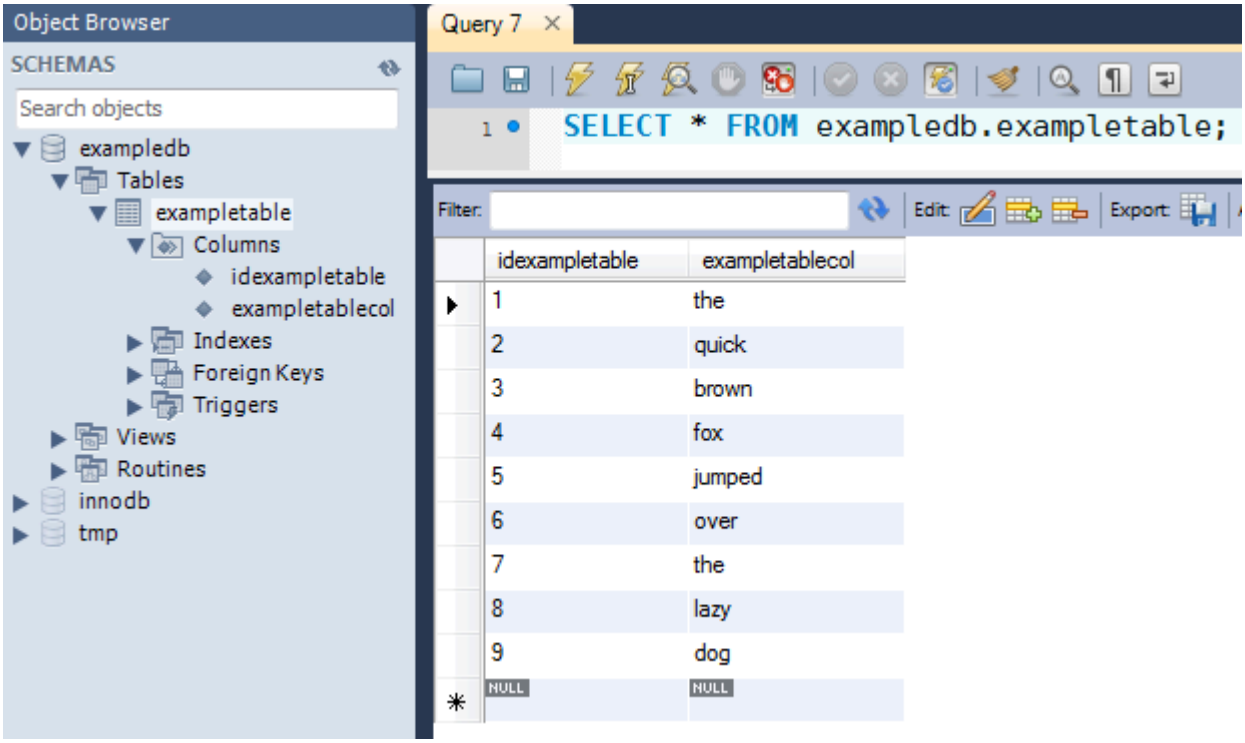
  Per ulteriori informazioni, consulta la Guida per l'utente di Create a Bucket in Amazon Simple Storage Service.
- Creare e avviare un'istanza di database MySQL come origine dati.

  Per ulteriori informazioni, consulta Launch a DB nella Amazon RDS Getting Started Guide. Dopo aver creato un'istanza Amazon RDS, consulta Create a Table nella documentazione MySQL.

  > **Note**
  >
  > Annotare il nome utente e la password che hai utilizzato per creare l'istanza di MySQL. Una volta avviata l'istanza di database MySQL, annotare l'endpoint dell'istanza. Queste informazioni serviranno in seguito.

- Collegare la tua istanza di database MySQL, creare una tabella e quindi aggiungere i valori dei dati di prova alla nuova tabella creata.

  A scopo illustrativo, abbiamo creato questo tutorial utilizzando una tabella di MySQL con i seguenti dati di esempio e la seguente configurazione. La seguente schermata viene da MySQL Workbench 5.2 CE:



Per ulteriori informazioni, consulta Creare una tabella nella documentazione di MySQL e nella pagina dei prodotti di MySQL Workbench.

- Creare un argomento per l'invio di e-mail di notifica e annotare l'ARN (Amazon Resource Name) dell'argomento. Per ulteriori informazioni, consulta la Guida introduttiva alla creazione di un argomento in Amazon Simple Notification Service.
- (Facoltativo) Questo tutorial utilizza le politiche di ruolo IAM predefinite create da AWS Data Pipeline. Se preferisci creare e configurare la tua policy di ruolo IAM e le tue relazioni di fiducia, segui le istruzioni descritte inRuoli IAM per AWS Data Pipeline.

Copia dei dati su MySQL tramite la riga di comando

Puoi creare una pipeline per copiare i dati da una tabella MySQL a un file in un bucket Amazon S3.

Prerequisiti

Prima di iniziare , devi completare le fasi seguenti:

1. Installa e configura un'interfaccia a riga di comando (CLI). Per ulteriori informazioni, consulta Accedere AWS Data Pipeline.

2. Assicurati che i ruoli IAM siano denominati DataPipelineDefaultRoleed DataPipelineDefaultResourceRoleesistano. La AWS Data Pipeline console crea questi ruoli automaticamente. Se non hai utilizzato la AWS Data Pipeline console almeno una volta, devi creare questi ruoli manualmente. Per ulteriori informazioni, consulta Ruoli IAM per AWS Data Pipeline.
3. Configura un bucket Amazon S3 e un'istanza Amazon RDS. Per ulteriori informazioni, consulta Prima di iniziare.

Processi

- Definire una pipeline in formato JSON
- Caricamento e attivazione della definizione della pipeline

Definire una pipeline in formato JSON

Questo scenario di esempio mostra come utilizzare le definizioni della pipeline JSON e la AWS Data Pipeline CLI per copiare dati (righe) da una tabella in un database MySQL a un file CSV (valori separati da virgole) in un bucket Amazon S3 a un intervallo di tempo specificato.

Questo è il file JSON completo di definizione della pipeline seguito da una spiegazione per ciascuna delle sue sezioni.

> **Note**
>
> È consigliabile utilizzare un editor di testo che può aiutare a verificare la sintassi di file in formato JSON e nominare il file utilizzando l'estensione del file .json.

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

```
      "input": {
        "ref": "MySqlDataNodeId115"
      },
      "schedule": {
        "ref": "ScheduleId113"
      },
      "name": "My Copy",
      "runsOn": {
        "ref": "Ec2ResourceId116"
      },
      "onSuccess": {
        "ref": "ActionId1"
      },
      "onFail": {
        "ref": "SnsAlarmId117"
      },
      "output": {
        "ref": "S3DataNodeId114"
      },
      "type": "CopyActivity"
    },
    {
      "id": "S3DataNodeId114",
      "schedule": {
        "ref": "ScheduleId113"
      },
      "filePath": "s3://amzn-s3-demo-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"
    }
  ]
}
```

Nodo di dati MySQL

Il componente della MySqlDataNode pipeline di input definisce una posizione per i dati di input; in questo caso, un'istanza Amazon RDS. Il MySqlDataNode componente di input è definito dai seguenti campi:

```
{
  "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

Il nome definito dall'utente, un'etichetta solo di riferimento.

Username

Il nome utente dell'account di database che dispone di autorizzazioni sufficienti per recuperare i dati dalla tabella di database. Sostituisci *my-username* con il nome del tuo utente.

Schedule

Un riferimento al componente di pianificazione che abbiamo creato nelle righe precedenti del file JSON.

Name

Il nome definito dall'utente, un'etichetta solo di riferimento.

*Password

La password per l'account del database con il prefisso asterisco per indicare che AWS Data Pipeline deve crittografare il valore della password. *my-password*Sostituiscila con la password corretta per il tuo utente. Il campo password è preceduto dal carattere speciale dell'asterisco. Per ulteriori informazioni, consulta Caratteri speciali.

Tabella

Il nome della tabella del database che contiene i dati da copiare. Sostituiscila *table-name* con il nome della tabella del database.

connectionString

La stringa di connessione JDBC per l' CopyActivity oggetto da connettere al database.

selectQuery

Una query SQL SELECT valida che specifichi quali dati copiare dalla tabella di database. Si noti che `#{table}` è un'espressione che riutilizza il nome della tabella fornito dalla variabile "tabella" nelle righe precedenti del file JSON.

Tipo

Il SqlDataNode tipo, che è un'istanza Amazon RDS che utilizza MySQL in questo esempio.

> **Note**
>
> Il MySqlDataNode tipo è obsoleto. Sebbene sia ancora possibile utilizzarlo MySqlDataNode, si consiglia di utilizzare. SqlDataNode

Nodo dati Amazon S3

Successivamente, il componente della pipeline S3Output definisce una posizione per il file di output; in questo caso un file CSV in una posizione del bucket Amazon S3. Il DataNode componente di output S3 è definito dai seguenti campi:

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

Id

L'ID definito dall'utente, un'etichetta solo di riferimento.

Schedule

Un riferimento al componente di pianificazione che abbiamo creato nelle righe precedenti del file JSON.

filePath

Il percorso ai dati associati al nodo di dati, cioè il file di output CSV in questo esempio.

Name

Il nome definito dall'utente, un'etichetta solo di riferimento.

Tipo

Il tipo di oggetto della pipeline, che è S3 in base DataNode alla posizione in cui risiedono i dati, in un bucket Amazon S3.

Risorsa

Questa è una definizione della risorsa di calcolo che esegue l'operazione di copia. In questo esempio, AWS Data Pipeline dovrebbe creare automaticamente un' EC2 istanza per eseguire l'attività di copia e terminare la risorsa al termine dell'attività. I campi qui definiti controllano la creazione e la funzione dell' EC2 istanza che esegue il lavoro. La EC2 risorsa è definita dai seguenti campi:

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

Id

L'ID definito dall'utente, un'etichetta solo di riferimento.

Schedule

La pianificazione su cui creare questa risorsa di calcolo.

Name

Il nome definito dall'utente, un'etichetta solo di riferimento.

Ruolo

Il ruolo IAM dell'account che accede alle risorse, ad esempio l'accesso a un bucket Amazon S3 per recuperare i dati.

Tipo

Il tipo di risorsa computazionale per eseguire il lavoro; in questo caso, un'istanza. EC2 Sono disponibili altri tipi di risorse, ad esempio un EmrCluster tipo.

resourceRole

Il ruolo IAM dell'account che crea risorse, ad esempio la creazione e la configurazione di un' EC2istanza per tuo conto. Ruolo e ResourceRole 3 possono essere lo stesso ruolo, ma forniscono separatamente una maggiore granularità nella configurazione di sicurezza.

Attività

L'ultima sezione del file JSON è la definizione dell'attività che rappresenta il lavoro da eseguire. In questo caso utilizziamo un CopyActivity componente per copiare i dati da un file in un bucket Amazon S3 a un altro file. Il CopyActivity componente è definito dai seguenti campi:

```
{
  "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

L'ID definito dall'utente, un'etichetta solo di riferimento

Input

Posizione dei dati MySQL da copiare

Schedule

La pianificazione su cui eseguire questa attività

Name

Il nome definito dall'utente, un'etichetta solo di riferimento

runsOn

La risorsa di calcolo che esegue il lavoro definito dall'attività. In questo esempio, forniamo un riferimento all' EC2 istanza definita in precedenza. L'utilizzo del `runsOn` campo AWS Data Pipeline consente di creare l' EC2istanza automaticamente. Il campo `runsOn` indica che la risorsa è disponibile nell'infrastruttura AWS, mentre il valore workerGroup indica che si desidera utilizzare le proprie risorse locali per eseguire il lavoro.

onSuccess

SnsAlarm da inviare se l'attività viene completata correttamente

onFail

SnsAlarm da inviare se l'attività non viene completata correttamente

Output

La posizione Amazon S3 del file di output CSV

Tipo

Il tipo di attività da eseguire.

Caricamento e attivazione della definizione della pipeline

È necessario caricare la definizione della pipeline e attivare la pipeline. Nei seguenti comandi di esempio, sostituiteli *pipeline_name* con un'etichetta per la pipeline e *pipeline_file* con il percorso completo per il file di definizione della pipeline. `.json`

AWS CLI

Per creare la definizione della pipeline e attivare la pipeline, utilizzate il seguente comando create-pipeline. Annota l'ID della pipeline, poiché utilizzerai questo valore con la maggior parte dei comandi CLI.

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

Per caricare la definizione della pipeline, utilizzate il seguente comando. put-pipeline-definition

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

Se la pipeline viene convalidata correttamente, il `validationErrors` campo è vuoto. È necessario esaminare eventuali avvertenze.

Per attivare la pipeline, usa il seguente comando activate-pipeline.

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

È possibile verificare che la pipeline venga visualizzata nell'elenco delle pipeline utilizzando il seguente comando list-pipelines.

```
aws datapipeline list-pipelines
```

Copia i dati su Amazon Redshift utilizzando AWS Data Pipeline

Questo tutorial illustra il processo di creazione di una pipeline che sposta periodicamente i dati da Amazon S3 ad Amazon Redshift utilizzando il modello Copy to Redshift nella console o un file di definizione della pipeline con AWS Data Pipeline la CLI. AWS Data Pipeline

Amazon S3 è un servizio Web che consente di archiviare dati nel cloud. Per ulteriori informazioni, consultare la Guida per l'utente di Amazon Simple archiviazione Service.

Amazon Redshift è un servizio di data warehouse nel cloud. Per ulteriori informazioni, consulta la Amazon Redshift Management Guide.

Questo tutorial ha diversi prerequisiti. Dopo aver completato i passaggi seguenti, è possibile proseguire utilizzando la console o l'interfaccia a riga di comando.

Indice

Prima di iniziare: configura le opzioni COPY e di caricamento dati

Prima di copiare i dati su Amazon Redshift AWS Data Pipeline Within, assicurati di:

- Carica dati da Amazon S3.
- Configura l'`COPY`attività in Amazon Redshift.

Quando queste opzioni sono attive e completano correttamente un caricamento dati, trasferiscile ad AWS Data Pipeline per eseguire la copia al suo interno.

Per `COPY` le opzioni, consulta COPY nella Amazon Redshift Database Developer Guide.

Per la procedura di caricamento dei dati da Amazon S3, consulta Loading data from Amazon S3 nella Amazon Redshift Database Developer Guide.

Ad esempio, il seguente comando SQL in Amazon Redshift crea una nuova tabella denominata `LISTING` e copia dati di esempio da un bucket disponibile pubblicamente in Amazon S3.

Sostituire `<iam-role-arn>` e la regione con la propria.

Per dettagli su questo esempio, consulta Load Sample Data from Amazon S3 nella 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';
```

Configura Pipeline, crea un gruppo di sicurezza e crea un cluster Amazon Redshift

Per configurare il tutorial

1. Completare le operazioni descritte in Configurazione per AWS Data Pipeline.
2. Creare un gruppo di sicurezza.
   a. Aprire la console Amazon EC2.
   b. Nel riquadro di navigazione fare clic su Security Groups (Gruppi di sicurezza).
   c. Fare clic su Create Security Group (Crea un gruppo di sicurezza).
   d. Specificare un nome e una descrizione per il gruppo di sicurezza.
   e. [EC2-Classic] Seleziona `No VPC` per VPC.
   f. [EC2-VPC] Seleziona l'ID del VPC per VPC.
   g. Fai clic su Create (Crea).
3. [EC2-Classic] Crea un gruppo di sicurezza del cluster Amazon Redshift e specifica il gruppo di sicurezza Amazon EC2.
   a. Apri la console Amazon Redshift.
   b. Nel riquadro di navigazione fare clic su Security Groups (Gruppi di sicurezza).
   c. Fai clic su Create Cluster Security Group (Crea gruppo di sicurezza cluster).
   d. Nella finestra di dialogo Create Cluster Security Group (Crea gruppo di sicurezza cluster) specificare un nome e una descrizione per il gruppo di sicurezza de cluster.
   e. Fai clic sul nome del nuovo gruppo di sicurezza del cluster.

f. Fai clic su Add Connection Type (Aggiungi tipo di connessione).

g. Nella finestra di dialogo Add Connection Type (Aggiungi tipo di connessione), selezionare EC2 Security Group (Gruppo di sicurezza EC2) da Connection Type (Tipo di connessione), selezionare il gruppo di sicurezza creato da EC2 Security Group Name (Nome gruppo di sicurezza EC2), quindi fare clic su Authorize (Autorizza).

4. [EC2-VPC] Crea un gruppo di sicurezza del cluster Amazon Redshift e specifica il gruppo di sicurezza VPC.

   a. Aprire la console Amazon EC2.

   b. Nel riquadro di navigazione fare clic su Security Groups (Gruppi di sicurezza).

   c. Fare clic su Create Security Group (Crea un gruppo di sicurezza).

   d. Nella finestra di dialogo Create Security Group (Crea gruppo di sicurezza), specificare un nome e una descrizione per il gruppo di sicurezza e selezionare l'ID del tuo VPC per VPC.

   e. Fai clic su Add Rule (Aggiungi regola). Specificare il tipo, il protocollo e l'intervallo di porte e iniziare a digitare l'ID del gruppo di sicurezza in Source (Origine). Selezionare il gruppo di sicurezza creato nel secondo passaggio.

   f. Fai clic su Create (Crea).

5. Di seguito è riportato un riepilogo delle fasi.

   Se disponi di un cluster Amazon Redshift esistente, prendi nota dell'ID del cluster.

   Per creare un nuovo cluster e caricare dati di esempio, segui i passaggi descritti in Getting Started with Amazon Redshift. Per ulteriori informazioni sulla creazione di cluster, consulta Creating a Cluster nella Amazon Redshift Management Guide.

   a. Apri la console Amazon Redshift.

   b. Fai clic su Launch Cluster (Avvia cluster).

   c. Fornire le informazioni richieste per il cluster, quindi fare clic su Continue (Continua).

   d. Fornire la configurazione del nodo, quindi fare clic su Continue (Continua).

   e. Nella pagina per ulteriori informazioni sulla configurazione, selezionare il gruppo di sicurezza del cluster creato, quindi fare clic su Continue (Continua).

   f. Esaminare le specifiche per il cluster, quindi fare clic su Launch Cluster (Avvia cluster).

Copia i dati su Amazon Redshift utilizzando la riga di comando

Questo tutorial dimostra come copiare dati da Amazon S3 ad Amazon Redshift. Creerai una nuova tabella in Amazon Redshift e la utilizzerai AWS Data Pipeline per trasferire i dati su questa tabella da un bucket Amazon S3 pubblico, che contiene dati di input di esempio in formato CSV. I log vengono salvati in un bucket Amazon S3 di tua proprietà.

Amazon S3 è un servizio Web che consente di archiviare dati nel cloud. Per ulteriori informazioni, consultare la Guida per l'utente di Amazon Simple archiviazione Service. Amazon Redshift è un servizio di data warehouse nel cloud. Per ulteriori informazioni, consulta la Amazon Redshift Management Guide.

Prerequisiti

Prima di iniziare , devi completare le fasi seguenti:

1. Installa e configura un'interfaccia a riga di comando (CLI). Per ulteriori informazioni, consulta Accedere AWS Data Pipeline.
2. Assicurati che i ruoli IAM siano denominati DataPipelineDefaultRoleed DataPipelineDefaultResourceRoleesistano. La AWS Data Pipeline console crea questi ruoli automaticamente. Se non hai utilizzato la AWS Data Pipeline console almeno una volta, devi creare questi ruoli manualmente. Per ulteriori informazioni, consulta Ruoli IAM per AWS Data Pipeline.
3. Imposta il `COPY` comando in Amazon Redshift, poiché avrai bisogno che queste stesse opzioni funzionino quando esegui la copia all'interno. AWS Data Pipeline Per informazioni, consulta Prima di iniziare: configura le opzioni COPY e di caricamento dati.
4. Configura un database Amazon Redshift. Per ulteriori informazioni, consulta Configura Pipeline, crea un gruppo di sicurezza e crea un cluster Amazon Redshift.

Processi

- Definire una pipeline in formato JSON
- Caricamento e attivazione della definizione della pipeline

Definire una pipeline in formato JSON

Questo scenario di esempio mostra come copiare i dati da un bucket Amazon S3 ad Amazon Redshift.

Questo è il file JSON completo di definizione della pipeline seguito da una spiegazione per ciascuna delle sue sezioni. È consigliabile utilizzare un editor di testo che può aiutare a verificare la sintassi di file in formato JSON e nominare il file utilizzando l'estensione del file `.json`.

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

```
    {
      "id": "Ec2ResourceId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "securityGroups": "MySecurityGroup",
      "name": "DefaultEc2Resource1",
      "role": "DataPipelineDefaultRole",
      "logUri": "s3://myLogs",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "type": "Ec2Resource"
    },
    {
      "id": "ScheduleId1",
      "startDateTime": "yyyy-mm-ddT00:00:00",
      "name": "DefaultSchedule1",
      "type": "Schedule",
      "period": "period",
      "endDateTime": "yyyy-mm-ddT00:00:00"
    },
    {
      "id": "S3DataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
      "name": "DefaultS3DataNode1",
      "dataFormat": {
        "ref": "CSVId1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "RedshiftCopyActivityId1",
      "input": {
        "ref": "S3DataNodeId1"
      },
      "schedule": {
        "ref": "ScheduleId1"
      },
      "insertMode": "KEEP_EXISTING",
      "name": "DefaultRedshiftCopyActivity1",
      "runsOn": {
        "ref": "Ec2ResourceId1"
```

```
      },
      "type": "RedshiftCopyActivity",
      "output": {
        "ref": "RedshiftDataNodeId1"
      }
    }
  ]
}
```

Per ulteriori informazioni su questi oggetti, consulta la seguente documentazione.

Oggetti

- Nodi di dati
- Risorsa
- Attività

Nodi di dati

Questo esempio utilizza un nodo di dati di input, un nodo di dati di output e un database.

Nodo di dati di input

Il componente della `S3DataNode` pipeline di input definisce la posizione dei dati di input in Amazon S3 e il formato dei dati di input. Per ulteriori informazioni, consulta S3 DataNode.

Questo componente di input è definito dai campi seguenti:

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

`id`

L'ID definito dall'utente, un'etichetta solo di riferimento.

`schedule`

Un riferimento al componente di pianificazione.

`filePath`

Il percorso ai dati associati al nodo di dati, cioè il file di input CSV in questo esempio.

`name`

Il nome definito dall'utente, un'etichetta solo di riferimento.

`dataFormat`

Un riferimento al formato dei dati dell'attività da elaborare.

Nodo dei dati di output

Il componente della `RedshiftDataNode` pipeline di output definisce una posizione per i dati di output; in questo caso, una tabella in un database Amazon Redshift. Per ulteriori informazioni, consulta RedshiftDataNode. Questo componente di output è definito dai campi seguenti:

```
{
  "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`

L'ID definito dall'utente, un'etichetta solo di riferimento.

`schedule`

Un riferimento al componente di pianificazione.

`tableName`

Nome della tabella Amazon Redshift.

`name`

Il nome definito dall'utente, un'etichetta solo di riferimento.

`createTableSql`

Un'espressione SQL per creare la tabella nel database.

`database`

Un riferimento al database Amazon Redshift.

Database

Questo componente `RedshiftDatabase` è definito dai campi seguenti. Per ulteriori informazioni, consulta RedshiftDatabase.

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

`id`

L'ID definito dall'utente, un'etichetta solo di riferimento.

`databaseName`

Il nome del database logico.

`username`

Il nome utente da fornire durante la connessione al database.

`name`

Il nome definito dall'utente, un'etichetta solo di riferimento.

`password`

La password per la connessione al database.

`clusterId`

L'ID del cluster Redshift.

Risorsa

Questa è una definizione della risorsa di calcolo che esegue l'operazione di copia. In questo esempio, AWS Data Pipeline dovrebbe creare automaticamente un'istanza EC2 per eseguire l'attività di copia e terminare l'istanza al termine dell'attività. I campi definiti qui controllano la creazione e la funzione dell'istanza che esegue il lavoro. Per ulteriori informazioni, consulta Ec2Resource.

Questo componente `Ec2Resource` è definito dai campi seguenti:

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

`id`

L'ID definito dall'utente, un'etichetta solo di riferimento.

`schedule`

La pianificazione su cui creare questa risorsa di calcolo.

`securityGroups`

Il gruppo di sicurezza da utilizzare per le istanze nel pool di risorse.

`name`

Il nome definito dall'utente, un'etichetta solo di riferimento.

`role`

Il ruolo IAM dell'account che accede alle risorse, ad esempio l'accesso a un bucket Amazon S3 per recuperare i dati.

`logUri`

Il percorso di destinazione di Amazon S3 per il backup dei log di Task Runner da. `Ec2Resource`

`resourceRole`

Il ruolo IAM dell'account che crea le risorse, ad esempio la creazione e la configurazione di un'istanza EC2 a tuo nome. Ruolo e ResourceRole 3 possono essere lo stesso ruolo, ma forniscono separatamente una maggiore granularità nella configurazione di sicurezza.

Attività

L'ultima sezione del file JSON è la definizione dell'attività che rappresenta il lavoro da eseguire. In questo caso, utilizziamo un `RedshiftCopyActivity` componente per copiare i dati da Amazon S3 ad Amazon Redshift. Per ulteriori informazioni, consulta RedshiftCopyActivity.

Questo componente `RedshiftCopyActivity` è definito dai campi seguenti:

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

```
},
```

`id`

L'ID definito dall'utente, un'etichetta solo di riferimento.

`input`

Un riferimento al file sorgente di Amazon S3.

`schedule`

La pianificazione su cui eseguire questa attività.

`insertMode`

Il tipo di inserimento (`KEEP_EXISTING`, `OVERWRITE_EXISTING` o `TRUNCATE`).

`name`

Il nome definito dall'utente, un'etichetta solo di riferimento.

`runsOn`

La risorsa di calcolo che esegue il lavoro definito dall'attività.

`output`

Un riferimento alla tabella di destinazione di Amazon Redshift.

Caricamento e attivazione della definizione della pipeline

Devi caricare la definizione della pipeline e attivarla. Nei seguenti comandi di esempio, sostituiteli *`pipeline_name`* con un'etichetta per la pipeline e *`pipeline_file`* con il percorso completo per il file di definizione della pipeline. `.json`

AWS CLI

Per creare la definizione della pipeline e attivare la pipeline, utilizzate il seguente comando create-pipeline. Annota l'ID della pipeline, poiché utilizzerai questo valore con la maggior parte dei comandi CLI.

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

```
}
```

Per caricare la definizione della pipeline, utilizzate il seguente comando. put-pipeline-definition

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

Se la pipeline viene convalidata correttamente, il `validationErrors` campo è vuoto. È necessario esaminare eventuali avvertenze.

Per attivare la pipeline, usa il seguente comando activate-pipeline.

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

È possibile verificare che la pipeline venga visualizzata nell'elenco delle pipeline utilizzando il seguente comando list-pipelines.

```
aws datapipeline list-pipelines
```

Funzioni ed espressioni della pipeline

Questa sezione illustra la sintassi per l'utilizzo di espressioni e funzioni in pipeline, inclusi i tipi di dati associati.

Tipi di dati di esempio

I seguenti tipi di dati possono essere impostati come valori di campi.

Tipi

- DateTime
- Numerico
- Riferimenti agli oggetti
- Periodo
- Stringa

DateTime

AWS Data Pipeline supporta solo la data e l'ora espresse nel formato «YYYY-MM-DDTHH:MM:SS». UTC/GMT L'esempio seguente imposta il campo di un oggetto su, nel fuso orario. `startDateTime` `Schedule` `1/15/2012, 11:59 p.m.` UTC/GMT

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

Numerico

AWS Data Pipeline supporta sia numeri interi che valori a virgola mobile.

Riferimenti agli oggetti

Un oggetto nella definizione di pipeline. Questo può essere l'oggetto corrente, il nome di un oggetto definito altrove nella pipeline o un oggetto che elenca l'oggetto corrente in un campo, a cui si fa riferimento con la parola chiave `node`. Per ulteriori informazioni su `node`, consultare Riferimento a campi e oggetti. Per ulteriori informazioni sui tipi di oggetti della pipeline, consulta Riferimento all'oggetto pipeline.

Periodo

Indica con quale frequenza deve essere eseguito un evento programmato. È espresso nel formato "N [`years`|`months`|`weeks`|`days`|`hours`|`minutes`]", dove N è un valore intero positivo.

La durata minima è pari a 15 minuti, mentre la durata massima è di 3 anni.

L'esempio seguente imposta il campo `period` di un oggetto `Schedule` su 3 ore. In questo modo si crea una pianificazione che viene eseguita ogni tre ore.

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

Stringa

Valori della stringa standard. Le stringhe devono essere racchiusi tra doppie virgolette ("). È possibile utilizzare la barra rovesciata (\) per ignorare i caratteri in una stringa. Le stringhe a più righe non sono supportate.

I seguenti esempi mostrano esempi di valori di stringhe validi per il campo `id`.

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

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

Le stringhe possono anche contenere espressioni che valutano i valori della stringa. Questi vengono inseriti nella stringa e sono delimitati con:"#{" e "}". L'esempio seguente utilizza un'espressione per inserire il nome dell'oggetto corrente in un percorso.

```
"filePath" : "s3://amzn-s3-demo-bucket/#{name}.csv"
```

Per ulteriori informazioni sull'utilizzo delle espressioni, vedi Riferimento a campi e oggetti e Valutazione delle espressioni.

Espressioni

Le espressioni consentono di condividere un valore negli oggetti correlati. Le espressioni vengono elaborate dal servizio AWS Data Pipeline Web in fase di esecuzione, assicurando che tutte le espressioni vengano sostituite con il valore dell'espressione.

Le espressioni sono delimitate da:"#{" e "}". È possibile utilizzare un'espressione in qualsiasi oggetto di definizione della pipeline in cui una stringa è legale. Se uno slot è un riferimento o uno di tipo ID, NAME, TYPE e SPHERE, il valore non viene valutato e viene utilizzato integralmente.

L'espressione seguente chiama una delle AWS Data Pipeline funzioni. Per ulteriori informazioni, consulta Valutazione delle espressioni.

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

Riferimento a campi e oggetti

Le espressioni possono utilizzare i campi dell'oggetto corrente in cui l'espressione esistente o i campi di un altro oggetto collegato da un riferimento.

Un formato di slot è composto da un tempo di creazione seguito dal momento della creazione dell'oggetto, ad esempio `@S3BackupLocation_2018-01-31T11:05:33`.

Puoi anche fare riferimento all'ID esatto dello slot specificato nella definizione della pipeline, ad esempio l'ID dello slot della posizione di backup di Amazon S3. Per fare riferimento all'ID dello slot, utilizzare `#{parent.@id}`.

In questo esempio il campo `filePath` si riferisce al campo `id` nello stesso oggetto per creare un nome di file. Il valore di `filePath` restituisce "`s3://amzn-s3-demo-bucket/ExampleDataNode.csv`".

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

Per usare un campo esistente su un altro oggetto collegato da un riferimento, utilizzare la parola chiave `node`. Questa parola chiave è disponibile solo con allarmi e oggetti di precondizione.

Per continuare con l'esempio precedente, un'espressione in un `SnsAlarm` può fare riferimento alla data e all'intervallo di tempo in un `Schedule`, poiché `S3DataNode` si riferisce a entrambi.

Nello specifico, il campo `message` di `FailureNotify` può utilizzare i campi di runtime `@scheduledStartTime` e `@scheduledEndTime` di `ExampleSchedule`, poiché il campo `onFail` di `ExampleDataNode` fa riferimento a `FailureNotify` e il relativo campo `schedule` fa riferimento a `ExampleSchedule`.

```
{
    "id" : "FailureNotify",
    "type" : "SnsAlarm",
    "subject" : "Failed to run pipeline component",
    "message": "Error for interval
 #{node.@scheduledStartTime}..#{node.@scheduledEndTime}.",
    "topicArn":"arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic"
},
```

> **Note**
>
> È possibile creare pipeline con dipendenze, ad esempio le attività nella pipeline che dipendono dal lavoro di altri sistemi o attività. Se la pipeline richiede alcune risorse, aggiungere tali dipendenze alla pipeline utilizzando le precondizioni associate ai nodi di dati e alle attività. In questo modo il debug della pipeline è più semplice e la pipeline stessa è più resiliente. Inoltre, mantenere le dipendenze all'interno di una singola pipeline quando è possibile, perché la risoluzione dei problemi in più pipeline è difficile.

Espressioni nidificate

AWS Data Pipeline consente di annidare valori per creare espressioni più complesse. Ad esempio, per eseguire un calcolo relativo al tempo (sottrarre 30 minuti da `scheduledStartTime`) e utilizzare il risultato in una definizione di pipeline, è possibile usare la seguente espressione in un'attività:

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

e utilizzando il node prefisso se l'espressione fa parte di una precondizione SnsAlarm or:

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

Elenchi

Le espressioni possono essere valutate su elenchi e su funzioni degli elenchi. Ad esempio, supponiamo che un elenco viene definito come il seguente: `"myList":["one","two"]`. Se l'elenco viene utilizzato nell'espressione `#{'this is ' + myList}`, restituirà `["this is one", "this is two"]`. Se si dispone di due elenchi, Data Pipeline finirà per appiattirli nella valutazione. Ad esempio, se `myList1` è definito come `[1,2]` e `myList2` è definito come `[3,4]`, allora l'espressione `[#{myList1}, #{myList2}]` restituirà `[1,2,3,4]`.

Espressione del nodo

AWS Data Pipeline utilizza l'`#{node.*}`espressione in una delle due `SnsAlarm` o `PreCondition` come riferimento all'oggetto principale di un componente della pipeline. Poiché `SnsAlarm` e `PreCondition` sono citati da un'attività o una risorsa senza alcun riferimento da parte loro, `node` fornisce il modo per consultare il referrer. Ad esempio, la seguente definizione di pipeline dimostra come una notifica di errore può utilizzare `node` per effettuare un riferimento al proprio padre, in questo caso `ShellCommandActivity`, e includere i tempi di inizio e di fine programmati nel messaggio `SnsAlarm`. Il scheduledStartTime riferimento on non ShellCommandActivity richiede il `node` prefisso perché scheduledStartTime si riferisce a se stesso.

> **Note**
>
> I campi preceduti dal segno AT (@) sono campi di runtime.

```
{
  "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 supporta riferimenti transitivi per i campi definiti dall'utente, ma non per i campi di runtime. Un riferimento transitivo è un riferimento tra due componenti di pipeline che dipende da un altro componente di pipeline come intermediario. L'esempio seguente mostra un riferimento a un campo transitivo definito dall'utente e un riferimento a un campo di runtime non transitivo, entrambi validi. Per ulteriori informazioni, consulta Campi definiti dall'utente.

```
{
  "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"
}
```

Valutazione delle espressioni

AWS Data Pipeline fornisce una serie di funzioni che è possibile utilizzare per calcolare il valore di un campo. L'esempio seguente utilizza la funzione `makeDate` per impostare il campo `startDateTime` di un oggetto `Schedule` su GMT/UTC `"2011-05-24T0:00:00"`.

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

Funzioni matematiche

Le seguenti funzioni sono disponibili per funzionare con i valori numerici.

Funzione	Description
+	Addizione. Ad esempio: `#{1 + 2}` Risultato: `3`
-	Sottrazione. Ad esempio: `#{1 - 2}` Risultato: `-1`
*	Moltiplicazione. Ad esempio: `#{1 * 2}` Risultato: `2`
/	Divisione. Se si dividono due numeri interi, il risultato è troncato. Esempio: `#{1 / 2}`, Risultato: `0` Esempio: `#{1.0 / 2}`, Risultato: `.5`
^	Esponente. Ad esempio: `#{2 ^ 2}` Risultato: `4.0`

Funzioni stringa

Le seguenti funzioni sono disponibili per funzionare con i valori delle stringhe.

Funzione	Description
+	Concatenazione. I valori non di stringa vengono prima convertiti in stringhe. Ad esempio: `#{"hel" + "lo"}` Risultato: `"hello"`

Funzioni di data e ora

Le seguenti funzioni sono disponibili per lavorare con DateTime i valori. Per gli esempi, il valore di `myDateTime` è `May 24, 2011 @ 5:10 pm GMT`.

> **Note**
>
> Il date/time formato di AWS Data Pipeline è Joda Time, che sostituisce le classi di data e ora Java. Per ulteriori informazioni, vedere Joda Time - Class. DateTimeFormat

Funzione	Description
`int day(DateTime myDateTime)`	Ottiene il giorno del DateTime valore come numero intero. Ad esempio: `#{day(myDateTime)}` Risultato: `24`
`int dayOfYear(DateTime myDateTime)`	Ottiene il giorno dell'anno del DateTime valore come numero intero. Ad esempio: `#{dayOfYear(myDateTime)}`

Funzione	Description
	Risultato: 144
DateTime firstOfMonth(DateTime myDateTime)	Crea un DateTime oggetto per l'inizio del mese nel periodo specificato DateTime. Ad esempio: #{firstOfMonth(myDateTime)} Risultato: "2011-05-01T17:10:00z"
String format(DateTime myDateTime,String format)	Crea un oggetto String che è il risultato della conversione di quanto specificato DateTime utilizzando la stringa di formato specificata. Ad esempio: #{format(myDateTime,'YYYY-MM-dd HH:mm:ss z')} Risultato: "2011-05-24T17:10:00 UTC"
int hour(DateTime myDateTime)	Ottiene l'ora del DateTime valore come numero intero. Ad esempio: #{hour(myDateTime)} Risultato: 17

Funzione	Description
`DateTime makeDate(int year,int month,int day)`	Crea un DateTime oggetto, in UTC, con l'anno, il mese e il giorno specificati, a mezzanotte. Ad esempio: `#{makeDate(2011,5,24)}` Risultato: `"2011-05-24T0:00:00z"`
`DateTime makeDateTime(int year,int month,int day,int hour,int minute)`	Crea un DateTime oggetto, in UTC, con l'anno, il mese, il giorno, l'ora e il minuto specificati. Ad esempio: `#{makeDateTime(2011,5,24,14,21)}` Risultato: `"2011-05-24T14:21:00z"`
`DateTime midnight(DateTime myDateTime)`	Crea un DateTime oggetto per la mezzanotte corrente, rispetto a quella specificata. DateTime Per esempio, se `MyDateTime` è `2011-05-25T17:10:00z`, il risultato è come segue. Ad esempio: `#{midnight(myDateTime)}` Risultato: `"2011-05-25T0:00:00z"`

Funzione	Description
`DateTime minusDays(DateTime myDateTime,int daysToSub)`	Crea un DateTime oggetto che è il risultato della sottrazione del numero di giorni specifica to dal valore specificato. DateTime Ad esempio: `#{minusDa ys(myDateTime,1)}` Risultato: `"2011-05- 23T17:10:00z"`
`DateTime minusHours(DateTime myDateTime,int hoursToSub)`	Crea un DateTime oggetto che è il risultato della sottrazione del numero di ore specifica to dal valore specificato. DateTime Ad esempio: `#{minusHo urs(myDateTime,1)}` Risultato: `"2011-05- 24T16:10:00z"`
`DateTime minusMinutes(DateTime myDateTim e,int minutesToSub)`	Crea un DateTime oggetto che è il risultato della sottrazione del numero di minuti specifica to dal valore specificato. DateTime Ad esempio: `#{minusMi nutes(myDateTime,1 )}` Risultato: `"2011-05- 24T17:09:00z"`

Funzione	Description
`DateTime minusMonths(DateTime myDateTime,int monthsToSub)`	Crea un DateTime oggetto che è il risultato della sottrazione del numero di mesi specifica to dal valore specificato. DateTime Ad esempio: `#{minusMo nths(myDateTime,1)}` Risultato: `"2011-04-24T17:10:00z"`
`DateTime minusWeeks(DateTime myDateTime,int weeksToSub)`	Crea un DateTime oggetto che è il risultato della sottrazio ne del numero di settimane specificato dal valore specifica to. DateTime Ad esempio: `#{minusWe eks(myDateTime,1)}` Risultato: `"2011-05-17T17:10:00z"`
`DateTime minusYears(DateTime myDateTime,int yearsToSub)`	Crea un DateTime oggetto che è il risultato della sottrazione del numero di anni specifica to dal valore specificato. DateTime Ad esempio: `#{minusYe ars(myDateTime,1)}` Risultato: `"2010-05-24T17:10:00z"`

Funzione	Description
`int minute(DateTime myDateTime)`	Ottiene il minuto del DateTime valore come numero intero. Ad esempio: `#{minute(myDateTime)}` Risultato: `10`
`int month(DateTime myDateTime)`	Ottiene il mese del DateTime valore come numero intero. Ad esempio: `#{month(myDateTime)}` Risultato: `5`
`DateTime plusDays(DateTime myDateTime,int daysToAdd)`	Crea un DateTime oggetto che è il risultato dell'aggiunta del numero di giorni specificato a quello specificato DateTime. Ad esempio: `#{plusDays(myDateTime,1)}` Risultato: `"2011-05-25T17:10:00z"`
`DateTime plusHours(DateTime myDateTime,int hoursToAdd)`	Crea un DateTime oggetto che è il risultato dell'aggiunta del numero di ore specificato a quello specificato DateTime. Ad esempio: `#{plusHours(myDateTime,1)}` Risultato: `"2011-05-24T18:10:00z"`

Funzione	Description
`DateTime plusMinutes(DateTime myDateTime,int minutesToAdd)`	Crea un DateTime oggetto che è il risultato dell'aggiunta del numero di minuti specificato a quello specificato DateTime. Ad esempio: `#{plusMinutes(myDateTime,1)}` Risultato: `"2011-05-24 17:11:00z"`
`DateTime plusMonths(DateTime myDateTime,int monthsToAdd)`	Crea un DateTime oggetto che è il risultato dell'aggiunta del numero di mesi specificato a quello specificato DateTime. Ad esempio: `#{plusMonths(myDateTime,1)}` Risultato: `"2011-06-24T17:10:00z"`
`DateTime plusWeeks(DateTime myDateTime,int weeksToAdd)`	Crea un DateTime oggetto che è il risultato dell'aggiunta del numero di settimane specificato a quello specificato DateTime. Ad esempio: `#{plusWeeks(myDateTime,1)}` Risultato: `"2011-05-31T17:10:00z"`

Funzione	Description
`DateTime plusYears(DateTime myDateTime,int yearsToAdd)`	Crea un DateTime oggetto che è il risultato dell'aggiunta del numero di anni specificato a quello specificato DateTime. Ad esempio: `#{plusYears(myDateTime,1)}` Risultato: `"2012-05-24T17:10:00z"`
`DateTime sunday(DateTime myDateTime)`	Crea un DateTime oggetto per la domenica precedente, relativo a quello specificato DateTime. Se il valore specificato DateTime è una domenica, il risultato è quello specificato DateTime. Ad esempio: `#{sunday(myDateTime)}` Risultato: `"2011-05-22 17:10:00 UTC"`
`int year(DateTime myDateTime)`	Ottiene l'anno del DateTime valore come numero intero. Ad esempio: `#{year(myDateTime)}` Risultato: `2011`

Funzione	Description
`DateTime yesterday(DateTime myDateTime)`	Crea un DateTime oggetto per il giorno precedente, relativo a quello specifica to DateTime. Il risultato è lo stesso di minusDays(1). Ad esempio: `#{yesterd ay(myDateTime)}` Risultato: `"2011-05- 23T17:10:00z"`

Caratteri speciali

AWS Data Pipeline utilizza determinati caratteri che hanno un significato speciale nelle definizioni delle pipeline, come illustrato nella tabella seguente.

Carattere speciale	Description	Esempi
@	Campo di runtime. Questo carattere è un prefisso del nome del campo per un campo che è disponibile solo quando viene eseguita una pipeline.	@actualStartTime @failureReason @resourceStatus
#	Espressione. Le espressioni sono delimitate da: «# {» e «}» e il contenuto delle parentesi viene valutato da. AWS Data Pipeline Per ulteriori informazi oni, consulta Espressioni.	# {format (myDateTime, 'YYYY-MM-dd hh:mm:ss')} s3://amzn-s3-demo-bucket/#{ id}.csv
*	Campo crittografato. Questo carattere è un prefisso del	*password

Carattere speciale	Description	Esempi
	nome di campo che indica che AWS Data Pipeline deve crittografare il contenuto di questo campo in transito tra la console o la CLI e il servizio. AWS Data Pipeline	

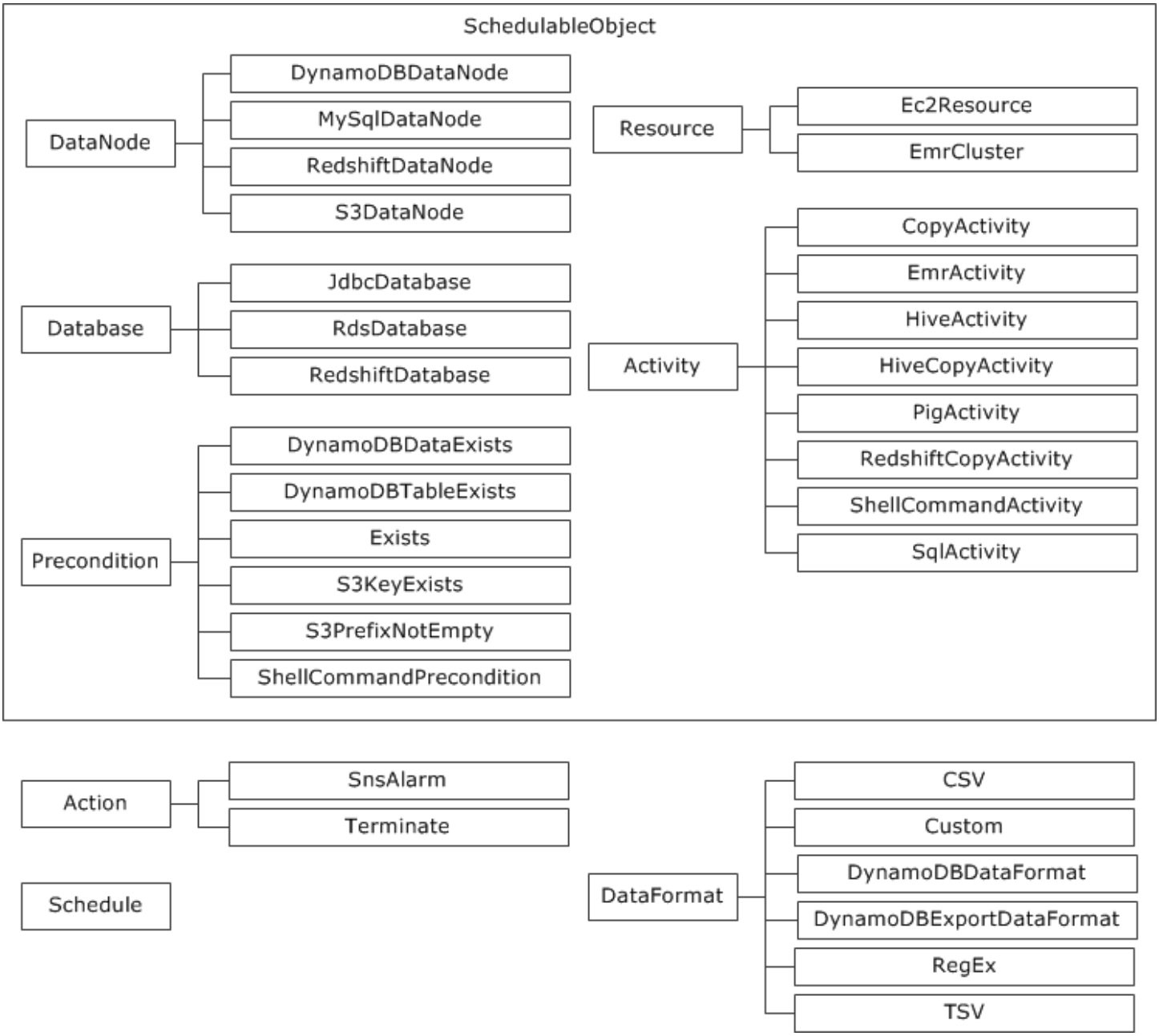
Riferimento all'oggetto pipeline

È possibile utilizzare gli oggetti e i componenti della pipeline nella definizione della pipeline.

Indice

> **Note**
>
> Per un'applicazione di esempio che utilizza AWS Data Pipeline Java SDK, consulta Data Pipeline DynamoDB Export Java Sample on. GitHub

Di seguito è riportata la gerarchia degli oggetti per. AWS Data Pipeline



Nodi di dati

Di seguito sono riportati gli oggetti del nodo AWS Data Pipeline dati:

Oggetti

- DBDataNodo Dynamo
- MySqlDataNode
- RedshiftDataNode

- S3 DataNode
- SqlDataNode

DBDataNodo Dynamo

Definisce un nodo di dati utilizzando DynamoDB, che viene specificato come input per `HiveActivity` un oggetto or. `EMRActivity`

> **Note**
>
> L'oggetto `DynamoDBDataNode` non supporta la precondizione `Exists`.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. Questo oggetto fa riferimento a due altri oggetti definiti nello stesso file di definizione della pipeline. `CopyPeriod` è un oggetto `Schedule` e `Ready` è un oggetto di precondizione.

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

Sintassi

Campi obbligatori	Description	Tipo di slot
tableName	La tabella DynamoDB.	Stringa

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Gli utenti possono soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref": "«}. DefaultSchedule Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta Pianificazione.	Oggetto di riferimento, ad esempio, «schedule»: {"ref»:» myScheduleId «}

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo

Campi opzionali	Description	Tipo di slot
dataFormat	DataFormat per i dati descritti da questo nodo dati. Attualmente supportato per HiveActivity e HiveCopyActivity.	Oggetto di riferimento, «dataFormat»: {"ref» DBDataFormatId :"myDynamo «}
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «} myActivityId
failureAndRerunModo	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}

Campi opzionali	Description	Tipo di slot
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondition»: {"ref»:» «} myPreconditionId
readThroughputPercent	Imposta la percentuale di operazioni di lettura per mantenere il tasso di throughput assegnato di DynamoDB nell'intervallo allocato per la tabella. Il valore è compreso tra 0,1 e 1,0 inclusi.	Double
region	Il codice per la regione in cui esiste la tabella DynamoDB. Ad esempio, us-east-1. Viene utilizzato HiveActivity quando esegue lo staging per le tabelle DynamoDB in Hive.	Enumerazione
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo

Campi opzionali	Description	Tipo di slot
retryDelay	La durata del timeout tra due tentativi.	Periodo
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}
scheduleType	Il tipo di pianificazione consente di specificare se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programmate al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzione successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
writeThroughputPercent	Imposta la percentuale di operazioni di scrittura per mantenere il tasso di throughput assegnato di DynamoDB nell'intervallo allocato per la tabella. Il valore è compreso tra 0,1 e 1,0 inclusi.	Double

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeInstances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa

Campi Runtime	Description	Tipo di slot
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime

Campi Runtime	Description	Tipo di slot
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

MySqlDataNode

Definisce un nodo dati utilizzando MySQL.

> **Note**
>
> Il tipo `MySqlDataNode` è obsoleto. Consigliamo di utilizzare invece SqlDataNode.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. Questo oggetto fa riferimento a due altri oggetti definiti nello stesso file di definizione della pipeline. `CopyPeriod` è un oggetto `Schedule` e `Ready` è un oggetto di precondizione.

```
{
  "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" }
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
table	Nome della tabella nel database MySQL.	Stringa

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla	Oggetto di riferimento, ad esempio

Campi Object Invocation	Description	Tipo di slot
	pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Gli utenti possono soddisfar e questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref": "«}. DefaultSc hedule Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianifica zione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.co m/datapipeline/latest/DeveloperGuide/dp-o bject-schedule.html	«schedule»: {"ref»:» «myScheduleId}

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attivi tà remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
createTableSql	Un'espressione SQL per creare la tabella che crea la tabella.	Stringa

Campi opzionali	Description	Tipo di slot
database	Nome del database.	Oggetto di riferimento, ad esempio «database»: {"ref»:» myDatabaseId «}
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» myActivityId «}
failureAndRerunModo	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
insertQuery	Un'istruzione SQL per inserire dati nella tabella.	Stringa
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}

Campi opzionali	Description	Tipo di slot
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondition»: {"ref»:» «} myPreconditionId
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato, le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programma te al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzion e successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione
schemaName	Il nome dello schema in cui è presente la tabella.	Stringa
selectQuery	Un'istruzione SQL per recuperare i dati dalla tabella.	Stringa
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza	Stringa

Campi Runtime	Description	Tipo di slot
	dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi Runtime	Description	Tipo di slot
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Vedi anche

- S3 DataNode

RedshiftDataNode

Definisce un nodo di dati utilizzando Amazon Redshift. `RedshiftDataNode`rappresenta le proprietà dei dati all'interno di un database, ad esempio una tabella di dati, utilizzata dalla pipeline.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "id" : "MyRedshiftDataNode",
  "type" : "RedshiftDataNode",
```

```
  "database": { "ref": "MyRedshiftDatabase" },
  "tableName": "adEvents",
  "schedule": { "ref": "Hour" }
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
database	Il database in cui risiede la tabella.	Oggetto di riferimento, ad esempio «database»: {"ref»:» myRedshiftDatabase Id "}
tableName	Nome della tabella Amazon Redshift. La tabella viene creata se non esiste già e se l'hai fornita createTableSql.	Stringa

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Gli utenti possono soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref»: "DefaultSchedule«}. Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» «mScheduleId}

Campi Object Invocation	Description	Tipo di slot
	pianificazioni (nidificate all'interno della pianificazione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
createTableSql	Un'espressione SQL per creare la tabella nel database. Si consiglia di specificare lo schema in cui creare la tabella, ad esempio: CREATE TABLE mySchema.myTable (bestColumn varchar (25) primary key distkey, integer sortKey). numberOfWins AWS Data Pipeline esegue lo script nel createTableSql campo se la tabella, specificata da TableName, non esiste nello schema specificato dal campo SchemaName. Ad esempio, se si specifica SchemaName come mySchema ma non si include mySchema nel createTableSql campo, la tabella viene creata nello schema sbagliato	Stringa

Campi opzionali	Description	Tipo di slot
	(per impostazione predefinita, verrebbe creata in PUBLIC). Questo avviene perché AWS Data Pipeline non analizza le istruzioni CREATE TABLE.	
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «} myActivityId
failureAndRerunModo	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}

Campi opzionali	Description	Tipo di slot
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondition»: {"ref»:» «} myPreconditionId
primaryKeys	Se non si specificano le primaryKeys per una tabella di destinazione in `RedShiftCopyActivity` , è possibile specificare un elenco di colonne utilizzando le primaryKeys che agiranno come mergeKey. Tuttavia, se hai una PrimaryKey esistente definita in una tabella Amazon Redshift, questa impostazione sostituisce la chiave esistente.	Stringa
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}
scheduleType	Il tipo di pianificazione consente di specificare se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programmate al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzione successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione
schemaName	Questo campo facoltativo specifica il nome dello schema per la tabella di Amazon Redshift. Se non è specificato, il nome dello schema è PUBLIC, che è lo schema predefinito in Amazon Redshift. Per ulteriori informazioni, consulta la Guida per sviluppatori del database di Amazon Redshift.	Stringa

Campi opzionali	Description	Tipo di slot
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa

Campi Runtime	Description	Tipo di slot
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime

Campi Runtime	Description	Tipo di slot
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

S3 DataNode

Definisce un nodo di dati utilizzando Amazon S3. Per impostazione predefinita, S3 DataNode utilizza la crittografia lato server. Se desideri disabilitarlo, imposta EncryptionType s3 su NONE.

> **Note**
>
> Quando si utilizza un `S3DataNode` come input `CopyActivity`, solo i formati di dati CSV e TSV sono supportati.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. Questo oggetto fa riferimento a un altro oggetto definito nello stesso file di definizione della pipeline. `CopyPeriod` è un oggetto `Schedule`.

```
{
  "id" : "OutputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://amzn-s3-demo-bucket/#{@scheduledStartTime}.csv"
}
```

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Gli utenti possono soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref»: "DefaultSchedule«}. Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianifica	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» «myScheduleId}

Campi Object Invocation	Description	Tipo di slot
	zione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
compressione	Il tipo di compressione per i dati descritto da S3DataNode. «none» non è compressione e «gzip» viene compresso con l'algoritmo gzip. Questo campo è supportato solo per l'uso con Amazon Redshift e quando usi DataNode S3 con. CopyActivity	Enumerazione
dataFormat	DataFormat per i dati descritti da questo S3. DataNode	Oggetto di riferimento, ad esempio «dataFormat»: {"ref»:» myDataFormat Id "}

Campi opzionali	Description	Tipo di slot
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «} myActivityId
directoryPath	Percorso della directory Amazon S3 come URI: s3://my-bucket/. my-key-for-directory È necessario fornire un valore filePath o directoryPath.	Stringa
failureAndRerunMode	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
filePath	Il percorso dell'oggetto in Amazon S3 come URI, ad esempio: s3://my-bucket/. my-key-for-file È necessario fornire un valore filePath o directoryPath. Questi rappresentano una cartella e un nome di file. Utilizzare il valore directoryPath per gestire più file in una directory.	Stringa
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
manifestFilePath	Il percorso di Amazon S3 a un file manifest nel formato supportato da Amazon Redshift. AWS Data Pipeline utilizza il file manifest per copiare i file Amazon S3 specificati nella tabella. Questo campo è valido solo quando un RedShiftCopyActivity fa riferimento a DataNode S3.	Stringa

Campi opzionali	Description	Tipo di slot
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimen to, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimen to, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondi tion»: {"ref»:» «} myPreconditionId

Campi opzionali	Description	Tipo di slot
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimen to, ad esempio «runSon»: {"ref»:» myResourceId «}
s3 EncryptionType	Ignora il tipo di crittografia di Amazon S3. I valori sono SERVER_SIDE_ENCRYPTION o NONE. La crittografia lato server è abilitata per impostazione predefinita.	Enumerazione

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programma te al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzion e successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject

Campi Runtime	Description	Tipo di slot
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa

Campi Runtime	Description	Tipo di slot
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi Runtime	Description	Tipo di slot
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Vedi anche

- MySqlDataNode

SqlDataNode

Definisce un nodo dati utilizzando SQL.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. Questo oggetto fa riferimento a due altri oggetti definiti nello stesso file di definizione della pipeline. `CopyPeriod` è un oggetto `Schedule` e `Ready` è un oggetto di precondizione.

```
{
  "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" }
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
table	Nome della tabella nel database SQL.	Stringa

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Gli utenti possono soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref»: "«}. DefaultSchedule Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» «myScheduleId}

Campi Object Invocation	Description	Tipo di slot
	alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
createTableSql	Un'espressione SQL per creare la tabella che crea la tabella.	Stringa
database	Nome del database.	Oggetto di riferimento, ad esempio «database»: {"ref»:» myDatabaseId «}
dependsOn	Specifica la dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» myActivityId «}

Campi opzionali	Description	Tipo di slot
failureAndRerunModo	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
insertQuery	Un'istruzione SQL per inserire dati nella tabella.	Stringa
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completat o. Viene attivato solo quando il tipo di pianifica zione non è impostato su. `ondemand`	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimen to, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimen to, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi opzionali	Description	Tipo di slot
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondition»: {"ref»:» «} myPreconditionId
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programma te al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzion e successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione
schemaName	Il nome dello schema in cui è presente la tabella.	Stringa
selectQuery	Un'istruzione SQL per recuperare i dati dalla tabella.	Stringa
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza	Stringa

Campi Runtime	Description	Tipo di slot
	dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi Runtime	Description	Tipo di slot
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Vedi anche

- S3 DataNode

Attività

I seguenti sono gli oggetti di attività: AWS Data Pipeline

Oggetti

- CopyActivity
- EmrActivity
- HadoopActivity
- HiveActivity
- HiveCopyActivity

- PigActivity
- RedshiftCopyActivity
- ShellCommandActivity
- SqlActivity

CopyActivity

Copia i dati da una posizione all'altra. `CopyActivity`supporta S3 DataNode e SqlDataNode come input e output e l'operazione di copia viene normalmente eseguita record-by-record. Tuttavia, `CopyActivity` fornisce una copia da Amazon S3 ad Amazon S3 ad alte prestazioni quando sono soddisfatte tutte le seguenti condizioni:

- L'input e l'output sono S3 DataNodes
- Il campo `dataFormat` è lo stesso per input e output

Se si forniscono file di dati compressi come input senza indicarlo utilizzando il campo `compression` sui nodi di dati S3, `CopyActivity` potrebbe non riuscire. In questo caso, `CopyActivity` non è in grado di rilevare correttamente il termine del carattere del record e l'operazione ha esito negativo. Inoltre, `CopyActivity` supporta la copia da una directory a un'altra directory e la copia di un file in una directory, ma la record-by-record copia si verifica quando si copia una directory in un file. Infine, non `CopyActivity` supporta la copia di file Amazon S3 multiparte.

`CopyActivity` ha limitazioni specifiche per il supporto CSV. Quando utilizzi un S3 DataNode come input per`CopyActivity`, puoi utilizzare solo una Unix/Linux variante del formato di file di dati CSV per i campi di input e output di Amazon S3. La Unix/Linux variante richiede quanto segue:

- Il separatore deve essere il carattere "," (virgola).
- I record non sono citati.
- Il carattere di escape predefinito è il valore ASCII 92 (barra rovesciata).
- La fine dell'identificatore del record è il valore ASCII 10 (o "\n").

I sistemi basati su Windows utilizzano in genere una sequenza di end-of-record caratteri diversa: un riage return e una riga di alimentazione insieme (valore ASCII 13 e valore ASCII 10). È necessario adeguarsi a questa differenza utilizzando un ulteriore meccanismo, come uno script di pre-copia per

modificare i dati di input, e assicurare che `CopyActivity` sia in grado di rilevare correttamente la fine di un record; in caso contrario, `CopyActivity` ha ripetutamente esito negativo.

Quando si utilizza `CopyActivity` per eseguire l'esportazione da un oggetto RDS PostgreSQL a un formato di dati TSV, il carattere NULL predefinito è \n.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. Questo oggetto fa riferimento a tre altri oggetti definiti nello stesso file di definizione della pipeline. `CopyPeriod` è un oggetto `Schedule` e `InputData` e `OutputData` sono oggetti di nodi di dati.

```
{
  "id" : "S3ToS3Copy",
  "type" : "CopyActivity",
  "schedule" : { "ref" : "CopyPeriod" },
  "input" : { "ref" : "InputData" },
  "output" : { "ref" : "OutputData" },
  "runsOn" : { "ref" : "MyEc2Resource" }
}
```

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Gli utenti possono soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref": "«}. DefaultSchedule Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione.	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» «myScheduleId}

Campi Object Invocation	Description	Tipo di slot
	O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro	Periodo

Campi opzionali	Description	Tipo di slot
	il tempo impostato di avvio viene tentata di nuovo.	
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «myActivityId}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
input	Origine dati di input.	Oggetto di riferimento, ad esempio «input»: {"ref»:» myDataNode Id "}
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}

Campi opzionali	Description	Tipo di slot
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
output	Origine dati di output.	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNode Id "}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondi tion»: {"ref»:» «} myPreconditionId
reportProgressTime out	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programma te al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzion e successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime

Campi Runtime	Description	Tipo di slot
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime

Campi Runtime	Description	Tipo di slot
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Vedi anche

- ShellCommandActivity
- EmrActivity
- Esportazione di dati MySQL su Amazon S3 utilizzando AWS Data Pipeline

EmrActivity

Esegue un cluster EMR.

AWS Data Pipeline utilizza un formato diverso per i passaggi rispetto ad Amazon EMR; ad esempio, AWS Data Pipeline utilizza argomenti separati da virgole dopo il nome JAR nel campo step. `EmrActivity` L'esempio seguente mostra un passaggio formattato per Amazon EMR, seguito dal AWS Data Pipeline suo equivalente:

```
s3://amzn-s3-demo-bucket/MyWork.jar arg1 arg2 arg3
```

```
"s3://amzn-s3-demo-bucket/MyWork.jar,arg1,arg2,arg3"
```

Esempi

Di seguito è illustrato un esempio di questo tipo di oggetto. Questo esempio utilizza versioni precedenti di Amazon EMR. Verifica la correttezza di questo esempio con la versione del cluster Amazon EMR che stai utilizzando.

Questo oggetto fa riferimento a tre altri oggetti definiti nello stesso file di definizione della pipeline. `MyEmrCluster` è un oggetto `EmrCluster` e `MyS3Input` e `MyS3Output` sono oggetti `S3DataNode`.

> **Note**
>
> In questo esempio, è possibile sostituire il campo `step` con la stringa di cluster desiderato, che può essere uno script Pig, un cluster di streaming Hadoop, un JAR personalizzato, inclusi i parametri, e così via.

Hadoop 2.x (AMI 3.x)

```
{
  "id" : "MyEmrActivity",
  "type" : "EmrActivity",
  "runsOn" : { "ref" : "MyEmrCluster" },
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : ["s3://amzn-s3-demo-bucket/myPath/myStep.jar,firstArg,secondArg,-files,s3://
amzn-s3-demo-bucket/myPath/myFile.py,-input,s3://myinputbucket/path,-output,s3://
myoutputbucket/path,-mapper,myFile.py,-reducer,reducerName","s3://amzn-s3-demo-bucket/
myPath/myotherStep.jar,..."],
  "postStepCommand" : "scp localFiles remoteFiles",
  "input" : { "ref" : "MyS3Input" },
  "output" : { "ref" : "MyS3Output" }
}
```

> **Note**
>
> Per passare argomenti a un'applicazione in una fase, è necessario specificare la regione nel percorso dello script, come nell'esempio seguente. Inoltre, potrebbe essere necessario uscire dagli argomenti passati. Ad esempio, se si utilizza `script-runner.jar` per eseguire uno script shell e si desidera passare argomenti allo script, è necessario eliminare le virgole che li separano. Lo slot della fase che segue dimostra come eseguire questa operazione:
>
> ```
> "step" : "s3://eu-west-1.elasticmapreduce/libs/script-runner/script-
> runner.jar,s3://datapipeline/echo.sh,a\\\\,b\\\\,c"
> ```
>
> In questa fase viene utilizzato `script-runner.jar` per eseguire lo `echo.sh` script shell e passare `a`, `b` e `c` come unico argomento allo script. Il primo carattere escape viene rimosso

dall'argomento risultante in modo da poterlo utilizzare di nuovo. Ad esempio, se si ha `File\.gz` come argomento in formato JSON, è possibile farlo usando `File\\\\.gz`. Tuttavia, poiché il primo carattere escape viene eliminato, è necessario utilizzare `File\\\\\\\\.gz` .

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione e delle dipendenze per questo oggetto. È possibile soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio, specificando `"schedule": {"ref": "DefaultSchedule"}` . Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), è possibile creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» myScheduleId «}

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	Il cluster Amazon EMR su cui verrà eseguito questo processo.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myEmrCluster Id "}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore `runs0n` ed esiste `workerGroup` , `workerGroup` verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» myActivityId «}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione

Campi opzionali	Description	Tipo di slot
input	Posizione dei dati di input.	Oggetto di riferimento, ad esempio, «input»: {"ref»:» myDataNode Id "}
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimen to, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimen to, ad esempio «onSuccess»: {"ref»:» myActionId «}
output	Posizione dei dati di output.	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNode Id "}

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
pipelineLogUri	L'URI di Amazon S3, ad esempio 's3://Buc ketName/Prefix/ 'per caricare i log per la pipeline.	Stringa
postStepCommand	Script di shell da eseguire dopo il completam ento di tutti i passaggi. Per specificare più script, fino a 255, aggiungere più campi `postStepCommand` .	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimen to, ad esempio, «precondition»: {"ref»:» «} myPrecond itionId
preStepCommand	Script di shell da eseguire prima dell'esec uzione di qualsiasi passaggio. Per specifica re più script, fino a 255, aggiungere più campi `preStepCommand` .	Stringa
reportProgressTime out	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo

Campi opzionali	Description	Tipo di slot
resizeClusterBeforeIn esecuzione	Ridimensiona il cluster prima di eseguire questa attività per adattarlo alle tabelle DynamoDB specificate come input o output. **Note** Se `EmrActivity` utilizzi a `DynamoDBDataNode` come nodo di dati di input o output e lo imposti su, inizia `resizeClusterBefor eRunning` a `TRUE` utilizzare i tipi di istanza. AWS Data Pipeline `m3.xlarge` Questo sovrascrive le tue scelte in termini di tipi di istanze con `m3.xlarge` , con un possibile aumento dei costi.	Booleano
resizeClusterMaxIs tanze	Un limite per il numero massimo di istanze che possono essere richieste dall'algoritmo di ridimensionamento.	Numero intero
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. I valori sono `cron`, `ondemand` e `timeseries` . La pianificazione `timeserie s` significa che le istanze sono programmate al termine di ogni intervallo. La pianificazione `cron` significa che le istanze sono programma te all'inizio di ogni intervallo. Una pianificazione `ondemand`consente di eseguire una pipeline una sola volta, per attivazione. Non è necessari o clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione `ondemand`, devi specificarlo nell'oggetto predefinito e deve essere l'unico `scheduleType` specificato per gli oggetti della pipeline. Per utilizzare le pipeline `ondemand`, chiama l'operazione `ActivatePipeline` per ogni esecuzione successiva.	Enumerazione
fase	Uno o più passaggi per il cluster da eseguire. Per specificare più passaggi, fino a 255, aggiungere più campi relativi a queste informazi oni. Utilizza argomenti separati da virgole dopo il nome JAR; ad esempio, "`s3://amzn-s3-demo-bucket/MyWork.jar,arg1,arg2 ,arg3 `".	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn

Campi Runtime	Description	Tipo di slot
		stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	I log dei passaggi di Amazon EMR sono disponibili solo nei tentativi di attività EMR	Stringa
errorId	`errorId` se l'oggetto non riuscito.	Stringa
errorMessage	`errorMessage` se l'oggetto non riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa

Campi Runtime	Description	Tipo di slot
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ Ora healthStatusUpdated	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per l'oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi Runtime	Description	Tipo di slot
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Vedi anche

- ShellCommandActivity
- CopyActivity
- EmrCluster

HadoopActivity

Esegue un MapReduce processo su un cluster. Il cluster può essere un cluster EMR gestito da AWS Data Pipeline o un'altra risorsa, se si utilizza. TaskRunner Da utilizzare HadoopActivity quando si desidera eseguire il lavoro in parallelo. Ciò consente di utilizzare le risorse di pianificazione del framework YARN o del negoziatore di MapReduce risorse in Hadoop 1. Se desideri eseguire il lavoro in sequenza utilizzando l'azione Amazon EMR Step, puoi comunque utilizzare. EmrActivity

Esempi

HadoopActivity utilizzando un cluster EMR gestito da AWS Data Pipeline

L' HadoopActivity oggetto seguente utilizza una EmrCluster risorsa per eseguire un programma:

```
 {
   "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://amzn-s3-demo-bucket/MyHadoopActivity/#{@pipelineId}/
#{format(@scheduledStartTime,'YYYY-MM-dd')}"
   ],
   "maximumRetries": "0",
   "postActivityTaskConfig":{"ref":"postTaskScriptConfig”},
   "hadoopQueue" : “high”
 }
```

Ecco il corrispondente*MyEmrCluster*, che configura le code FairScheduler and in YARN per sistemi basati su Hadoop 2: AMIs

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
   "hadoopSchedulerType" : "PARALLEL_FAIR_SCHEDULING",
  “amiVersion” : “3.7.0”,
  "bootstrapAction" : ["s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
\,high\,default,-z,yarn.scheduler.capacity.root.high.capacity=50,-
```

```
z,yarn.scheduler.capacity.root.low.capacity=10,-
z,yarn.scheduler.capacity.root.default.capacity=30”]
}
```

Questo è quello che si usa per configurare in EmrCluster Hadoop 1: FairScheduler

```
{
      "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"
          }
```

Le seguenti EmrCluster configurazioni CapacityScheduler per sistemi basati su Hadoop 2: AMIs

```
{
      "id": "MyEmrCluster",
      "type": "EmrCluster",
      "hadoopSchedulerType": "PARALLEL_CAPACITY_SCHEDULING",
      "amiVersion": "3.7.0",
      "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
\\\\,high,-z,yarn.scheduler.capacity.root.high.capacity=40,-
z,yarn.scheduler.capacity.root.low.capacity=60"
    }
```

HadoopActivity utilizzo di un cluster EMR esistente

In questo esempio, si utilizzano workergroups e TaskRunner a per eseguire un programma su un cluster EMR esistente. La seguente definizione di pipeline consente di: HadoopActivity

- Esegue un MapReduce programma solo sulle *myWorkerGroup* risorse. Per ulteriori informazioni sui gruppi di lavoratori, consulta Esecuzione del lavoro su risorse esistenti utilizzando Task Runner.
- Esegui un preActivityTask Config e Config postActivityTask

```
{
  "objects": [
    {
```

```
      "argument": [
        "-files",
        "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
        "-mapper",
        "wordSplitter.py",
        "-reducer",
        "aggregate",
        "-input",
        "s3://elasticmapreduce/samples/wordcount/input/",
        "-output",
        "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/scripts/preTaskScript.sh",
      "name": "preTaskScriptConfig",
      "scriptArgument": [
        "test",
        "argument"
      ],
      "type": "ShellScriptConfig"
    },
    {
      "id": "ShellScriptConfig",
      "scriptUri": "s3://amzn-s3-demo-bucket/scripts/postTaskScript.sh",
      "name": "postTaskScriptConfig",
      "scriptArgument": [
        "test",
        "argument"
      ],
```

```
      "type": "ShellScriptConfig"
    },
    {
      "id": "Default",
      "scheduleType": "cron",
      "schedule": {
        "ref": "SchedulePeriod"
      },
      "name": "Default",
      "pipelineLogUri": "s3://amzn-s3-demo-bucket/
logs/2015-05-22T18:02:00.343Z642f3fe415",
      "maximumRetries": "0",
      "workerGroup": "myWorkerGroup",
      "preActivityTaskConfig": {
        "ref": "preTaskScriptConfig"
      },
      "postActivityTaskConfig": {
        "ref": "postTaskScriptConfig"
      }
    }
  ]
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
jarUri	Posizione di un JAR in Amazon S3 o nel file system locale del cluster con cui eseguire. HadoopActivity	Stringa

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» «mySchedule Id}

Campi Object Invocation	Description	Tipo di slot
	questo oggetto. Gli utenti possono soddisfar e questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref»: "DefaultS chedule«}. Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianifica zione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.co m/datapipeline/latest/DeveloperGuide/dp-o bject-schedule.html	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	Cluster EMR su cui il processo verrà eseguito.	Oggetto di riferimen to, ad esempio «runSon»: {"ref»:» myEmrCluster Id "}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
argument	Gli argomenti da trasmettere al JAR.	Stringa
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «myActivityId}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
hadoopQueue	Il nome della coda del pianificatore Hadoop a cui verrà inviata l'attività.	Stringa
input	Posizione dei dati di input.	Oggetto di riferimento, ad esempio «input»: {"ref»:» myDataNode Id "}
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
mainClass	La classe principale del JAR con HadoopActivity cui stai eseguendo.	Stringa

Campi opzionali	Description	Tipo di slot
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimen to, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimen to, ad esempio «onSuccess»: {"ref»:» myActionId «}
output	Posizione dei dati di output.	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNode Id "}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa

Campi opzionali	Description	Tipo di slot
postActivityTaskConfig	Lo script di configurazione post-attività da eseguire. Questo è composto da un URI dello script della shell in Amazon S3 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "postActivityTaskConfig»: {"ref»:» myShellScriptConfigId «}
preActivityTaskConfig	Lo script di configurazione pre-attività da eseguire. Questo è composto da un URI dello script della shell in Amazon S3 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "preActivityTaskConfig»: {"ref»:» myShellScriptConfigId «}
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondizione»: {"ref»:» «myPreconditionId}
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programma te al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzion e successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime

Campi Runtime	Description	Tipo di slot
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime

Campi Runtime	Description	Tipo di slot
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Vedi anche

- ShellCommandActivity
- CopyActivity
- EmrCluster

HiveActivity

Esegue una query Hive su un cluster EMR. `HiveActivity`semplifica la configurazione di un'attività di Amazon EMR e crea automaticamente tabelle Hive in base ai dati di input provenienti da Amazon S3 o Amazon RDS. Tutto ciò che devi specificare è l'HiveQL da eseguire sui dati di origine. AWS Data Pipeline crea automaticamente tabelle Hive con `${input1}${input2}`, e così via, in base ai campi di input nell'oggetto. `HiveActivity`

Per gli input di Amazon S3, il `dataFormat` campo viene utilizzato per creare i nomi delle colonne Hive.

Per gli input MySQL (Amazon RDS), i nomi delle colonne per la query SQL vengono utilizzati per creare i nomi delle colonne Hive.

> **Note**
>
> Questa attività utilizza il CSV Serde di Hive.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. Questo oggetto fa riferimento a tre altri oggetti definiti nello stesso file di definizione della pipeline. `MySchedule` è un oggetto `Schedule` e `MyS3Input` e `MyS3Output` sono oggetti di nodi di dati.

```
{
  "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" }
}
```

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione e delle dipendenze per questo oggetto. È possibile soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref»: "«}. DefaultSchedule Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), è possibile creare un oggetto padre che dispone di un	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» «myScheduleId}

Campi Object Invocation	Description	Tipo di slot
	riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianifica zione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
hiveScript	Lo script Hive da eseguire.	Stringa
scriptUri	La posizione dello script Hive da eseguire (ad esempio, s3:// scriptLocation).	Stringa

Gruppo obbligatorio	Description	Tipo di slot
runsOn	Il cluster EMR in cui viene eseguita questa `HiveActivity` .	Oggetto di riferimen to, ad esempio «runSon»: {"ref»:» myEmrCluster Id "}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore `runs0n` ed esiste `workerGroup` , `workerGroup` verrà ignorato.	Stringa
input	Origine dati di input.	Oggetto di riferimento, ad esempio «input»: {"ref»:» myDataNode Id "}

Gruppo obbligatorio	Description	Tipo di slot
output	Origine dati di output.	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNode Id "}

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» myActivityId «}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
hadoopQueue	Il nome della coda del pianificatore Hadoop a cui verrà inviato il processo.	Stringa
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo

Campi opzionali	Description	Tipo di slot
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
postActivityTaskConfig	Lo script di configurazione post-attività da eseguire. Questo è composto da un URI dello script della shell in Amazon S3 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "postActivityTaskConfig»: {"ref»:» myShellScriptConfigId «}

Campi opzionali	Description	Tipo di slot
preActivityTaskConfig	Lo script di configurazione pre-attività da eseguire. Questo è composto da un URI dello script della shell in Amazon S3 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "preActivityTaskConfig»: {"ref»:» myShellScriptConfigId «}
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondition»: {"ref»:» myPreconditionId «}
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
resizeClusterBeforeInesecuzione	Ridimensiona il cluster prima di eseguire questa attività per adattare i nodi di dati DynamoDB specificati come input o output. **Note** Se la tua attività utilizza un `DynamoDBDataNode` come nodo di dati di input o output e lo imposti su, inizia `resizeClusterBeforeRunning` a `TRUE` utilizzare i tipi di istanza. AWS Data Pipeline `m3.xlarge` Questo sovrascrive le tue scelte in termini di tipi di istanze con `m3.xlarge` , con un possibile aumento dei costi.	Booleano

Campi opzionali	Description	Tipo di slot
resizeClusterMaxIstanze	Un limite per il numero massimo di istanze che possono essere richieste dall'algoritmo di ridimensionamento.	Numero intero
retryDelay	La durata del timeout tra due tentativi.	Periodo
scheduleType	Il tipo di pianificazione consente di specificare se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programmate al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l'ActivatePipeline operazione per ogni esecuzione successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione

Campi opzionali	Description	Tipo di slot
scriptVariable	Speciifica le variabili di script per Amazon EMR da passare a Hive durante l'esecuzione di uno script. Ad esempio, le seguenti variabili di script di esempio passano le variabili SAMPLE e FILTER_DATE a Hive: `SAMPLE=s3://elasticmapreduce/samples/hive-ads e FILTER_DATE=#{format(@scheduledStartTime,'YYYY-MM-dd')}%` . Questo campo accetta più valori e funziona con entrambi i campi `script` e `scriptUri` . Inoltre, `scriptVariable` funziona indipendentemente dall'impostazione della fase di sviluppo su `true` o `false`. Il campo è particolarmente utile per inviare valori dinamici a Hive utilizzando le espressioni e le funzioni AWS Data Pipeline .	Stringa
fase	Stabilisce se è abilitata la gestione temporane a prima o dopo aver eseguito lo script. Non consentito con Hive 11, quindi usa un'AMI Amazon EMR versione 3.2.0 o successiva.	Booleano

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime

Campi Runtime	Description	Tipo di slot
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	I log dei passaggi di Amazon EMR sono disponibili solo nei tentativi di attività EMR.	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa

Campi Runtime	Description	Tipo di slot
@ Ora healthSta tusUpdated	L'ora in cui lo stato di integrità è stato aggiornat o l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTi me	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCom pletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per un oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimen to, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Vedi anche

- ShellCommandActivity
- EmrActivity

HiveCopyActivity

Esegue una query Hive su un cluster EMR. `HiveCopyActivity`semplifica la copia dei dati tra tabelle DynamoDB. `HiveCopyActivity`accetta un'istruzione HiveQL per filtrare i dati di input da DynamoDB a livello di colonna e riga.

Esempio

L'esempio seguente mostra come usare `HiveCopyActivity` e `DynamoDBExportDataFormat` per copiare i dati da una versione `DynamoDBDataNode` a un'altra, mentre i dati vengono filtrati, in base a un timestamp.

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

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Gli utenti possono soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref": "«}. DefaultSchedule Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» «myScheduleId}

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	Specificare il cluster per l'esecuzione.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore `runs0n` ed esiste `workerGroup` , `workerGroup` verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Il timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
dependsOn	Specifica la dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «myActivityId}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
filterSql	Un frammento di istruzione SQL Hive che filtra un sottoinsieme di dati DynamoDB o	Stringa

Campi opzionali	Description	Tipo di slot
	Amazon S3 da copiare. Il filtro deve contenere solo predicati e non iniziare con una `WHERE` clausola, perché la aggiunge automaticamente. AWS Data Pipeline	
input	Origine dati di input. Questo deve essere `S3DataNode` o `DynamoDBDataNode` . Se utilizzi `DynamoDBNode` , specifica `DynamoDBExportDataFormat` .	Oggetto di riferimento, ad esempio «input»: {"ref»:» myDataNode Id "}
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}

Campi opzionali	Description	Tipo di slot
output	Origine dati di output. Se l'input è `S3DataNode` , questo deve essere `DynamoDBDataNode` . Altrimenti può essere `S3DataNode` o `DynamoDBDataNode` . Se utilizzi `DynamoDBNode` , specifica `DynamoDBExportDataFormat` .	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNode Id "}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
pipelineLogUri	L'URI di Amazon S3, ad esempio, per il `'s3://BucketName/Key/'` caricamento dei log per la pipeline.	Stringa
postActivityTaskConfig	Lo script di configurazione post-attività da eseguire. Questo è composto da un URI dello script della shell in Amazon S3 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "postActivityTaskConfig»: {"ref»:» myShellScript ConfigId «}
preActivityTaskConfig	Lo script di configurazione pre-attività da eseguire. Questo è composto da un URI dello script della shell in Amazon S3 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "preActivityTaskConfig»: {"ref»:» myShellScript ConfigId «}
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondizione»: {"ref»:» «myPreconditionId}

Campi opzionali	Description	Tipo di slot
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
resizeClusterBeforeInesecuzione	Ridimensiona il cluster prima di eseguire questa attività per adattare i nodi di dati DynamoDB specificati come input o output. **Note** Se la tua attività utilizza un `DynamoDBDataNode` come nodo di dati di input o output e lo imposti su, inizia `resizeClusterBeforeRunning` a `TRUE` utilizzare i tipi di istanza. AWS Data Pipeline `m3.xlarge` Questo sovrascrive le tue scelte in termini di tipi di istanze con `m3.xlarge` , con un possibile aumento dei costi.	Booleano
resizeClusterMaxIstanze	Un limite per il numero massimo di istanze che possono essere richieste dall'algoritmo di ridimensionamento	Numero intero
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programma te al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzion e successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime

Campi Runtime	Description	Tipo di slot
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	I log dei passaggi di Amazon EMR sono disponibili solo nei tentativi di attività EMR.	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ Ora healthStatusUpdated	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime

Campi Runtime	Description	Tipo di slot
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Object.	Stringa

Vedi anche

- ShellCommandActivity
- EmrActivity

PigActivity

PigActivity fornisce supporto nativo per gli script Pig AWS Data Pipeline senza la necessità di utilizzare o. `ShellCommandActivity` `EmrActivity` Inoltre, PigActivity supporta l'archiviazione dei dati. Quando il campo della fase è impostato su true, AWS Data Pipeline gestisce temporaneamente i dati di input come schema in Pig senza codice aggiuntivo da parte dell'utente.

Esempio

La seguente pipeline di esempio mostra come utilizzare `PigActivity`. La pipeline di esempio esegue le operazioni seguenti:

- MyPigActivity1 carica i dati da Amazon S3 ed esegue uno script Pig che seleziona alcune colonne di dati e li carica su Amazon S3.
- MyPigActivity2 carica il primo output, seleziona alcune colonne e tre righe di dati e lo carica su Amazon S3 come secondo output.
- MyPigActivity3 carica il secondo dato di output, inserisce due righe di dati e solo la colonna denominata «quinta» in Amazon RDS.
- MyPigActivity4 carica i dati Amazon RDS, seleziona la prima riga di dati e la carica su Amazon S3.

```
{
  "objects": [
    {
      "id": "MyInputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "directoryPath": "s3://amzn-s3-demo-bucket/pigTestInput",
      "name": "MyInputData1",
      "dataFormat": {
        "ref": "MyInputDataType1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyPigActivity4",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyOutputData3"
      },
      "pipelineLogUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-bucket/PigActivityOutput2",
      "dataFormat": {
        "ref": "MyOutputDataType2"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyOutputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "name": "MyOutputData1",
      "directoryPath": "s3://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-bucket/path",
      "scriptUri": "s3://amzn-s3-demo-bucket/script/pigTestScipt.q",
      "name": "MyPigActivity1",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "scriptVariable": [
        "column1=First",
        "column2=Second",
        "three=3"
      ],
      "type": "PigActivity",
      "output": {
        "ref": "MyOutputData1"
      },
      "stage": "true"
    }
  ]
}
```

Il contenuto di `pigTestScript.q` è il seguente.

```
B = LIMIT ${input1} $three; ${output1} = FOREACH B GENERATE $column1, $column2, Third,
 Fourth, Fifth, Sixth, Seventh, Eighth;
```

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Gli utenti devono specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per	Oggetto di riferimento, ad esempio, «schedule»: {"ref»:» mySchedule Id «}

Campi Object Invocation	Description	Tipo di slot
	questo oggetto. Gli utenti possono soddisfar e questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio specificando «schedule»: {"ref»: "«}. DefaultSc hedule Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianifica zione principale), gli utenti possono creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.co m/datapipeline/latest/DeveloperGuide/dp-o bject-schedule.html	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
script	Lo script Pig da eseguire	Stringa
scriptUri	La posizione dello script Pig da eseguire (ad esempio, s3:// scriptLocation).	Stringa

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	Cluster EMR su cui viene eseguito. PigActivity	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myEmrCluster Id "}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore `runs0n` ed esiste `workerGroup` , `workerGroup` verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Il timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
dependsOn	Specifica la dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» myActivityId «}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione

Campi opzionali	Description	Tipo di slot
input	Origine dati di input.	Oggetto di riferimento, ad esempio, «input»: {"ref»:» myDataNode Id "}
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimen to, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimen to, ad esempio «onSuccess»: {"ref»:» myActionId «}
output	Origine dati di output.	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNode Id "}

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
pipelineLogUri	L'URI di Amazon S3 (ad esempio 's3://Buc ketName/Key/ ') per caricare i log per la pipeline.	Stringa
postActivityTaskCo nfig	Lo script di configurazione post-attività da eseguire. È costituito da un URI dello script di shell in Amazon S33 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "postActi vityTaskConfig»: {"ref»:» myShellScript ConfigId «}
preActivityTaskConfig	Lo script di configurazione pre-attività da eseguire. Questo è composto da un URI dello script della shell in Amazon S3 e da un elenco di argomenti.	Oggetto di riferimento, ad esempio "preActiv ityTaskConfig»: {"ref»:» myShellScript ConfigId «}
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimen to, ad esempio, «precondition»: {"ref»:» myPrecond itionId «}
reportProgressTime out	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo

Campi opzionali	Description	Tipo di slot
resizeClusterBeforeInesecuzione	Ridimensiona il cluster prima di eseguire questa attività per adattare i nodi di dati DynamoDB specificati come input o output. **Note** Se la tua attività utilizza un `DynamoDBDataNode` come nodo di dati di input o output e lo imposti su, inizia `resizeClusterBeforeRunning` a `TRUE` utilizzare i tipi di istanza. AWS Data Pipeline `m3.xlarge` Questo sovrascrive le tue scelte in termini di tipi di istanze con `m3.xlarge` , con un possibile aumento dei costi.	Booleano
resizeClusterMaxIstanze	Un limite per il numero massimo di istanze che possono essere richieste dall'algoritmo di ridimensionamento.	Numero intero
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. Time Series Style Scheduling significa che le istanze vengono programma te al termine di ogni intervallo e Cron Style Scheduling significa che le istanze vengono programmate all'inizio di ogni intervallo. Una pianificazione on demand consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico scheduleType specificato per gli oggetti della pipeline. Per utilizzare le pipeline su richiesta, è sufficiente chiamare l' ActivatePipeline operazione per ogni esecuzion e successiva. I valori sono: cron, ondemand e timeseries.	Enumerazione
scriptVariable	Gli argomenti da passare allo script Pig. È possibile utilizzare scriptVariable con lo script o scriptUri.	Stringa
fase	Determina se lo staging è abilitato e consente allo script Pig di accedere alle tabelle di dati staged-data, come $ {} e $ {}INPUT1. OUTPUT1	Booleano

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio, «activeIn

Campi Runtime	Description	Tipo di slot
		stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	I log dei passaggi di Amazon EMR sono disponibili solo nei tentativi di attività EMR.	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa

Campi Runtime	Description	Tipo di slot
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ Ora healthStatusUpdated	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per l'oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi Runtime	Description	Tipo di slot
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Vedi anche

- ShellCommandActivity
- EmrActivity

RedshiftCopyActivity

Copia i dati da DynamoDB o Amazon S3 su Amazon Redshift. È possibile caricare i dati in una nuova tabella, oppure unirli facilmente alla tabella esistente.

Questa è una panoramica di un caso d'uso in cui utilizzare `RedshiftCopyActivity`:

1. Inizia a AWS Data Pipeline utilizzarlo per lo staging dei dati in Amazon S3.
2. `RedshiftCopyActivity`Utilizzalo per spostare i dati da Amazon RDS e Amazon EMR ad Amazon Redshift.

In questo modo puoi caricare i dati in Amazon Redshift dove puoi analizzarli.

3. SqlActivityUtilizzalo per eseguire query SQL sui dati che hai caricato in Amazon Redshift.

Inoltre, `RedshiftCopyActivity` consente di lavorare con un `S3DataNode`, poiché supporta un file manifest. Per ulteriori informazioni, consulta S3 DataNode.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

Per garantire la conversione dei formati, questo esempio utilizza i parametri speciali di conversione EMPTYASNULL e IGNOREBLANKLINES in `commandOptions`. Per informazioni, consulta i parametri di conversione dei dati nella 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"]
}
```

L'esempio seguente di definizione di pipeline mostra un'attività che utilizza la modalità di inserimento `APPEND`:

```
{
  "objects": [
    {
      "id": "CSVId1",
      "name": "DefaultCSV1",
      "type": "CSV"
    },
    {
      "id": "RedshiftDatabaseId1",
      "databaseName": "dbname",
      "username": "user",
      "name": "DefaultRedshiftDatabase1",
      "*password": "password",
```

```
      "type": "RedshiftDatabase",
      "clusterId": "redshiftclusterId"
    },
    {
      "id": "Default",
      "scheduleType": "timeseries",
      "failureAndRerunMode": "CASCADE",
      "name": "Default",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "id": "RedshiftDataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "tableName": "orders",
      "name": "DefaultRedshiftDataNode1",
      "createTableSql": "create table StructuredLogs (requestBeginTime CHAR(30)
PRIMARY KEY DISTKEY SORTKEY, requestEndTime CHAR(30), hostname CHAR(100), requestDate
varchar(20));",
      "type": "RedshiftDataNode",
      "database": {
        "ref": "RedshiftDatabaseId1"
      }
    },
    {
      "id": "Ec2ResourceId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "securityGroups": "MySecurityGroup",
      "name": "DefaultEc2Resource1",
      "role": "DataPipelineDefaultRole",
      "logUri": "s3://myLogs",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "type": "Ec2Resource"
    },
    {
      "id": "ScheduleId1",
      "startDateTime": "yyyy-mm-ddT00:00:00",
      "name": "DefaultSchedule1",
      "type": "Schedule",
      "period": "period",
```

```
      "endDateTime": "yyyy-mm-ddT00:00:00"
    },
    {
      "id": "S3DataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
      "name": "DefaultS3DataNode1",
      "dataFormat": {
        "ref": "CSVId1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "RedshiftCopyActivityId1",
      "input": {
        "ref": "S3DataNodeId1"
      },
      "schedule": {
        "ref": "ScheduleId1"
      },
      "insertMode": "APPEND",
      "name": "DefaultRedshiftCopyActivity1",
      "runsOn": {
        "ref": "Ec2ResourceId1"
      },
      "type": "RedshiftCopyActivity",
      "output": {
        "ref": "RedshiftDataNodeId1"
      }
    }
  ]
}
```

L'operazione APPEND aggiunge gli elementi a una tabella indipendentemente dalle chiavi di ordinamento o primarie. Ad esempio, se si ha la seguente tabella, è possibile aggiungere un record con lo stesso ID e valore utente.

```
ID(PK)     USER
1          aaa
2          bbb
```

È possibile aggiungere un record con lo stesso ID e valore utente:

```
ID(PK)     USER
1          aaa
2          bbb
1          aaa
```

> **Note**
>
> Se un'operazione APPEND viene interrotta e riprovata, la risultante pipeline rieseguita viene potenzialmente aggiunta dall'inizio. L'operazione potrebbe causare un ulteriore doppione, quindi è necessario essere a conoscenza di questo comportamento, soprattutto se si ha una logica che conteggia il numero di righe.

Per un tutorial, vedere Copia i dati su Amazon Redshift utilizzando AWS Data Pipeline.

Sintassi

Campi obbligatori	Description	Tipo di slot
insertMode	Determina AWS Data Pipeline cosa fare con i dati preesistenti nella tabella di destinazione che si sovrappongono alle righe dei dati da caricare. I valori validi sono: KEEP_EXISTING , OVERWRITE_EXISTING , TRUNCATE e APPEND. KEEP_EXISTING aggiunge nuove righe alla tabella, mentre non modifica le righe esistenti. KEEP_EXISTING e OVERWRITE_EXISTING utilizzano la chiave primaria, l'ordinamento e le chiavi di distribuzione per identificare quali righe in entrata abbinare alle righe esistenti. Consulta Aggiornamento e	Enumerazione

Campi obbligatori	Description	Tipo di slot
	inserimento di nuovi dati nella Amazon Redshift Database Developer Guide. TRUNCATE elimina tutti i dati della tabella di destinazione prima di scrivere i nuovi dati. APPEND aggiunge tutti i record alla fine della tabella Redshift. APPEND non richiede una chiave primaria e di distribuzione o la chiave di ordinamento, in modo da poter aggiungere gli elementi che potrebbero essere potenziali duplicati.	

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. Nella maggior parte dei casi, è preferibile inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. Ad esempio, è possibile impostare una pianificazione esplicitamente sull'oggetto, specificando `"schedule": {"ref": "DefaultSchedule"}` . Se la pianificazione master nella pipeline contiene pianificazioni nidificate, è possibile	Reference Object, ad esempio: `"schedule":{"ref":"mySchedule Id"}`

Campi Object Invocation	Description	Tipo di slot
	creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta Pianificazione.	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore `runs0n` ed esiste `workerGroup` , workerGroup verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo

Campi opzionali	Description	Tipo di slot
commandOptions	Richiede i parametri da passare al nodo di dati Amazon Redshift durante l'COPYoperazione. Per informazioni sui parametri, consulta COPY nella Amazon Redshift Database Developer Guide. Mentre carica la tabella, COPY tenta di convertir e in modo implicito le stringhe al tipo di dati della colonna di destinazione. Oltre alle conversioni predefinite dei dati che si verifican o automaticamente, se si ricevono errori o altre esigenze di conversione, è possibile specifica re i parametri di conversione aggiuntivi. Per informazioni, consulta i parametri di conversio ne dei dati nella Amazon Redshift Database Developer Guide. Se un formato di dati è associato al nodo di dati in ingresso o in uscita, allora i parametri forniti vengono ignorati. Poiché l'operazione di copia utilizza per prima cosa COPY per inserire i dati in una tabella intermedia, quindi usa un comando INSERT per copiare i dati dalla tabella intermedia nella tabella di destinazione, alcuni parametri COPY non sono applicabili, ad esempio la capacità del comando COPY di abilitare la compressione automatica della tabella. Se la compressione è obbligatoria, aggiungi i dettagli di codifica della colonna all'istruzione CREATE TABLE. Inoltre, in alcuni casi, quando deve scaricare dati dal cluster Amazon Redshift e creare file in Amazon S3, si affida RedshiftCopyActivi	Stringa

Campi opzionali	Description	Tipo di slot
	`ty` al funzionamento `UNLOAD` di Amazon Redshift. Per migliorare le prestazioni durante la copia e lo scaricamento, specificare il parametro `PARALLEL OFF` del comando `UNLOAD`. Per informazioni sui parametri, consulta UNLOAD nella Amazon Redshift Database Developer Guide.	
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Reference Object: `"dependsOn": {"ref":"myActivityId"}`
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
input	Nodo dei dati di input. L'origine dati può essere Amazon S3, DynamoDB o Amazon Redshift.	Reference Object: `"input":{ "ref":"myDataNodeId"}`
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero

Campi opzionali	Description	Tipo di slot
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Reference Object: `"onFail": {"ref":"myActionId"}`
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Reference Object: `"onLateAction": {"ref":"myActionId"}`
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Reference Object: `"onSuccess": {"ref":"myActionId"}`
output	Nodo dei dati di output. Il percorso di output può essere Amazon S3 o Amazon Redshift.	Reference Object: `"output": {"ref":"myDataNodeId"}`
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Reference Object: `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	L'URI S3 (ad esempio 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Reference Object: `"precondition": {"ref":"myPreconditionId"}`

Campi opzionali	Description	Tipo di slot
coda	Corrisponde all'`query_group` impostazione di Amazon Redshift, che consente di assegnare e dare priorità alle attività simultanee in base alla loro collocazione nelle code. Amazon Redshift limita il numero di connessio ni simultanee a 15. Per ulteriori informazioni, consulta Assigning Queries to Queues nella Amazon RDS Database Developer Guide.	Stringa
reportProgressTime out	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato, le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Consente di specificare se la pianificazione per gli oggetti è nella pipeline. I valori sono `cron`, `ondemand` e `timeseries` . La pianificazione `timeseries` significa che le istanze sono programmate al termine di ogni intervallo. La pianificazione `Cron` significa che le istanze sono programmate all'inizio di ogni intervallo. Una pianificazione `ondemand`consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Per utilizzare le pipeline `ondemand`, chiama l'operazione `ActivatePipeline` per ogni esecuzione successiva. Se utilizzi una pianificazione `ondemand`, devi specificarlo nell'oggetto predefinito e deve essere l'unica `scheduleType` specificata per gli oggetti della pipeline.	Enumerazione

Campi opzionali	Description	Tipo di slot
transformSql	L'espressione `SQL SELECT` utilizzata per trasformare i dati di input. Esegui l'espressione `transformSql` nella tabella denominata `staging`. Quando copi dati da DynamoDB o Amazon S3 AWS Data Pipeline , crea una tabella chiamata «staging» e inizialmente carica i dati al suo interno. I dati di questa tabella vengono utilizzat i per aggiornare la tabella di destinazione. Lo schema di output di `transformSql` deve corrispondere allo schema della tabella di destinazione finale. Se si specifica l'opzione `transformSql` , viene creata una seconda tabella intermedia dall'istruzione SQL specificata. I dati di questa seconda tabella intermedia vengono quindi aggiornati nella tabella di destinazione finale.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Reference Object: `"activeIn stances": {"ref":"m yRunnable ObjectId"}`
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime

Campi Runtime	Description	Tipo di slot
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Reference Object: `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa

Campi Runtime	Description	Tipo di slot
@ Ora healthStatusUpdated	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Reference Object: `"waitingOn": {"ref":"myRunnableObjectId"}`

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto. Indica la propria posizione nel ciclo di vita. Ad esempio, i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

ShellCommandActivity

Consente di eseguire un comando o uno script. È possibile usare `ShellCommandActivity` per eseguire operazioni pianificate di tipo Cron o con serie temporali.

Quando il `stage` campo è impostato su true e utilizzato con un`S3DataNode`, `ShellCommandActivity` supporta il concetto di staging dei dati, il che significa che puoi spostare i dati da Amazon S3 a una posizione di stage, ad esempio Amazon EC2 o il tuo ambiente locale, eseguire operazioni sui dati utilizzando script e `ShellCommandActivity` poi spostarli nuovamente su Amazon S3.

In questo caso, quando il comando shell è connesso a un input `S3DataNode`, gli script shell operano direttamente sui dati utilizzando `${INPUT1_STAGING_DIR}`, `${INPUT2_STAGING_DIR}` e altri campi, con riferimento ai campi di input `ShellCommandActivity`.

Allo stesso modo, l'output del comando shell può essere archiviato in una directory di output per essere inviato automaticamente ad Amazon S3, a cui si fa riferimento da `${OUTPUT1_STAGING_DIR}` e così via. `${OUTPUT2_STAGING_DIR}`

Queste espressioni possono passare come argomenti della riga di comando al comando shell per l'utilizzo in logiche di trasformazione dei dati.

`ShellCommandActivity` restituisce codici e stringhe di errore in stile Linux. Se una `ShellCommandActivity` presenta un errore, l'`error` restituito è un valore diverso da zero.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

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

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di `schedule`. Per impostare l'ordine di esecuzione delle dipendenze per questo oggetto, specificare un riferimento `schedule`a un altro oggetto. Per soddisfare questo requisito, impostare esplicitamente un `schedule` sull'oggetto, ad esempio, specificando `"schedule": {"ref": "DefaultSchedule"}` . Nella maggior parte dei casi, è preferibile inserire il riferimento `schedule` all'oggetto pipeline di default, in modo che tutti gli oggetti ereditano tale pianificazione. Se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), è possibile creare un oggetto padre che dispone di un riferimento alla pianificazione.	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» myScheduleId «}

Campi Object Invocation	Description	Tipo di slot
	Per distribuire il carico, AWS Data Pipeline crea oggetti fisici leggermente prima del previsto, ma li esegue nei tempi previsti. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
command	Il comando da eseguire. Utilizzare $ per fare riferimento ai parametri posizionali e `scriptArgument` per specificare i parametri del comando. Questo valore ed eventuali parametri associati devono funzionare nell'ambiente da cui si sta eseguendo il Task Runner.	Stringa
scriptUri	Un percorso URI di Amazon S3 per un file da scaricare ed eseguire come comando shell. Specifica solo uno `scriptUri` o solo un campo `command`. Se `scriptUri` non è in grado di utilizzare i parametri, utilizzare `command`.	Stringa

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	La risorsa di calcolo per eseguire l'attività o il comando, ad esempio un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio «runSon»: {"ref»:» myResourceId «}
workerGroup	Utilizzato per le attività di routing. Se si fornisce un valore `runs0n` ed esiste `workerGroup` , `workerGroup` verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Il timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «myActivityId}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
input	Posizione dei dati di input.	Oggetto di riferimento, ad esempio «input»:

Campi opzionali	Description	Tipo di slot
		{"ref»:» myDataNode Id "}
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è completo.	Oggetto di riferimen to, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimen to, ad esempio «onSuccess»: {"ref»:» myActionId «}
output	Posizione dei dati di output.	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNode Id "}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi opzionali	Description	Tipo di slot
pipelineLogUri	L'URI di Amazon S3, ad esempio `'s3://BucketName/Key/'` per il caricamento dei log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondition»: {"ref»:» «} myPreconditionId
reportProgressTimeout	Il timeout per chiamate successive a `reportProgress` da parte di attività in remoto. Se impostato, le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Consente di specificare se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. I valori possibili sono: `cron`, `ondemand` e `timeseries` . Se le istanze sono impostate su `timeseries` significa che sono programmate al termine di ogni intervallo. Se le istanze sono impostate su `Cron` significa che sono programmate all'inizio di ogni intervall o. Se sono impostate su `ondemand`, è possibile eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianifica zione `ondemand`, devi specificarlo nell'ogge tto predefinito come l'unico `scheduleType` per gli oggetti della pipeline. Per utilizzare le pipeline `ondemand`, chiama l'operazione `ActivatePipeline` per ogni esecuzione successiva.	Enumerazione

Campi opzionali	Description	Tipo di slot
scriptArgument	Un array di stringhe in formato JSON da passare al comando specificato dal comando. Ad esempio, se il comando è `echo $1 $2`, specificare `scriptArgument` come `"param1", "param2"`. Per più argomenti e parametri, passare `scriptArgument` come segue: `"scriptArgument":" arg1","scriptArgument":"par am1","scriptArgument":"arg2 ","scriptArgument":"param2"` . `scriptArgument` può essere utilizzato solo con `command`; se si utilizza con `scriptUri` viene generato un errore.	Stringa
fase	Stabilisce se è abilitata la gestione temporane a e consente ai comandi shell di accedere alle variabili dei dati gestiti temporaneamente, ad esempio `${INPUT1_STAGING_DIR}` e `${OUTPUT1_STAGING_DIR}` .	Booleano
stderr	Il percorso che riceve messaggi di errore del sistema reindirizzati dal comando. Se utilizzi il `runs0n` campo, deve trattarsi di un percorso Amazon S3 a causa della natura transitoria della risorsa che esegue la tua attività. Tuttavia, se specifichi il campo `workerGroup` , viene autorizzato un percorso file locale.	Stringa
stdout	Il percorso Amazon S3 che riceve l'output reindirizzato dal comando. Se utilizzi il `runs0n` campo, deve trattarsi di un percorso Amazon S3 a causa della natura transitoria della risorsa che esegue la tua attività. Tuttavia, se specifich i il campo `workerGroup` , viene autorizzato un percorso file locale.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	L'elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	`cancellationReason` se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	La descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
emrStepLog	I log dei passaggi di Amazon EMR sono disponibili solo per i tentativi di attività di Amazon EMR.	Stringa
errorId	`errorId` se l'oggetto non riuscito.	Stringa
errorMessage	`errorMessage` se l'oggetto non riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione dell'ogge tto.	DateTime
hadoopJobLog	Registri di lavoro Hadoop disponibili sui tentativi di attività basate su Amazon EMR.	Stringa

Campi Runtime	Description	Tipo di slot
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	L'Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ Ora healthStatusUpdated	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per l'oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato dell'oggetto.	Stringa

Campi Runtime	Description	Tipo di slot
@version	La AWS Data Pipeline versione utilizzata per creare l'oggetto.	Stringa
@waitingOn	La descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive l'oggetto con il formato errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La posizione di un oggetto nel ciclo di vita. I Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Vedi anche

- CopyActivity
- EmrActivity

SqlActivity

Esegue una query SQL (script) su un database.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "id" : "MySqlActivity",
  "type" : "SqlActivity",
  "database" : { "ref": "MyDatabaseID" },
  "script" : "SQLQuery" | "scriptUri" : s3://scriptBucket/query.sql,
  "schedule" : { "ref": "MyScheduleID" },
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
database	Il database su cui eseguire lo script SQL fornito.	Oggetto di riferimento, ad esempio «database»: {"ref»:» myDatabaseId «}

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. È necessario specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione delle dipendenze per questo oggetto. È possibile impostare una pianificazione esplicitamente sull'oggetto, ad esempio, specificando `"schedule": {"ref": "DefaultSchedule"}` . Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione.	Oggetto di riferimento, ad esempio «schedule»: {"ref»:» mySchedule Id «}

Campi Object Invocation	Description	Tipo di slot
	Se la pipeline dispone di una struttura di pianificazioni nidificate all'interno della pianificazione principale, è possibile creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
script	Lo script SQL da eseguire. È necessario specificare lo script o lo scriptUri. Quando lo script viene archiviato in Amazon S3, lo script non viene valutato come espressione. Specificare più valori per ScriptArgument è utile quando lo script è archiviato in Amazon S3.	Stringa
scriptUri	Un URI che specifica il percorso di uno script SQL da eseguire in questa attività.	Stringa

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
runsOn	Le risorse di calcolo per eseguire l'attività o il comando. Ad esempio, un'istanza Amazon EC2 o un cluster Amazon EMR.	Oggetto di riferimento, ad esempio

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
		«runSon»: {"ref»:» myResourceId «}
workerGroup	Il gruppo di lavoro. Utilizzato per le attività di routing. Se si fornisce un valore `runs0n` ed esiste `workerGroup` , `workerGroup` verrà ignorato.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
dependsOn	Specifica una dipendenza su un altro oggetto eseguibile.	Oggetto di riferimento, ad esempio «dependsOn»: {"ref»:» «myActivityId}
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
input	Posizione dei dati di input.	Oggetto di riferimento, ad esempio «input»: {"ref»:» myDataNode Id "}

Campi opzionali	Description	Tipo di slot
lateAfterTimeout	Il periodo di tempo dall'inizio programmato della pipeline all'interno del quale deve essere avviata l'esecuzione dell'oggetto.	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è ancora stato pianificato o non è ancora stato completato nel periodo di tempo trascorso dall'inizio programmato della pipeline, come specificato da ". lateAfterTimeout	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
output	Posizione dei dati di output. Questo è utile solo per fare riferimento dall'interno di uno script (ad esempio`#{output.tablename}`) e per creare la tabella di output impostando 'createTableSql' nel nodo dati di output. L'output della query SQL non è scritto nel nodo dei dati di output.	Oggetto di riferimento, ad esempio «output»: {"ref»:» myDataNodeId "}

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
pipelineLogUri	L'URI S3 (come 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
precondizione	Definisce eventualmente una precondizione. Un nodo dati non è contrassegnato come "READY" finché tutte le precondizioni non siano state soddisfatte.	Oggetto di riferimento, ad esempio «precondi tion»: {"ref»:» «} myPreconditionId
coda	[solo Amazon Redshift] Corrisponde all'impos tazione query_group in Amazon Redshift che consente di assegnare e stabilire le priorità di attività simultanee in base al loro posiziona mento nelle code. Amazon Redshift limita il numero di connessioni simultanee a 15. Per ulteriori informazioni, consulta Assegnazione di query alle code nella Guida per gli sviluppatori di database Amazon Redshift.	Stringa
reportProgressTime out	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. I valori sono `cron`, `ondemand` e `timeseries` . La pianificazione `timeseries` significa che le istanze sono programmate al termine di ogni intervallo. La pianificazione `cron` significa che le istanze sono programmate all'inizio di ogni intervallo. Una pianificazione `ondemand`consente di eseguire una pipeline una sola volta, per attivazione. Questo significa che non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianifica zione `ondemand`, devi specificarlo nell'oggetto predefinito e deve essere l'unico `scheduleT ype` specificato per gli oggetti della pipeline. Per utilizzare le pipeline `ondemand`, chiama l'operazione `ActivatePipeline` per ogni esecuzione successiva.	Enumerazione
scriptArgument	Un elenco di variabili per lo script. In alternati va, è possibile inserire le espressioni direttame nte nel campo dello script. Per scriptArg ument sono utili più valori quando lo script viene memorizzato in Amazon S3. Esempio: # {format (@scheduledStartTime, "YY-MM-DD HH:MM:SS"}\n# {format (PlusPeriod (@schedul edStartTime, «1 day»), "HH:MM:SS"} YY-MM-DD	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza	Stringa

Campi Runtime	Description	Tipo di slot
	dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ healthStatusUpdated Ora	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi Runtime	Description	Tipo di slot
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Resources

I seguenti sono gli oggetti risorsa: AWS Data Pipeline

Oggetti

- Ec2Resource
- EmrCluster
- HttpProxy

Ec2Resource

Un'istanza Amazon EC2 che esegue il lavoro definito da un'attività di pipeline.

AWS Data Pipeline ora supporta IMDSv2 per l'istanza Amazon EC2, che utilizza un metodo orientato alla sessione per gestire meglio l'autenticazione durante il recupero delle informazioni sui metadati

dalle istanze. Una sessione inizia e termina una serie di richieste che il software in esecuzione su un'istanza Amazon EC2 utilizza per accedere ai metadati e alle credenziali dell'istanza Amazon EC2 archiviati localmente. Il software avvia una sessione con una semplice richiesta HTTP PUT a IMDSv2. IMDSv2 restituisce un token segreto al software in esecuzione sull'istanza Amazon EC2, che utilizzerà il token come password per effettuare IMDSv2 richieste di metadati e credenziali.

> **Note**
>
> Per utilizzare IMDSv2 per la tua istanza Amazon EC2, dovrai modificare le impostazioni, poiché l'AMI predefinita non è compatibile con. IMDSv2 Puoi specificare una nuova versione AMI che puoi recuperare tramite il seguente parametro SSM:. `/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs`

Per informazioni sulle istanze Amazon EC2 predefinite che vengono AWS Data Pipeline create se non specifichi un'istanza, consulta. Istanze Amazon EC2 predefinite per regione AWS

Esempi

EC2-Classic

> **Important**
>
> Solo AWS gli account creati prima del 4 dicembre 2013 supportano la piattaforma EC2-Classic. Se disponi di uno di questi account, potresti avere la possibilità di creare oggetti EC2Resource per una pipeline in una rete EC2-Classic anziché in un VPC. Ti consigliamo vivamente di creare risorse per tutte le tue pipeline in VPC. Inoltre, se disponi di risorse esistenti in EC2-Classic, ti consigliamo di migrarle su un VPC.

L'oggetto di esempio seguente avvia un'istanza EC2 in EC2-Classic, con alcuni campi opzionali impostati.

```
{
  "id" : "MyEC2Resource",
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
```

```
  "instanceType" : "m5.large",
  "securityGroups" : [
    "test-group",
    "default"
  ],
  "keyPair" : "my-key-pair"
}
```

EC2-VPC

L'oggetto di esempio seguente avvia un'istanza EC2 in un VPC non di default, con alcuni campi opzionali impostati.

```
{
  "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"
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
resourceRole	Il ruolo IAM che controlla le risorse a cui può accedere l'istanza Amazon EC2.	Stringa
role	Il ruolo IAM AWS Data Pipeline utilizzato per creare l'istanza EC2.	Stringa

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Per impostare l'ordine di esecuzione delle dipendenze per questo oggetto, specificare un riferimento di pianificazione a un altro oggetto. Questa operazione può essere eseguita in uno dei seguenti modi: • Per garantire che tutti gli oggetti nella pipeline possano ereditare la pianificazione, impostare una pianificazione sull'oggetto esplicitamente: `"schedule": {"ref": "DefaultSchedule"}` . Nella maggior parte dei casi, è utile inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. • Se la pipeline dispone di pianificazioni nidificate all'interno della pianificazione principale, è possibile creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Oggetto di riferimento, ad esempio `"schedule":{"ref": "mySchedule Id"}`

Campi opzionali	Description	Tipo di slot
actionOnResourceFallimento	L'operazione intrapresa dopo il fallimento di una risorsa per questa risorsa. I valori validi sono `"retryall"` e `"retrynone"` .	Stringa
actionOnTaskFallimento	L'operazione intrapresa dopo il fallimento di un'attività per questa risorsa. I valori validi sono `"continue"` e `"terminate"` .	Stringa
associatePublicIpIndirizzo	Indica se assegnare automaticamente un indirizzo IP pubblico all'istanza. Se l'istanza è in Amazon EC2 o Amazon VPC, il valore predefinito è. `true` In caso contrario, il valore predefinito è `false`.	Booleano
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
availabilityZone	La zona di disponibilità in cui avviare l'istanza Amazon EC2.	Stringa
disabilitare IMDSv1	Il valore predefinito è false e abilita entrambi IMDSv1 e IMDSv2. Se lo imposti su true, disabilita IMDSv1 e fornisce solo IMDSv2s	Booleano
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
httpProxy	L'host proxy utilizzato dai client per connettersi ai AWS servizi.	Oggetto di riferimento, ad esempio `"httpProxy":`

Campi opzionali	Description	Tipo di slot
		`{"ref":"myHttpProxyId"}`
imageId	L'ID dell'AMI utilizzato per l'istanza. Per impostazione predefinita, AWS Data Pipeline utilizza il tipo di virtualizzazione AMI HVM. L'AMI specifica IDs utilizzata si basa su una regione. È possibile sovrascrivere l'AMI predefinita specificando l'AMI HVM di tua scelta. Per ulteriori informazioni sui tipi di AMI, consulta Tipi di virtualizzazione dell'AMI Linux e Ricerca di un AMI Linux nella Guida per l'utente di Amazon EC2.	Stringa
initTimeout	Il tempo di attesa prima dell'avvio della risorsa.	Periodo
instanceCount	Obsoleta.	Numero intero
instanceType	Il tipo di istanza Amazon EC2 da avviare.	Stringa
keyPair	Nome della coppia di chiavi. Se avvii un'istanza Amazon EC2 senza specificare una coppia di key pair, non puoi accedervi.	Stringa
lateAfterTimeout	Il tempo trascorso dall'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
minInstanceCount	Obsoleta.	Numero intero

Campi opzionali	Description	Tipo di slot
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio `"onFail": {"ref":"myActionId"}`
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o se è ancora in esecuzione.	Oggetto di riferimento, ad esempio `"onLateAction": {"ref":"myActionId"}`
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio `"onSuccess": {"ref":"myActionId"}`
parent	Padre dell'oggetto corrente da cui vengono ereditati gli slot.	Oggetto di riferimento, ad esempio `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	L'URI di Amazon S3 (ad esempio `'s3://BucketName/Key/'`) per il caricamento dei log per la pipeline.	Stringa
region	Il codice per la regione in cui deve essere eseguita l'istanza Amazon EC2. Per impostazione predefinita, l'istanza viene eseguita nella stessa regione della pipeline. È possibile eseguire l'istanza nella stessa regione del set di dati dipendenti.	Enumerazione

Campi opzionali	Description	Tipo di slot
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate ed essere quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo
runAsUser	L'utente che deve eseguire il. TaskRunner	Stringa
runsOn	Campo non consentito su questo oggetto.	Oggetto di riferimento, ad esempio, `"runsOn": {"ref":"myResourceId"}`

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo oppure on demand. I valori sono: • `timeseries` . Le istanze sono programma te alla fine di ogni intervallo. • `cron`. Le istanze sono programmate all'inizio di ogni intervallo. • `ondemand`. Consente di eseguire una pipeline una volta per attivazione. Non è necessario clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione on demand, devi specificarlo nell'oggetto predefinito e deve essere l'unico `scheduleType` specificato per gli oggetti della pipeline. Per utilizzare le pipeline on demand, chiama l'operazione `ActivateP ipeline` per ogni esecuzione successiva.	Enumerazione
securityGroupIds	Gli ID di uno o più gruppi di sicurezza Amazon EC2 da utilizzare per le istanze nel pool di risorse.	Stringa
securityGroups	Uno o più gruppi di sicurezza Amazon EC2 da utilizzare per le istanze nel pool di risorse.	Stringa
spotBidPrice	L'importo massimo per ora per la tua istanza Spot in dollari, un valore decimale compreso tra 0 e 20,00, esclusi.	Stringa
subnetId	L'ID della sottorete Amazon EC2 in cui avviare l'istanza.	Stringa

Campi opzionali	Description	Tipo di slot
terminateAfter	Il numero di ore dopo cui terminare la risorsa.	Periodo
useOnDemandOnLastAttempt	Nell'ultimo tentativo di richiesta di una risorsa Spot, effettuare una richiesta per istanze on demand invece che per istanze Spot. In questo modo, se tutti i tentativi precedenti non sono andati a buon fine, l'ultimo tentativo non viene interrotto.	Booleano
workerGroup	Campo non consentito su questo oggetto.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio, `"activeInstances": {"ref":"myRunnableObjectId"}`
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	`cancellationReason` se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio, `"cascadeFailedOn": {"ref":"m`

Campi Runtime	Description	Tipo di slot
		`yRunnable ObjectId"}`
emrStepLog	I log dei passaggi sono disponibili solo per i tentativi di attività di Amazon EMR.	Stringa
errorId	ID dell'errore se l'oggetto non è riuscito.	Stringa
errorMessage	Messaggio di errore se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@failureReason	Il motivo dell'errore della risorsa.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	I log dei lavori Hadoop sono disponibili sui tentativi di attività di Amazon EMR.	Stringa
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ Ora healthStatusUpdated	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime

Campi Runtime	Description	Tipo di slot
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per l'oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	La descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio `"waitingOn": {"ref":"myRunnableObjectId"}`

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa

Campi di sistema	Description	Tipo di slot
@sphere	La posizione di un oggetto nel ciclo di vita. I Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

EmrCluster

Rappresenta la configurazione di un cluster Amazon EMR. Questo oggetto viene utilizzato da EmrActivity e HadoopActivity per avviare un cluster.

Indice

Pianificatori

I pianificatori forniscono un modo per specificare l'allocazione di risorse e la prioritizzazione dei processi all'interno di un cluster Hadoop. Gli amministratori o gli utenti possono scegliere un pianificatore per diverse classi di utenti e applicazioni. Un pianificatore può usare le code per allocare risorse per utenti e applicazioni. È possibile configurare tali code quando si crea il cluster. È quindi possibile configurare priorità per determinati tipi di lavoro e utenti rispetto ad altri. In questo modo si ottiene un utilizzo efficiente delle risorse del cluster, consentendo a più utenti di inviare dati al cluster. Esistono tre tipi di pianificatori disponibili:

- FairScheduler— Tenta di pianificare le risorse in modo uniforme per un periodo di tempo significativo.
- CapacityScheduler— Utilizza le code per consentire agli amministratori del cluster di assegnare gli utenti a code con priorità e allocazione delle risorse diverse.
- Predefinito: utilizzato dal cluster, che può essere configurato dal sito.

Versioni di rilascio di Amazon EMR

Un rilascio di Amazon EMR è un insieme di applicazioni open source dell'ecosistema di big data. Ogni versione comprende diverse applicazioni, componenti e funzionalità per i big data che scegli di installare e configurare Amazon EMR quando crei un cluster. È possibile specificare la versione di rilascio con l'etichetta del rilascio. Le etichette di rilascio sono sotto forma di `emr-x.x.x` Ad esempio, `emr-5.30.0`. I cluster Amazon EMR si basano sull'etichetta di rilascio `emr-4.0.0` e successivamente utilizzano la `releaseLabel` proprietà per specificare l'etichetta di rilascio di un oggetto. `EmrCluster` Le versioni precedenti utilizzano la proprietà `amiVersion`.

> **Important**
>
> Tutti i cluster Amazon EMR creati utilizzando la versione 5.22.0 o successiva utilizzano Signature Version 4 per autenticare le richieste verso Amazon S3. Alcune versioni di rilascio precedenti utilizzano Signature Version 2. Il supporto di Signature Version 2 è stato interrotto. Per ulteriori informazioni, consulta Amazon S3 Update – SigV2 Deprecation Period Extended and Modified. Ti consigliamo vivamente di utilizzare una versione di Amazon EMR che supporti la versione 4 di Signature. Per i rilasci di versioni precedenti, a partire da EMR 4.7.x, la versione più recente della serie è stata aggiornata per supportare Signature Version 4. Quando si utilizza una versione precedente del rilascio EMR, si consiglia di utilizzare la versione più recente della serie. Inoltre, evitare versioni precedenti a EMR 4.7.0.

Considerazioni e limitazioni

Usa la versione più recente di Task Runner

Se si utilizza un `EmrCluster` oggetto autogestito con un'etichetta di rilascio, utilizzare la versione più recente di Task Runner. Per ulteriori informazioni sui Task Runner, consulta Lavorare con Task Runner. Puoi configurare i valori delle proprietà per tutte le classificazioni di configurazione di Amazon EMR. Per ulteriori informazioni, consulta Configurazione delle applicazioni nella Amazon EMR Release Guide, the section called “Proprietà” e riferimenti the section called “EmrConfiguration” agli oggetti.

Support per IMDSv2

In precedenza, AWS Data Pipeline supportata solo IMDSv1. Ora AWS Data Pipeline supporta IMDSv2 Amazon EMR 5.23.1, 5.27.1 e 5.32 o versioni successive e Amazon EMR 6.2 o versioni

successive. IMDSv2 utilizza un metodo orientato alla sessione per gestire meglio l'autenticazione durante il recupero delle informazioni sui metadati dalle istanze. È necessario configurare le istanze per effettuare IMDSv2 chiamate creando risorse gestite dall'utente utilizzando -2.0. TaskRunner

Amazon EMR 5.32 o versione successiva e Amazon EMR 6.x

Le serie di release di Amazon EMR 5.32 o successive e 6.x utilizzano la versione 3.x di Hadoop, che ha introdotto importanti cambiamenti nel modo in cui viene valutato il classpath di Hadoop rispetto alla versione 2.x di Hadoop. Le librerie comuni come Joda-Time sono state rimosse dal classpath.

Se EmrActivity o HadoopActivity esegue un file Jar che ha dipendenze da una libreria rimossa in Hadoop 3.x, il passaggio ha esito negativo e restituisce l'errore o. `java.lang.NoClassDefFoundError` `java.lang.ClassNotFoundException` Questo può accadere per i file Jar eseguiti senza problemi utilizzando le versioni di release di Amazon EMR 5.x.

Per risolvere il problema, devi copiare le dipendenze dei file Jar nel classpath di Hadoop su un `EmrCluster` oggetto prima di avviare o il. `EmrActivity` `HadoopActivity` Forniamo uno script bash per farlo. Lo script bash è disponibile nella seguente posizione, dove si trova *`MyRegion`* la AWS regione in cui viene eseguito l'`EmrCluster`oggetto, ad esempio. `us-west-2`

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

Il modo di eseguire lo script dipende dal fatto che venga `EmrActivity` `HadoopActivity` eseguito su una risorsa gestita da AWS Data Pipeline o su una risorsa autogestita.

Se utilizzi una risorsa gestita da AWS Data Pipeline, aggiungi un `bootstrapAction` all'`EmrCluster`oggetto. `bootstrapAction`specifica lo script e i file Jar da copiare come argomenti. È possibile aggiungere fino a 255 `bootstrapAction` campi per `EmrCluster` oggetto e aggiungere un `bootstrapAction` campo a un `EmrCluster` oggetto che dispone già di azioni bootstrap.

Per specificare questo script come azione di bootstrap, usa la seguente sintassi, dove si trova la regione in cui `JarFileRegion` viene salvato il file Jar e ognuna *`MyJarFilen`* è il percorso assoluto in Amazon S3 di un file Jar da copiare nel classpath Hadoop. Non specificate i file Jar che si trovano nel classpath Hadoop per impostazione predefinita.

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh,JarFileRegion,MyJarFile1,MyJarFile2[, ...]
```

L'esempio seguente specifica un'azione di bootstrap che copia due file Jar in Amazon S3`my-jar-file.jar`: e il. `emr-dynamodb-tool-4.14.0-jar-with-dependencies.jar` La regione utilizzata nell'esempio è 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"]
}
```

È necessario salvare e attivare la pipeline affinché la modifica `bootstrapAction` alla nuova abbia effetto.

Se si utilizza una risorsa autogestita, è possibile scaricare lo script nell'istanza del cluster ed eseguirlo dalla riga di comando utilizzando SSH. Lo script crea una directory denominata `/etc/hadoop/conf/shellprofile.d` e un file denominato `datapipeline-jars.sh` in quella directory. I file jar forniti come argomenti della riga di comando vengono copiati in una directory denominata creata dallo script. `/home/hadoop/datapipeline_jars` Se il cluster è configurato in modo diverso, modifica lo script in modo appropriato dopo averlo scaricato.

La sintassi per eseguire lo script sulla riga di comando è leggermente diversa da quella `bootstrapAction` mostrata nell'esempio precedente. Utilizzate gli spazi anziché le virgole tra gli argomenti, come illustrato nell'esempio seguente.

```
./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
```

Autorizzazioni Amazon EMR

Quando crei un ruolo IAM personalizzato, valuta attentamente le autorizzazioni minime necessarie al cluster per svolgere il suo lavoro. Assicurati di concedere l'accesso alle risorse richieste, come i

file in Amazon S3 o i dati in Amazon RDS, Amazon Redshift o DynamoDB. Se si desidera impostare `visibleToAllUsers` su False, il ruolo deve disporre delle autorizzazioni appropriate. Si noti che `DataPipelineDefaultRole` non dispone di tali autorizzazioni. È necessario fornire un'unione dei `DataPipelineDefaultRole` ruoli `DefaultDataPipelineResourceRole` and come ruolo `EmrCluster` oggetto oppure creare un ruolo personalizzato per questo scopo.

Sintassi

Campi Object Invocation	Description	Tipo di slot
schedule	Questo oggetto viene richiamato entro l'esecuzione di un intervallo di pianificazione. Specificare un riferimento alla pianificazione di un altro oggetto per impostare l'ordine di esecuzione e delle dipendenze per questo oggetto. È possibile soddisfare questo requisito impostando esplicitamente una pianificazione sull'oggetto, ad esempio, specificando `"schedule": {"ref": "DefaultSchedule"}` . Nella maggior parte dei casi, è meglio inserire il riferimento alla pianificazione nell'oggetto pipeline di default, in modo che tutti gli oggetti possano ereditare tale pianificazione. O, se la pipeline consiste di una struttura di pianificazioni (nidificate all'interno della pianificazione principale), è possibile creare un oggetto padre che dispone di un riferimento alla pianificazione. Per ulteriori informazioni sulle configurazioni di pianificazione opzionali di esempio, consulta https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Oggetto di riferimento, ad esempio `"schedule":{"ref": "myScheduleId"}`

Campi opzionali	Description	Tipo di slot
actionOnResourceFallimento	L'operazione intrapresa dopo il fallimento di una risorsa per questa risorsa. I valori validi sono"`retryall`", che ritenta tutte le attività per il cluster per la durata specificata, e "`retrynone `".	Stringa
actionOnTaskFallimento	L'operazione intrapresa dopo il fallimento dell'attività per questa risorsa. I valori validi sono "continue", ossia non terminare il cluster, e "terminate".	Stringa
additionalMasterSecurityGroupIds	L'identificatore dei gruppi di sicurezza principali aggiuntivi del cluster EMR, che segue il modulo sg-01. XXXX6a Per ulteriori informazioni, consulta Amazon EMR Additional Security Groups nella Amazon EMR Management Guide.	Stringa
additionalSlaveSecurityGroupIds	Identificatore di gruppi di sicurezza slave aggiuntivi del cluster EMR, che segue il modulo `sg-01XXXX6a` .	Stringa
amiVersion	La versione di Amazon Machine Image (AMI) utilizzata da Amazon EMR per installare i nodi del cluster. Per ulteriori informazioni, consulta la Guida alla gestione di Amazon EMR.	Stringa
applications	Applicazioni da installare nel cluster con argomenti separati da virgole. Hive e Pig vengono installati per impostazione predefinita. Questo parametro è applicabile solo per Amazon EMR versione 4.0 e successive.	Stringa
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa

Campi opzionali	Description	Tipo di slot
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
availabilityZone	La zona di disponibilità in cui eseguire il cluster.	Stringa
bootstrapAction	Un'operazione da eseguire all'avvio del cluster. Puoi specificare argomenti separati da virgole. Per specificare più operazioni, fino a 255, aggiungere più campi `bootstrapAction` . Il comportamento predefinito è avviare il cluster senza operazioni di bootstrap.	Stringa
configurazione	Configurazione per il cluster Amazon EMR. Questo parametro è applicabile solo per Amazon EMR versione 4.0 e successive.	Oggetto di riferimento, ad esempio, `"configuration":{"ref":"myEmrConfigurationId"}`
coreInstanceBidPrezzo	Il prezzo Spot massimo che sei disposto a pagare per le istanze Amazon EC2. Se viene specificato un prezzo di offerta, Amazon EMR utilizza le istanze Spot per il gruppo di istanze. Il prezzo è specificato in USD.	Stringa
coreInstanceCount	Numero di nodi principali da utilizzare per il cluster.	Numero intero
coreInstanceType	Il tipo di istanza Amazon EC2 da utilizzare per i nodi principali. Per informazioni, consultare Istanze Amazon EC2 supportate per cluster Amazon EMR.	Stringa

Campi opzionali	Description	Tipo di slot
coreGroupConfiguration	La configurazione per il gruppo di istanze principali del cluster Amazon EMR. Questo parametro è applicabile solo per Amazon EMR versione 4.0 e successive.	Oggetto di riferimento, ad esempio `“configuration”: {“ref”: “myEmrConfigurationId”}`
coreEbsConfiguration	La configurazione per i volumi Amazon EBS che verranno collegati a ciascuno dei nodi principali del gruppo principale del cluster Amazon EMR. Per ulteriori informazioni, consulta Tipi di istanze che supportano l'ottimizzazione di EBS nella Guida per l'utente di Amazon EC2.	Oggetto di riferimento, ad esempio `“coreEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
customAmiId	Si applica solo alla versione 5.7.0 e successive di Amazon EMR. Speciifica l'ID AMI di un'AMI personalizzata da utilizzare quando Amazon EMR effettua il provisioning delle istanze Amazon EC2. Può anche essere usato al posto delle azioni di bootstrap per personalizzare le configurazioni dei nodi del cluster. Per ulteriori informazioni, consulta il seguente argomento nella Amazon EMR Management Guide. Utilizzo di un’AMI personalizzata	Stringa

Campi opzionali	Description	Tipo di slot
EbsBlockDeviceConfig	La configurazione di un dispositivo a blocchi Amazon EBS richiesto associato al gruppo di istanze. Include un numero specificato di volumi che saranno associati a ogni istanza presente nel gruppo di istanze. Include `volumesPerInstance` e `volumeSpecification`, dove: • `volumesPerInstance` è il numero di volumi EBS con una configurazione dei volumi specifica che verrà associata a ogni istanza presente nel gruppo di istanze. • `volumeSpecification` sono le specifiche del volume Amazon EBS, come tipo di volume, IOPS e dimensione in Gigibyte (GiB), che verranno richieste per il volume EBS collegato a un'istanza EC2 nel cluster Amazon EMR.	Oggetto di riferimento, ad esempio `“EbsBlockDeviceConfig”: {“ref”: “myEbsBlockDeviceConfig”}`
emrManagedMasterSecurityGroupId	L'identificatore del gruppo di sicurezza principale del cluster Amazon EMR, che segue la forma di. `sg-01XXXX6a` Per ulteriori informazioni, consulta Configure Security Groups nella Amazon EMR Management Guide.	Stringa
emrManagedSlaveSecurityGroupId	L'identificatore del gruppo di sicurezza slave del cluster Amazon EMR, che segue il modulo. `sg-01XXXX6a`	Stringa
enableDebugging	Abilita il debug sul cluster Amazon EMR.	Stringa
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione

Campi opzionali	Description	Tipo di slot
hadoopSchedulerType	Il tipo di pianificatore del cluster. I tipi validi sono: `PARALLEL_FAIR_SCHEDULING`, `PARALLEL_CAPACITY_SCHEDULING` e `DEFAULT_SCHEDULER`.	Enumerazione
httpProxy	Host proxy che i clienti utilizzano per connettersi ai servizi AWS.	Oggetto di riferimento, ad esempio «HttpProxy»: {"ref»:» myHttpProxy Id "}
initTimeout	Il tempo di attesa prima dell'avvio della risorsa.	Periodo
keyPair	La coppia di chiavi Amazon EC2 da utilizzare per accedere al nodo master del cluster Amazon EMR.	Stringa
lateAfterTimeout	Il tempo trascorso dopo l'avvio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
masterInstanceBidPrezzo	Il prezzo Spot massimo che sei disposto a pagare per le istanze Amazon EC2. Si tratta di un valore decimale compreso tra 0 e 20,00, esclusi. Il prezzo è specificato in USD. L'impostazione di questo valore abilita le istanze Spot per il nodo master del cluster Amazon EMR. Se viene specificato un prezzo di offerta, Amazon EMR utilizza le istanze Spot per il gruppo di istanze.	Stringa
masterInstanceType	Il tipo di istanza Amazon EC2 da utilizzare per il nodo master. Per informazioni, consultare Istanze Amazon EC2 supportate per cluster Amazon EMR.	Stringa

Campi opzionali	Description	Tipo di slot
masterGroupConfiguration	La configurazione per il gruppo di istanze master del cluster Amazon EMR. Questo parametro è applicabile solo per Amazon EMR versione 4.0 e successive.	Oggetto di riferimento, ad esempio `“configuration”: {“ref”: “myEmrConfigurationId”}`
masterEbsConfiguration	La configurazione per i volumi Amazon EBS che verranno collegati a ciascuno dei nodi master del gruppo principale nel cluster Amazon EMR. Per ulteriori informazioni, consulta Tipi di istanze che supportano l'ottimizzazione di EBS nella Guida per l'utente di Amazon EC2.	Oggetto di riferimento, ad esempio `“masterEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
maxActiveInstances	Il numero massimo di istanze attive simultanee di un componente. Le riesecuzioni non contano ai fini del numero di istanze attive.	Numero intero
maximumRetries	Numero massimo di tentativi in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio, `"onFail": {"ref":"myActionId"}`
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio, `"onLateAction": {"ref":"myActionId"}`

Campi opzionali	Description	Tipo di slot
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio, `"onSuccess": {"ref":"myActionId"}`
parent	Padre dell'oggetto corrente da cui vengono ereditati gli slot.	Oggetto di riferimento, ad esempio. `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	L'URI di Amazon S3 (ad esempio 's3://BucketName/Key/ ') per caricare i log per la pipeline.	Stringa
region	Il codice per la regione in cui deve essere eseguito il cluster Amazon EMR. Per impostazione predefinita, il cluster viene eseguito nella stessa regione della pipeline. È possibile eseguire il cluster nella stessa regione del set di dati dipendenti.	Enumerazione
releaseLabel	Etichetta release per il cluster EMR.	Stringa
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
resourceRole	Il ruolo IAM AWS Data Pipeline utilizzato per creare il cluster Amazon EMR. Il ruolo predefinito è `DataPipelineDefaultRole` .	Stringa

Campi opzionali	Description	Tipo di slot
retryDelay	La durata del timeout tra due tentativi.	Periodo
role	Il ruolo IAM è passato ad Amazon EMR per creare nodi EC2.	Stringa
runsOn	Campo non consentito su questo oggetto.	Oggetto di riferimento, ad esempio, `"runsOn": {"ref":"myResourceId"}`
Configurazione di sicurezza	L'identificatore della configurazione di sicurezza EMR che verrà applicata al cluster. Questo parametro è applicabile solo per Amazon EMR versione 4.8.0 e successive.	Stringa
serviceAccessSecurityGroupId	L'identificatore per il gruppo di sicurezza dell'accesso al servizio del cluster Amazon EMR.	Stringa. Segue la forma `sg-01XXXX6a` , ad esempio, `sg-1234abcd` .

Campi opzionali	Description	Tipo di slot
scheduleType	Il tipo di pianificazione consente di specifica re se gli oggetti nella definizione di pipeline devono essere programmati all'inizio o alla fine dell'intervallo. I valori sono `cron`, `ondemand` e `timeseries` . La pianificazione `timeserie s` significa che le istanze sono programmate al termine di ogni intervallo. La pianificazione `cron` significa che le istanze sono programma te all'inizio di ogni intervallo. Una pianificazione `ondemand`consente di eseguire una pipeline una sola volta, per attivazione. Non è necessari o clonare o ricreare la pipeline per eseguirla di nuovo. Se utilizzi una pianificazione `ondemand`, devi specificarlo nell'oggetto predefinito e deve essere l'unico `scheduleType` specificato per gli oggetti della pipeline. Per utilizzare le pipeline `ondemand`, chiama l'operazione `ActivatePipeline` per ogni esecuzione successiva.	Enumerazione
subnetId	L'identificatore della sottorete in cui avviare il cluster Amazon EMR.	Stringa
supportedProducts	Un parametro che installa software di terze parti su un cluster Amazon EMR, ad esempio una distribuzione di terze parti di Hadoop.	Stringa
taskInstanceBidPre zzo	Il prezzo Spot massimo che sei disposto a pagare per le istanze EC2. Un valore decimale compreso tra 0 e 20,00, esclusi. Il prezzo è specificato in USD. Se viene specificato un prezzo di offerta, Amazon EMR utilizza le istanze Spot per il gruppo di istanze.	Stringa

Campi opzionali	Description	Tipo di slot
taskInstanceCount	Il numero di nodi di attività da utilizzare per il cluster Amazon EMR.	Numero intero
taskInstanceType	Il tipo di istanza Amazon EC2 da utilizzare per i nodi di attività.	Stringa
taskGroupConfiguration	La configurazione per il gruppo di attività del cluster Amazon EMR. Questo parametro è applicabile solo per Amazon EMR versione 4.0 e successive.	Oggetto di riferimento, ad esempio `“configuration”: {“ref”: “myEmrConfigurationId”}`
taskEbsConfiguration	La configurazione per i volumi Amazon EBS che verranno collegati a ciascuno dei nodi di attività del gruppo di attività nel cluster Amazon EMR. Per ulteriori informazioni, consulta Tipi di istanze che supportano l'ottimizzazione di EBS nella Guida per l'utente di Amazon EC2.	Oggetto di riferimento, ad esempio `“taskEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
terminateAfter	Termina la risorsa dopo queste numerose ore.	Numero intero

Campi opzionali	Description	Tipo di slot
VolumeSpecification	Le specifiche del volume di Amazon EBS, come tipo di volume, IOPS e dimensione in Gigibyte (GiB), che verranno richieste per il volume Amazon EBS collegato a un'istanza Amazon EC2 nel cluster Amazon EMR. Il nodo può essere principale, master o di task. `VolumeSpecification` include: • `iops()`intero. Il numero di I/O operazioni al secondo (IOPS) supportato dal volume Amazon EBS, ad esempio 1000. Per ulteriori informazioni, consulta EBS I/O Characteristics nella Amazon EC2 User Guide. • `sizeinGB()` . Numero intero. La dimensione del volume Amazon EBS, in gibibyte (GiB), ad esempio 500. Per informazioni sulle combinazioni valide di tipi di volume e dimensioni dei dischi rigidi, consulta EBS Volume Types nella Amazon EC2 User Guide. • `volumetType` . Stringa. Il tipo di volume Amazon EBS, ad esempio gp2. I tipi di volume supportati includono standard, gp2, io1, st1, sc1 e molti altri. Per ulteriori informazioni, consulta EBS Volume Types nella Amazon EC2 User Guide.	Oggetto di riferimento, ad esempio `“VolumeSpecification”: {“ref”: “myVolumeSpecification”}`
useOnDemandOnLastAttempt	Nell'ultimo tentativo di richiesta di una risorsa, effettuare una richiesta per istanze on demand invece che per istanze Spot. In questo modo, se tutti i tentativi precedenti non sono andati a buon fine, l'ultimo tentativo non viene interrotto.	Booleano
workerGroup	Campo non consentito su questo oggetto.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
emrStepLog	I log dei passaggi sono disponibili solo per i tentativi di attività di Amazon EMR.	Stringa
errorId	ID dell'errore se l'oggetto non è riuscito.	Stringa
errorMessage	Messaggio di errore se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
@failureReason	Il motivo dell'errore della risorsa.	Stringa
@finishedTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
hadoopJobLog	I log dei lavori Hadoop sono disponibili sui tentativi di attività di Amazon EMR.	Stringa

Campi Runtime	Description	Tipo di slot
@healthStatus	Lo stato di integrità dell'oggetto che riflette l'esito positivo o negativo dell'ultima istanza dell'oggetto che ha raggiunto lo stato di un'istanza terminata.	Stringa
@healthStatusFromInstanceId	Id dell'ultimo oggetto dell'istanza che ha raggiunto lo stato terminato.	Stringa
@ Ora healthStatusUpdated	L'ora in cui lo stato di integrità è stato aggiornato l'ultima volta.	DateTime
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
@lastDeactivatedTime	L'ora in cui l'oggetto è stato disattivato.	DateTime
@ latestCompletedRun Ora	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata completata.	DateTime
@latestRunTime	L'orario dell'esecuzione più recente durante il quale l'esecuzione è stata pianificata.	DateTime
@nextRunTime	L'orario dell'esecuzione da programmare come successiva.	DateTime
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	La descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La posizione di un oggetto nel ciclo di vita. I Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Esempi

I seguenti sono esempi di questo tipo di oggetto.

Indice

- Usa EmrCluster la risorsa nell'SDK AWS per Java
- Configurare un cluster Amazon EMR in una sottorete privata
- Collegare i volumi EBS ai nodi del cluster

Avvia un cluster Amazon EMR con HadoopVersion

Example

L'esempio seguente avvia un cluster Amazon EMR utilizzando AMI versione 1.0 e Hadoop 0.20.

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
  "hadoopVersion" : "0.20",
  "keyPair" : "my-key-pair",
  "masterInstanceType" : "m3.xlarge",
  "coreInstanceType" : "m3.xlarge",
  "coreInstanceCount" : "10",
  "taskInstanceType" : "m3.xlarge",
  "taskInstanceCount": "10",
  "bootstrapAction" : ["s3://Region.elasticmapreduce/bootstrap-actions/configure-
hadoop,arg1,arg2,arg3","s3://Region.elasticmapreduce/bootstrap-actions/configure-
hadoop/configure-other-stuff,arg1,arg2"]
}
```

Avvia un cluster Amazon EMR con etichetta di rilascio emr-4.x o successiva

Example

L'esempio seguente avvia un cluster Amazon EMR utilizzando il `releaseLabel` campo più recente:

```
{
  "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"}
}
```

Installa software aggiuntivo sul tuo cluster Amazon EMR

Example

`EmrCluster`fornisce il `supportedProducts` campo che installa software di terze parti su un cluster Amazon EMR, ad esempio consente di installare una distribuzione personalizzata di Hadoop, come MapR. Accetta un elenco separato da virgole di argomenti per il software di terze parti da leggere e in base al quale agire. L'esempio seguente mostra come usare il campo `supportedProducts` di `EmrCluster` per creare un cluster personalizzato edizione MapR M3 con Karmasphere Analytics installato ed eseguire un oggetto `EmrActivity` su di esso.

```
{
    "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"
}
```

Disabilitare la crittografia lato server sulle versioni 3.x

Example

Un'`EmrCluster`attività con una versione Hadoop 2.x creata da abilita la crittografia lato server per impostazione predefinita. AWS Data Pipeline Se si desidera disattivare la crittografia lato server, è necessario specificare un'operazione di bootstrap nella definizione di un oggetto del cluster.

L'esempio seguente crea un'attività `EmrCluster` con la crittografia lato server disabilitata:

```
{
   "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"]
}
```

Disabilitare la crittografia lato server sulle versioni 4.x

Example

È necessario disabilitare la crittografia lato server utilizzando un oggetto `EmrConfiguration`.

L'esempio seguente crea un'attività `EmrCluster` con la crittografia lato server disabilitata:

```
   {
      "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"
    }
```

Configura Hadoop KMS e crea zone di crittografia in HDFS ACLs

Example

I seguenti oggetti vengono creati ACLs per Hadoop KMS e creano zone di crittografia e chiavi di crittografia corrispondenti 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"
    }
```

Specificare ruoli IAM personalizzati

Example

Per impostazione predefinita, AWS Data Pipeline passa `DataPipelineDefaultRole` come ruolo del servizio Amazon EMR e `DataPipelineDefaultResourceRole` come profilo dell'istanza Amazon EC2 per creare risorse per tuo conto. Tuttavia, puoi creare un ruolo di servizio Amazon EMR personalizzato e un profilo di istanza personalizzato e utilizzarli al loro posto. AWS Data Pipeline deve disporre di autorizzazioni sufficienti per creare cluster utilizzando il ruolo personalizzato e deve essere aggiunto AWS Data Pipeline come entità attendibile.

L'oggetto di esempio seguente specifica i ruoli personalizzati per il cluster Amazon EMR:

```
{
   "id":"MyEmrCluster",
```

```
   "type":"EmrCluster",
   "hadoopVersion":"2.x",
   "keyPair":"my-key-pair",
   "masterInstanceType":"m3.xlarge",
   "coreInstanceType":"m3.large",
   "coreInstanceCount":"10",
   "taskInstanceType":"m3.large",
   "taskInstanceCount":"10",
   "role":"emrServiceRole",
   "resourceRole":"emrInstanceProfile"
}
```

Usa EmrCluster la risorsa nell'SDK AWS per Java

Example

L'esempio seguente mostra come utilizzare un `EmrCluster` e `EmrActivity` creare un cluster Amazon EMR 4.x per eseguire una fase Spark utilizzando Java SDK:

```
public class dataPipelineEmr4 {

  public static void main(String[] args) {

 AWSCredentials credentials = null;
 credentials = new ProfileCredentialsProvider("/path/to/
AwsCredentials.properties","default").getCredentials();
 DataPipelineClient dp = new DataPipelineClient(credentials);
 CreatePipelineRequest createPipeline = new
 CreatePipelineRequest().withName("EMR4SDK").withUniqueId("unique");
 CreatePipelineResult createPipelineResult = dp.createPipeline(createPipeline);
 String pipelineId = createPipelineResult.getPipelineId();

 PipelineObject emrCluster = new PipelineObject()
     .withName("EmrClusterObj")
     .withId("EmrClusterObj")
     .withFields(
   new Field().withKey("releaseLabel").withStringValue("emr-4.1.0"),
   new Field().withKey("coreInstanceCount").withStringValue("3"),
   new Field().withKey("applications").withStringValue("spark"),
   new Field().withKey("applications").withStringValue("Presto-Sandbox"),
   new Field().withKey("type").withStringValue("EmrCluster"),
   new Field().withKey("keyPair").withStringValue("myKeyName"),
   new Field().withKey("masterInstanceType").withStringValue("m3.xlarge"),
```

```
    new Field().withKey("coreInstanceType").withStringValue("m3.xlarge")
    );

 PipelineObject emrActivity = new PipelineObject()
     .withName("EmrActivityObj")
     .withId("EmrActivityObj")
     .withFields(
    new Field().withKey("step").withStringValue("command-runner.jar,spark-submit,--
executor-memory,1g,--class,org.apache.spark.examples.SparkPi,/usr/lib/spark/lib/spark-
examples.jar,10"),
    new Field().withKey("runsOn").withRefValue("EmrClusterObj"),
    new Field().withKey("type").withStringValue("EmrActivity")
    );

 PipelineObject schedule = new PipelineObject()
     .withName("Every 15 Minutes")
     .withId("DefaultSchedule")
     .withFields(
    new Field().withKey("type").withStringValue("Schedule"),
    new Field().withKey("period").withStringValue("15 Minutes"),
    new Field().withKey("startAt").withStringValue("FIRST_ACTIVATION_DATE_TIME")
    );

 PipelineObject defaultObject = new PipelineObject()
     .withName("Default")
     .withId("Default")
     .withFields(
    new Field().withKey("failureAndRerunMode").withStringValue("CASCADE"),
    new Field().withKey("schedule").withRefValue("DefaultSchedule"),
    new
 Field().withKey("resourceRole").withStringValue("DataPipelineDefaultResourceRole"),
    new Field().withKey("role").withStringValue("DataPipelineDefaultRole"),
    new Field().withKey("pipelineLogUri").withStringValue("s3://myLogUri"),
    new Field().withKey("scheduleType").withStringValue("cron")
    );

 List<PipelineObject> pipelineObjects = new ArrayList<PipelineObject>();

 pipelineObjects.add(emrActivity);
 pipelineObjects.add(emrCluster);
 pipelineObjects.add(defaultObject);
 pipelineObjects.add(schedule);

 PutPipelineDefinitionRequest putPipelineDefintion = new PutPipelineDefinitionRequest()
```

```
      .withPipelineId(pipelineId)
      .withPipelineObjects(pipelineObjects);

  PutPipelineDefinitionResult putPipelineResult =
  dp.putPipelineDefinition(putPipelineDefintion);
  System.out.println(putPipelineResult);

  ActivatePipelineRequest activatePipelineReq = new ActivatePipelineRequest()
      .withPipelineId(pipelineId);
  ActivatePipelineResult activatePipelineRes = dp.activatePipeline(activatePipelineReq);

       System.out.println(activatePipelineRes);
       System.out.println(pipelineId);

     }

}
```

Configurare un cluster Amazon EMR in una sottorete privata

Example

Questo esempio include una configurazione che avvia il cluster in una sottorete privata in un VPC. Per ulteriori informazioni, consulta Launch Amazon EMR Clusters in un VPC nella Amazon EMR Management Guide. Questa configurazione è opzionale. È possibile utilizzarla in qualsiasi pipeline che utilizza un oggetto `EmrCluster`.

Per avviare un cluster Amazon EMR in una sottorete privata, specifica `SubnetId` «,» e `serviceAccessSecurityGroupId` nella `emrManagedSlaveSecurityGroupId` tua configurazione. `emrManagedMasterSecurityGroupId` `EmrCluster`

```
{
  "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"
  }
}
```

Collegare i volumi EBS ai nodi del cluster

Example

È possibile allegare i volumi EBS a qualsiasi tipo di nodo nel cluster EMR all'interno della pipeline. Per allegare volumi EBS ai nodi, utilizzare `coreEbsConfiguration`, `masterEbsConfiguration` e `TaskEbsConfiguration` nella configurazione `EmrCluster`.

Questo esempio di cluster Amazon EMR utilizza i volumi Amazon EBS per i nodi master, task e core. Per ulteriori informazioni, consulta i volumi Amazon EBS in Amazon EMR nella Amazon EMR Management Guide.

Queste configurazioni sono opzionali. È possibile utilizzarle in qualsiasi pipeline che utilizza un oggetto `EmrCluster`.

Nella pipeline, fare clic sulla configurazione dell'oggetto `EmrCluster`, scegliere Master EBS Configuration (Configurazione Master EBS), Core EBS Configuration (Configurazione EBS Core) o Task EBS Configuration (Configurazione EBS Task), quindi immettere i dettagli di configurazione simili a quello dell'esempio seguente.

```
{
  "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"
  }
}
```

Vedi anche

- EmrActivity

HttpProxy

HttpProxy consente di configurare il proprio proxy e fare in modo che Task Runner acceda al AWS Data Pipeline servizio tramite esso. Non è necessario configurare un Task Runner in esecuzione con queste informazioni.

Esempio di un in HttpProxy TaskRunner

La seguente definizione di pipeline mostra un oggetto `HttpProxy`:

```
{
  "objects": [
    {
      "schedule": {
```

```
        "ref": "Once"
      },
      "pipelineLogUri": "s3://myDPLogUri/path",
      "name": "Default",
      "id": "Default"
    },
    {
      "name": "test_proxy",
      "hostname": "hostname",
      "port": "port",
      "username": "username",
      "*password": "password",
      "windowsDomain": "windowsDomain",
      "type": "HttpProxy",
      "id": "test_proxy",
    },
    {
      "name": "ShellCommand",
      "id": "ShellCommand",
      "runsOn": {
        "ref": "Resource"
      },
      "type": "ShellCommandActivity",
      "command": "echo 'hello world' "
    },
    {
      "period": "1 day",
      "startDateTime": "2013-03-09T00:00:00",
      "name": "Once",
      "id": "Once",
      "endDateTime": "2013-03-10T00:00:00",
      "type": "Schedule"
    },
    {
      "role": "dataPipelineRole",
      "httpProxy": {
        "ref": "test_proxy"
      },
      "actionOnResourceFailure": "retrynone",
      "maximumRetries": "0",
      "type": "Ec2Resource",
      "terminateAfter": "10 minutes",
      "resourceRole": "resourceRole",
      "name": "Resource",
```

```
      "actionOnTaskFailure": "terminate",
      "securityGroups": "securityGroups",
      "keyPair": "keyPair",
      "id": "Resource",
      "region": "us-east-1"
    }
  ],
  "parameters": []
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
hostname	Hosting del proxy che i clienti possano utilizzar e per connettersi ai servizi AWS.	Stringa
port	Porta dell'host proxy che i clienti possano utilizzare per connettersi ai Servizi AWS.	Stringa

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
*password	Password per il proxy.	Stringa
s3 NoProxy	Disabilita il proxy HTTP durante la connessione ad Amazon S3	Booleano
username	Nome utente per il proxy	Stringa
windowsDomain	Il nome del dominio di Windows per il proxy NTLM.	Stringa

Campi opzionali	Description	Tipo di slot
windowsWorkgroup	Il nome del gruppo di lavoro di Windows per il proxy NTLM.	Stringa

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Precondizioni

I seguenti sono gli oggetti AWS Data Pipeline precondizionati:

Oggetti

- La dinamo esiste DBData
- La dinamo DBTable esiste
- Exists
- S3 KeyExists
- S3 PrefixNotEmpty

- ShellCommandPrecondition

La dinamo esiste DBData

Un prerequisito per verificare l'esistenza di dati in una tabella DynamoDB.

Sintassi

Campi obbligatori	Description	Tipo di slot
role	Specifica il ruolo da utilizzare per eseguire la precondizione.	Stringa
tableName	Tabella DynamoDB da verificare.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero

Campi opzionali	Description	Tipo di slot
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
preconditionTimeout	Il periodo dall'inizio dopo il quale la precondizione viene contrassegnata come non riuscita se ancora non è stata soddisfatta.	Periodo
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato, le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» myRunnableObject Id "}
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
currentRetryCount	Numero di volte in cui la precondizione è stato provata in questo tentativo.	Stringa
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa

Campi Runtime	Description	Tipo di slot
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
lastRetryTime	L'ultima volta in cui la precondizione è stato provata all'interno di questo tentativo.	Stringa
nodo	Il nodo per il quale viene eseguita questa precondizione	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnabl eObject Id "}
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimen to, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa

Campi di sistema	Description	Tipo di slot
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

La dinamo DBTable esiste

Una precondizione per verificare l'esistenza della tabella DynamoDB.

Sintassi

Campi obbligatori	Description	Tipo di slot
role	Specifica il ruolo da utilizzare per eseguire la precondizione.	Stringa
tableName	Tabella DynamoDB da verificare.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo

Campi opzionali	Description	Tipo di slot
failureAndRerunModo	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
preconditionTimeout	Il periodo dall'inizio dopo il quale la precondizione viene contrassegnata come non riuscita se ancora non è stata soddisfatta.	Periodo

Campi opzionali	Description	Tipo di slot
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» myRunnableObject Id "}
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
currentRetryCount	Numero di volte in cui la precondizione è stato provata in questo tentativo.	Stringa

Campi Runtime	Description	Tipo di slot
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
lastRetryTime	L'ultima volta in cui la precondizione è stato provata all'interno di questo tentativo.	Stringa
nodo	Il nodo per il quale viene eseguita questa precondizione	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnabl eObject Id "}
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi Runtime	Description	Tipo di slot
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Exists

Verifica se esiste un oggetto nodo di dati.

> **Note**
>
> È consigliabile utilizzare precondizioni gestite dal sistema. Per ulteriori informazioni, consulta Precondizioni.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. L'oggetto `InputData` fa riferimento a questo oggetto, `Ready`, più a un altro oggetto definito nello stesso file di definizione della pipeline. `CopyPeriod` è un oggetto `Schedule`.

```
{
  "id" : "InputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://amzn-s3-demo-bucket/InputData/#{@scheduledStartTime.format('YYYY-
MM-dd-hh:mm')}.csv",
  "precondition" : { "ref" : "Ready" }
},
{
  "id" : "Ready",
  "type" : "Exists"
}
```

Sintassi

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero

Campi opzionali	Description	Tipo di slot
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
preconditionTimeout	Il periodo dall'inizio dopo il quale la precondizione viene contrassegnata come non riuscita se ancora non è stata soddisfatta.	Periodo
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato, le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» myRunnableObject Id "}
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio "cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa

Campi Runtime	Description	Tipo di slot
nodo	Il nodo per il quale viene eseguita questa precondizione.	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnabl eObject Id "}
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimen to, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects	Stringa

Campi di sistema	Description	Tipo di slot
	generano Instance Objects che eseguono Attempt Objects.	

Vedi anche

- ShellCommandPrecondition

S3 KeyExists

Verifica se esiste una chiave in un nodo di dati Amazon S3.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. La precondizione sarà attivata quando esiste la chiave `s3://amzn-s3-demo-bucket/mykey`, a cui si fa riferimento tramite il parametro `s3Key`.

```
{
"id" : "InputReady",
"type" : "S3KeyExists",
"role" : "test-role",
"s3Key" : "s3://amzn-s3-demo-bucket/mykey"
}
```

È inoltre possibile utilizzare `S3KeyExists` come precondizione nella seconda pipeline che attende il termine della prima pipeline. A tale scopo:

1. Scrivi un file su Amazon S3 al termine del completamento della prima pipeline.
2. Crea una precondizione `S3KeyExists` nella seconda pipeline.

Sintassi

Campi obbligatori	Description	Tipo di slot
role	Specifica il ruolo da utilizzare per eseguire la precondizione.	Stringa
s3Key	La chiave Amazon S3.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout prima di tentare nuovamente di completare il lavoro in remoto. Se impostato , un'attività remota che non viene completat a entro il tempo impostato dopo l'avvio viene tentata di nuovo.	Periodo
failureAndRerunMod alità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite.	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completat o. Viene attivato solo quando il tipo di pianifica zione non è impostato su. ondemand	Periodo
maximumRetries	Numero massimo di tentativi che vengono avviati in caso di errore.	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}

Campi opzionali	Description	Tipo di slot
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
preconditionTimeout	Il periodo dall'inizio dopo il quale la precondizione viene contrassegnata come non riuscita se ancora non è stata soddisfatta.	Periodo
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a `reportProgress` . Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebbero essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi consecutivi.	Periodo

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeInstances»: {"ref»:»

Campi Runtime	Description	Tipo di slot
		myRunnableObject Id "}
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
currentRetryCount	Numero di volte in cui la precondizione è stato provata in questo tentativo.	Stringa
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa

Campi Runtime	Description	Tipo di slot
lastRetryTime	L'ultima volta in cui la precondizione è stato provata all'interno di questo tentativo.	Stringa
nodo	Il nodo per il quale viene eseguita questa precondizione	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnabl eObject Id "}
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimen to, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa

Campi di sistema	Description	Tipo di slot
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Vedi anche

- ShellCommandPrecondition

S3 PrefixNotEmpty

Una precondizione per verificare la presenza degli oggetti Amazon S3 con il prefisso specificato (rappresentato come URI).

Esempio

Di seguito è riportato un esempio di questo tipo di oggetto utilizzando campi obbligatori, facoltativi e di espressioni.

```
{
  "id" : "InputReady",
  "type" : "S3PrefixNotEmpty",
  "role" : "test-role",
  "s3Prefix" : "#{node.filePath}"
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
role	Specifica il ruolo da utilizzare per eseguire la precondizione.	Stringa
s3Prefix	Il prefisso Amazon S3 per verificare l'esistenza di oggetti.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attività remota che non viene completata entro il tempo impostato di avvio viene tentata di nuovo.	Periodo
failureAndRerunModalità	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. `ondemand`	Periodo
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»:

Campi opzionali	Description	Tipo di slot
		{"ref»:» myBaseObj ect Id "}
preconditionTimeout	Il periodo dall'inizio dopo il quale la precondiz ione viene contrassegnata come non riuscita se ancora non è stata soddisfatta.	Periodo
reportProgressTime out	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» myRunnableObject Id "}
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimen to, ad esempio

Campi Runtime	Description	Tipo di slot
		"cascadeFailedOn«: {" ref»:» myRunnabl eObject Id "}
currentRetryCount	Numero di volte in cui la precondizione è stato provata in questo tentativo.	Stringa
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
lastRetryTime	L'ultima volta in cui la precondizione è stato provata all'interno di questo tentativo.	Stringa
nodo	Il nodo per il quale viene eseguita questa precondizione.	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnabl eObject Id "}
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto.	DateTime

Campi Runtime	Description	Tipo di slot
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto.	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Vedi anche

- ShellCommandPrecondition

ShellCommandPrecondition

Un comando di Unix/Linux shell che può essere eseguito come condizione preliminare.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "id" : "VerifyDataReadiness",
  "type" : "ShellCommandPrecondition",
  "command" : "perl check-data-ready.pl"
}
```

Sintassi

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
command	Il comando da eseguire. Questo valore ed eventuali parametri associati devono funzionar e nell'ambiente da cui si sta eseguendo il Task Runner.	Stringa
scriptUri	Un percorso URI di Amazon S3 per un file da scaricare ed eseguire come comando shell. Deve essere presente solo uno scriptUri o un campo di comando. Se lo scriptUri non è in grado di utilizzare i parametri, utilizzare il comando.	Stringa

Campi opzionali	Description	Tipo di slot
attemptStatus	Lo stato segnalato più di recente dall'attività remota.	Stringa
attemptTimeout	Timeout per il completamento del lavoro in remoto. Se questo campo è impostato, un'attivi tà remota che non viene completata entro	Periodo

Campi opzionali	Description	Tipo di slot
	il tempo impostato di avvio viene tentata di nuovo.	
failureAndRerunModo	Descrive il comportamento del nodo consumer quando le dipendenze presentano un errore o vengono di nuovo eseguite	Enumerazione
lateAfterTimeout	Il tempo trascorso dopo l'inizio della pipeline entro il quale l'oggetto deve essere completato. Viene attivato solo quando il tipo di pianificazione non è impostato su. ondemand	Periodo
maximumRetries	Numero massimo di tentativi in caso di errore	Numero intero
onFail	Un'azione da eseguire quando l'oggetto corrente ha esito negativo.	Oggetto di riferimento, ad esempio «onFail»: {"ref»:» myActionId «}
onLateAction	Azioni che devono essere attivate se un oggetto non è stato ancora pianificato o non è ancora completo.	Oggetto di riferimento, ad esempio "onLateAction«: {" ref»:» myActionId «}
onSuccess	Un'operazione da eseguire quando l'oggetto corrente ha esito positivo.	Oggetto di riferimento, ad esempio «onSuccess»: {"ref»:» myActionId «}
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObject Id "}
preconditionTimeout	Il periodo dall'inizio dopo il quale la precondizione viene contrassegnata come non riuscita se ancora non è stata soddisfatta.	Periodo

Campi opzionali	Description	Tipo di slot
reportProgressTimeout	Timeout per chiamate successive di attività in remoto a reportProgress. Se impostato , le attività in remoto che non presentano avanzamenti nel periodo specificato potrebber o essere considerate bloccate e sono quindi oggetto di un altro tentativo.	Periodo
retryDelay	La durata del timeout tra due tentativi.	Periodo
scriptArgument	Argomento da passare allo script della shell	Stringa
stderr	Il percorso Amazon S3 che riceve i messaggi di errore di sistema reindirizzati dal comando. Se utilizzi il `runs0n` campo, deve trattarsi di un percorso Amazon S3 a causa della natura transitoria della risorsa che esegue la tua attività. Tuttavia, se specifichi il campo `workerGroup` , viene autorizzato un percorso file locale.	Stringa
stdout	Il percorso Amazon S3 che riceve l'output reindirizzato dal comando. Se utilizzi il `runs0n` campo, deve trattarsi di un percorso Amazon S3 a causa della natura transitoria della risorsa che esegue la tua attività. Tuttavia, se specifich i il campo `workerGroup` , viene autorizzato un percorso file locale.	Stringa

Campi Runtime	Description	Tipo di slot
@activeInstances	Elenco di oggetti di istanze attive attualmente programmate.	Oggetto di riferimento, ad esempio «activeIn stances»: {"ref»:» Id "} myRunnableObject

Campi Runtime	Description	Tipo di slot
@actualEndTime	L'ora in cui è terminata l'esecuzione di questo oggetto.	DateTime
@actualStartTime	L'ora in cui è stata avviata l'esecuzione di questo oggetto.	DateTime
cancellationReason	CancellationReason se questo oggetto è stato annullato.	Stringa
@cascadeFailedOn	Descrizione della catena di dipendenza che ha generato l'errore dell'oggetto.	Oggetto di riferimento, ad esempio "cascadeFailedOn«: {" ref»:» myRunnableObject Id "}
emrStepLog	Log della fase EMR disponibili solo sui tentativi delle attività EMR	Stringa
errorId	ErrorId se l'oggetto non è riuscito.	Stringa
errorMessage	ErrorMessage se l'oggetto non è riuscito.	Stringa
errorStackTrace	Traccia dello stack di errore se l'oggetto non è riuscito.	Stringa
hadoopJobLog	Log delle attività Hadoop disponibili per le attività basate su EMR.	Stringa
hostname	Il nome host del client che si è aggiudicato il tentativo dell'attività.	Stringa
nodo	Il nodo per il quale viene eseguita questa precondizione	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnableObject Id "}

Campi Runtime	Description	Tipo di slot
reportProgressTime	Il periodo di tempo più recente in cui l'attività remota ha segnalato un progresso.	DateTime
@scheduledEndTime	L'orario di termine della pianificazione per un oggetto	DateTime
@scheduledStartTime	L'orario di inizio della pianificazione per l'oggetto	DateTime
@status	Lo stato di questo oggetto.	Stringa
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa
@waitingOn	Descrizione dell'elenco di dipendenze per cui questo oggetto è in attesa.	Oggetto di riferimento, ad esempio «waitingOn»: {"ref»:» myRunnableObject Id "}

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Vedi anche

- ShellCommandActivity

- Exists

Database

I seguenti sono gli oggetti del AWS Data Pipeline database:

Oggetti

- JdbcDatabase
- RdsDatabase
- RedshiftDatabase

JdbcDatabase

Rimuove un database JDBC.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "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"
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
connectionString	La stringa di connessione JDBC per accedere al database.	Stringa
jdbcDriverClass	La classe del driver da caricare prima di stabilire la connessione JDBC.	Stringa

Campi obbligatori	Description	Tipo di slot
*password	La password da fornire.	Stringa
username	Il nome utente da fornire durante la connessio ne al database.	Stringa

Campi opzionali	Description	Tipo di slot
databaseName	Nome del database logico a cui collegarsi	Stringa
jdbcDriverJarUri	Il percorso in Amazon S3 del file JAR del driver JDBC utilizzato per la connessione al database. AWS Data Pipeline deve avere l'autorizzazione a leggere questo file JAR.	Stringa
jdbcProperties	Coppie della forma A=B da impostare come proprietà sulle connessioni JDBC per questo database.	Stringa
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa

Campi di sistema	Description	Tipo di slot
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

RdsDatabase

Definisce un database Amazon RDS.

> **Note**
>
> RdsDatabase non supporta Aurora. Usalo the section called “JdbcDatabase” per Aurora, invece.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "id" : "MyRdsDatabase",
  "type" : "RdsDatabase",
  "region" : "us-east-1",
  "username" : "user_name",
  "*password" : "my_password",
  "rdsInstanceId" : "my_db_instance_identifier"
}
```

Per il motore Oracle, il campo `jdbcDriverJarUri` è obbligatorio ed è possibile specificare il seguente driver: `http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html`. Per il motore SQL, il campo `jdbcDriverJarUri` è obbligatorio ed è possibile specificare il seguente driver: `https://www.microsoft.com/en-us/`

`download/details.aspx?displaylang=en&id=11774`. Per i motori MySQL e PostgreSQL, il campo `jdbcDriverJarUri` è facoltativo.

Sintassi

Campi obbligatori	Description	Tipo di slot
*password	La password da fornire.	Stringa
rdsInstanceId	La `DBInstanceIdentifier` proprietà dell'istanza DB.	Stringa
username	Il nome utente da fornire durante la connessio ne al database.	Stringa

Campi opzionali	Description	Tipo di slot
databaseName	Nome del database logico a cui collegarsi	Stringa
jdbcDriverJarUri	Il percorso in Amazon S3 del file JAR del driver JDBC utilizzato per la connessione al database. AWS Data Pipeline deve avere l'autorizzazione a leggere questo file JAR. Per i motori MySQL e PostgreSQL, il driver predefini to viene utilizzato se questo campo non è specificato, ma è possibile sostituire l'imposta zione predefinita utilizzando tale campo. Per i motori Oracle e SQL Server, questo campo è obbligatorio.	Stringa
jdbcProperties	Coppie della forma A=B da impostare come proprietà sulle connessioni JDBC per questo database.	Stringa
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»:

Campi opzionali	Description	Tipo di slot
		{"ref»:» myBaseObj ect Id "}
region	Il codice per la regione in cui esiste il database. Ad esempio, us-east-1.	Stringa

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

RedshiftDatabase

Definisce un database Amazon Redshift. `RedshiftDatabase`appresenta le proprietà del database utilizzato dalla pipeline.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
```

```
  "id" : "MyRedshiftDatabase",
  "type" : "RedshiftDatabase",
  "clusterId" : "myRedshiftClusterId",
  "username" : "user_name",
  "*password" : "my_password",
  "databaseName" : "database_name"
}
```

Per impostazione predefinita, l'oggetto usa il driver Postgres, che richiede il campo `clusterId`. Per utilizzare il driver Amazon Redshift, specifica invece la stringa di connessione al database Amazon Redshift dalla console Amazon Redshift (inizia con «jdbc:redshift:») nel campo. `connectionString`

Sintassi

Campi obbligatori	Description	Tipo di slot
*password	La password da fornire.	Stringa
username	Il nome utente da fornire durante la connessio ne al database.	Stringa

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
clusterId	L'identificatore fornito dall'utente al momento della creazione del cluster Amazon Redshift. Ad esempio, se l'endpoint per il tuo cluster Amazon Redshift è mydb.example.us-ea st-1.redshift.amazonaws.com, l'identificatore corretto è. `mydb` Nella console Amazon Redshift, puoi ottenere questo valore da Cluster Identifier o Cluster Name.	Stringa
connectionString	L'endpoint JDBC per la connessione a un'istanz a Amazon Redshift di proprietà di un account	Stringa

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
	diverso dalla pipeline. Non è possibile specifica re `connectionString` e `clusterId` .	

Campi opzionali	Description	Tipo di slot
databaseName	Nome del database logico a cui collegarsi.	Stringa
jdbcProperties	Coppie della forma A=B da impostare come proprietà sulle connessioni JDBC per questo database.	Stringa
parent	Padre dell'oggetto corrente da cui vengono ereditati gli slot.	Oggetto di riferimen to, ad esempio «parent»: {"ref»:» Id "} myBaseObject
region	Il codice per la regione in cui esiste il database. Ad esempio, us-east-1.	Enumerazione

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa

Campi di sistema	Description	Tipo di slot
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Formati dei dati

Di seguito sono riportati gli oggetti in formato AWS Data Pipeline dati:

Oggetti

- Formato dei dati CSV
- Formato di dati personalizzato
- Formato Dynamo DBData
- Dinamo DBExport DataFormat
- RegEx Formato dei dati
- Formato dei dati TSV

Formato dei dati CSV

Un formato di dati separati da virgola in cui il separatore di colonna è una virgola e il separatore di record è un carattere in una nuova riga.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "id" : "MyOutputDataType",
  "type" : "CSV",
  "column" : [
    "Name STRING",
```

```
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintassi

Campi opzionali	Description	Tipo di slot
column	Il nome della colonna con il tipo di dati specifica to da ogni campo per i dati descritti da questo nodo di dati. Ex: hostname STRING Per più valori, utilizzare i nomi delle colonne e i tipi di dati separati da uno spazio.	Stringa
escapeChar	Carattere, ad esempio"\", che indica al parser di ignorare il carattere successivo.	Stringa
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa

Campi di sistema	Description	Tipo di slot
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Formato di dati personalizzato

Un formato di dati personalizzato definito da una combinazione di un determinato separatore di colonne, da un separatore di record e da un carattere escape.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "id" : "MyOutputDataType",
  "type" : "Custom",
  "columnSeparator" : ",",
  "recordSeparator" : "\n",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
columnSeparator	Carattere che indica la fine di una colonna in un file di dati.	Stringa

Campi opzionali	Description	Tipo di slot
column	Il nome della colonna con il tipo di dati specifica to da ogni campo per i dati descritti da questo nodo di dati. Ex: hostname STRING Per più valori, utilizzare i nomi delle colonne e i tipi di dati separati da uno spazio.	Stringa
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
recordSeparator	Carattere che indica la fine di una riga in un file di dati, ad esempio "\n". Supporta solo caratteri singoli.	Stringa

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Formato Dynamo DBData

Applica uno schema a una tabella DynamoDB per renderla accessibile tramite una query Hive. `DynamoDBDataFormat`viene utilizzato con un `HiveActivity` oggetto e un `DynamoDBDataNode` input e output. `DynamoDBDataFormat`richiede che tu specifichi tutte le colonne nella tua query Hive. Per una maggiore flessibilità nello specificare determinate colonne in una query Hive o nel supporto di Amazon S3, consulta. Dinamo DBExport DataFormat

> **Note**
>
> I tipi DynamoDB Boolean non sono mappati sui tipi Hive Boolean. Tuttavia, è possibile mappare valori interi DynamoDB pari a 0 o 1 su tipi Hive Boolean.

Esempio

L'esempio seguente mostra come usare `DynamoDBDataFormat` per assegnare uno schema a un input `DynamoDBDataNode`, che consente a un oggetto `HiveActivity` di accedere ai dati in base alle colonne denominate e copiare i dati in un output `DynamoDBDataNode`.

```
{
  "objects": [
    {
      "id" : "Exists.1",
      "name" : "Exists.1",
      "type" : "Exists"
    },
    {
      "id" : "DataFormat.1",
      "name" : "DataFormat.1",
      "type" : "DynamoDBDataFormat",
      "column" : [
         "hash STRING",
        "range STRING"
      ]
    },
    {
      "id" : "DynamoDBDataNode.1",
      "name" : "DynamoDBDataNode.1",
      "type" : "DynamoDBDataNode",
      "tableName" : "$INPUT_TABLE_NAME",
```

```
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.1" }
    },
    {
      "id" : "DynamoDBDataNode.2",
      "name" : "DynamoDBDataNode.2",
      "type" : "DynamoDBDataNode",
      "tableName" : "$OUTPUT_TABLE_NAME",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.1" }
    },
    {
      "id" : "EmrCluster.1",
      "name" : "EmrCluster.1",
      "type" : "EmrCluster",
      "schedule" : { "ref" : "ResourcePeriod" },
      "masterInstanceType" : "m1.small",
      "keyPair" : "$KEYPAIR"
    },
    {
      "id" : "HiveActivity.1",
      "name" : "HiveActivity.1",
      "type" : "HiveActivity",
      "input" : { "ref" : "DynamoDBDataNode.1" },
      "output" : { "ref" : "DynamoDBDataNode.2" },
      "schedule" : { "ref" : "ResourcePeriod" },
      "runsOn" : { "ref" : "EmrCluster.1" },
      "hiveScript" : "insert overwrite table ${output1} select * from ${input1} ;"
    },
    {
      "id" : "ResourcePeriod",
      "name" : "ResourcePeriod",
      "type" : "Schedule",
      "period" : "1 day",
      "startDateTime" : "2012-05-04T00:00:00",
      "endDateTime" : "2012-05-05T00:00:00"
    }
  ]
}
```

Sintassi

Campi opzionali	Description	Tipo di slot
column	Il nome della colonna con il tipo di dati specifica to da ogni campo per i dati descritti da questo nodo di dati. Ad esempio, `hostname STRING`. Per più valori, utilizzare i nomi delle colonne e i tipi di dati separati da uno spazio.	Stringa
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
@version	La versione della pipeline utilizzata per creare l'oggetto.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive l'oggetto con il formato errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Dinamo DBExport DataFormat

Applica uno schema a una tabella DynamoDB per renderla accessibile tramite una query Hive. Utilizzare `DynamoDBExportDataFormat` con un oggetto `HiveCopyActivity` e `DynamoDBDataNode` o con input e output `S3DataNode`. `DynamoDBExportDataFormat` ha i seguenti benefici:

- Fornisce supporto sia per DynamoDB che per Amazon S3
- Consente di filtrare i dati in base a determinate colonne nella query Hive
- Esporta tutti gli attributi da DynamoDB anche se hai uno schema sparso

> **Note**
>
> I tipi DynamoDB Boolean non sono mappati sui tipi Hive Boolean. Tuttavia, è possibile mappare valori interi DynamoDB pari a 0 o 1 su tipi Hive Boolean.

Esempio

L'esempio seguente mostra come usare `HiveCopyActivity` e `DynamoDBExportDataFormat` per copiare i dati da una versione `DynamoDBDataNode` a un'altra, mentre i dati vengono filtrati in base a un timestamp.

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

Sintassi

Campi opzionali	Description	Tipo di slot
column	Il nome della colonna con il tipo di dati specifica to da ogni campo per i dati descritti da questo nodo di dati. Ex: hostname STRING	Stringa
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

RegEx Formato dei dati

Un formato di dati personalizzato definito da un'espressione regolare.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "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"
  ]
}
```

Sintassi

Campi opzionali	Description	Tipo di slot
column	Il nome della colonna con il tipo di dati specifica to da ogni campo per i dati descritti da questo nodo di dati. Ex: hostname STRING Per più valori, utilizzare i nomi delle colonne e i tipi di dati separati da uno spazio.	Stringa
inputRegEx	L'espressione regolare per analizzare un file di input S3. inputRegEx fornisce un modo per recuperare colonne da dati relativamente non strutturati in un file.	Stringa

Campi opzionali	Description	Tipo di slot
outputFormat	I campi delle colonne recuperati da inputRegEx, ma a cui si fa riferimento come %1$s %2$s utilizzando la sintassi del formattatore Java.	Stringa
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» Id "} myBaseObject

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Formato dei dati TSV

Un formato di dati separati da virgola in cui il separatore di colonna è un carattere tab e il separatore di record è un carattere in una nuova riga.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto.

```
{
  "id" : "MyOutputDataType",
  "type" : "TSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintassi

Campi opzionali	Description	Tipo di slot
column	Il nome della colonna e il tipo di dati per i dati descritti da questo nodo di dati. Ad esempio `"Name STRING"` denota una colonna denominata `Name` con campi di tipo dati `STRING`. Separare più coppie di nomi di colonne e tipi di dati con virgole (come illustrato nell'esempio).	Stringa
columnSeparator	Carattere che separa i campi in una colonna dai campi nella colonna successiva. Impostazi one predefinita su '\t'.	Stringa
escapeChar	Carattere, ad esempio"\", che indica al parser di ignorare il carattere successivo.	Stringa
parent	Padre dell'oggetto corrente da cui vengono ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi opzionali	Description	Tipo di slot
recordSeparator	Carattere che separa i record. Impostazione predefinita su '\n'.	Stringa

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Azioni

I seguenti sono gli oggetti AWS Data Pipeline d'azione:

Oggetti

- SnsAlarm
- Interruzione

SnsAlarm

Invia un messaggio di notifica Amazon SNS quando un'attività fallisce o termina con successo.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. I valori per `node.input` e `node.output` provengono dal nodo di dati o da un'attività che fa riferimento a questo oggetto nel relativo campo `onSuccess`.

```
{
  "id" : "SuccessNotify",
  "name" : "SuccessNotify",
  "type" : "SnsAlarm",
  "topicArn" : "arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic",
  "subject" : "COPY SUCCESS: #{node.@scheduledStartTime}",
  "message" : "Files were copied from #{node.input} to #{node.output}."
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
message	Il testo del corpo della notifica Amazon SNS.	Stringa
role	Il ruolo IAM da utilizzare per creare l'allarme Amazon SNS.	Stringa
subject	L'oggetto del messaggio di notifica di Amazon SNS.	Stringa
topicArn	L'ARN dell'argomento Amazon SNS di destinazione per il messaggio.	Stringa

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
nodo	Il nodo per il quale viene eseguita questa azione.	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnabl eObject Id "}
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Interruzione

Un'azione per attivare l'annullamento di un'attività, una risorsa o un nodo di dati in sospeso o incompiuto. AWS Data Pipeline tenta di impostare l'attività, la risorsa o il nodo dati nello stato ANNULLATO se non inizia con il valore. `lateAfterTimeout`

Non è possibile terminare azioni che includono risorse `onSuccess`, `OnFail` o `onLateAction`.

Esempio

Di seguito è illustrato un esempio di questo tipo di oggetto. In questo esempio, il campo `onLateAction` di `MyActivity` contiene un riferimento all'azione `DefaultAction1`. Quando si fornisce un'azione per `onLateAction`, è necessario fornire un valore `lateAfterTimeout` per

indicare il periodo di tempo dall'inizio programmato della pipeline dopo il quale l'attività è considerata in ritardo.

```
{
  "name" : "MyActivity",
  "id" : "DefaultActivity1",
  "schedule" : {
    "ref" : "MySchedule"
  },
  "runsOn" : {
    "ref" : "MyEmrCluster"
  },
  "lateAfterTimeout" : "1 Hours",
  "type" : "EmrActivity",
  "onLateAction" : {
    "ref" : "DefaultAction1"
  },
  "step" : [
    "s3://amzn-s3-demo-bucket/myPath/myStep.jar,firstArg,secondArg",
    "s3://amzn-s3-demo-bucket/myPath/myOtherStep.jar,anotherArg"
  ]
},
{
  "name" : "TerminateTasks",
  "id" : "DefaultAction1",
  "type" : "Terminate"
}
```

Sintassi

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui vengono ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
nodo	Il nodo per il quale viene eseguita questa azione.	Oggetto di riferimento, ad esempio «node»: {"ref»:» myRunnabl eObject Id "}
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Schedule

Definisce la temporizzazione di un evento programmato, ad esempio quando si esegue un'attività.

> **Note**
>
> Quando l'ora di inizio di una pianificazione è AWS Data Pipeline passata, riempie la pipeline e avvia immediatamente le esecuzioni di pianificazione a partire dall'ora di inizio specificata. Per il test/lo sviluppo, utilizzare un intervallo relativamente breve. Altrimenti, AWS Data Pipeline tenta di mettere in coda e pianificare tutte le esecuzioni della pipeline per quell'intervallo. AWS Data Pipeline tenta di prevenire riempimenti accidentali se il componente della pipeline risale a prima di 1 giorno fa `scheduledStartTime` bloccando l'attivazione della pipeline.

Esempi

Di seguito è illustrato un esempio di questo tipo di oggetto. Definisce una pianificazione per ogni ora, a partire da 00:00:00 il 2012-09-01 e finisce alle 00:00:00 il 2012-10-01. Il primo periodo termina alle 01:00:00 il 2012-09-01.

```
{
  "id" : "Hourly",
  "type" : "Schedule",
  "period" : "1 hours",
  "startDateTime" : "2012-09-01T00:00:00",
  "endDateTime" : "2012-10-01T00:00:00"
}
```

La pipeline seguente inizierà al `FIRST_ACTIVATION_DATE_TIME` e verrà eseguita ogni ora fino alle 22:00:00 il 2014-04-25.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "endDateTime": "2014-04-25T22:00:00"
   }
```

La pipeline seguente inizierà al `FIRST_ACTIVATION_DATE_TIME` e verrà eseguita ogni ora e completata dopo tre occorrenze.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

La seguente pipeline inizierà alle 22:00:00 il 2014-04-25, verrà eseguita ogni ora e terminerà dopo tre occorrenze.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startDateTime": "2014-04-25T22:00:00",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

On-demand utilizzando l'oggetto predefinito

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
}
```

On-demand con un oggetto di pianificazione esplicita

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
},
{
  "name": "DefaultSchedule",
  "type": "Schedule",
  "id": "DefaultSchedule",
  "period": "ONDEMAND_PERIOD",
  "startAt": "ONDEMAND_ACTIVATION_TIME"
},
```

I seguenti esempi illustrano come una pianificazione può essere ereditata dall'oggetto predefinito, essere esplicitamente impostata per l'oggetto o derivare da un riferimento padre:

Pianificazione ereditata da un oggetto predefinito

```
{
  "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!'"
    }
  ]
}
```

Pianificazione esplicita sull'oggetto

```
{
  "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!'"
    }
  ]
}
```

Pianificazione da un riferimento padre

```
{
  "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!'"
    }
  ]
}
```

Sintassi

Campi obbligatori	Description	Tipo di slot
punto	Frequenza di esecuzione della pipeline. Il formato è "N [minuti|ore|giorni|settimane|mesi]", dove N è un numero seguito da uno degli specificatori di tempo. Ad esempio, "15 minuti" esegue la pipeline ogni 15 minuti. La durata	Periodo

Campi obbligatori	Description	Tipo di slot
	minima è pari a 15 minuti, mentre la durata massima è di 3 anni.	

Gruppo richiesto (uno dei seguenti è obbligatorio)	Description	Tipo di slot
startAt	La data e l'ora in cui avviare l'esecuzione della pipeline programmata. Il valore valido è FIRST_ACTIVATION_DATE_TIME, che viene sostituito dalla creazione di una pipeline on demand.	Enumerazione
startDateTime	La data e l'ora di avvio delle esecuzioni programmate. È necessario utilizzare uno startDateTime o startAt ma non entrambi.	DateTime

Campi opzionali	Description	Tipo di slot
endDateTime	La data e l'ora in cui terminare le esecuzion i programmate. Deve essere una data e un'ora successive al valore di startDateTime o startAt. Il comportamento predefinito è quello di pianificare l'esecuzione finché la pipeline non viene terminata.	DateTime
occorrenze	Il numero di volte in cui eseguire la pipeline quando viene attivata. Non è possibile utilizzare le occorrenze con. endDateTime	Numero intero
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»:

Campi opzionali	Description	Tipo di slot
		{"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@firstActivationTime	L'ora della creazione di oggetti.	DateTime
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Utilità

I seguenti oggetti della utility configurano altri oggetti della pipeline:

Argomenti

- ShellScriptConfig
- EmrConfiguration
- Proprietà

ShellScriptConfig

Da utilizzare con un'attività per eseguire uno script di shell per preActivityTask Config e postActivityTask Config. Questo oggetto è disponibile per HadoopActivity, HiveActivityHiveCopyActivity, e. PigActivity Specificare un URI S3 e un elenco di argomenti per lo script.

Esempio

A ShellScriptConfig con argomenti:

```
{
   "id" : "ShellScriptConfig_1”,
   "name" : “prescript”,
   "type" : "ShellScriptConfig",
   "scriptUri": “s3://my-bucket/shell-cleanup.sh”,
   "scriptArgument" : ["arg1","arg2"]
 }
```

Sintassi

Questo oggetto include i campi seguenti.

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui vengono ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
scriptArgument	Un elenco di argomenti da utilizzare con lo script della shell.	Stringa
scriptUri	L'URI dello script in Amazon S3 che deve essere scaricato ed eseguito.	Stringa

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

EmrConfiguration

L' EmrConfiguration oggetto è la configurazione utilizzata per i cluster EMR con versioni 4.0.0 o successive. Le configurazioni (sotto forma di elenco) sono un parametro per la chiamata API. RunJobFlow L'API di configurazione per Amazon EMR richiede una classificazione e proprietà. AWS Data Pipeline utilizza EmrConfiguration con gli oggetti Property corrispondenti per configurare un'EmrClusterapplicazione come Hadoop, Hive, Spark o Pig su cluster EMR avviati in un'esecuzione di pipeline. Poiché la configurazione può essere modificata solo per i nuovi cluster, non è possibile fornire un oggetto per le risorse esistenti. EmrConfiguration Per ulteriori informazioni, consulta https://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/.

Esempio

Il seguente oggetto di configurazione imposta le proprietà `io.file.buffer.size` e `fs.s3.block.size` in `core-site.xml`:

```
[
   {
      "classification":"core-site",
```

```
      "properties":
      {
         "io.file.buffer.size": "4096",
         "fs.s3.block.size": "67108864"
      }
   }
]
```

La definizione dell'oggetto pipeline corrispondente utilizza un EmrConfiguration oggetto e un elenco di oggetti Property nel `property` campo:

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      },
      {
        "ref": "fs-s3-block-size"
      }
      ]
    },
    {
      "name": "io-file-buffer-size",
      "id": "io-file-buffer-size",
      "type": "Property",
      "key": "io.file.buffer.size",
      "value": "4096"
    },
```

```
    {
      "name": "fs-s3-block-size",
      "id": "fs-s3-block-size",
      "type": "Property",
      "key": "fs.s3.block.size",
      "value": "67108864"
    }
  ]
}
```

L'esempio seguente è una configurazione nidificata utilizzata per impostare l'ambiente Hadoop con la classificazione `hadoop-env`:

```
[
  {
    "classification": "hadoop-env",
    "properties": {},
    "configurations": [
      {
        "classification": "export",
        "properties": {
          "YARN_PROXYSERVER_HEAPSIZE": "2396"
        }
      }
    ]
  }
]
```

L'oggetto di definizione corrispondente della pipeline che utilizza questa configurazione è il seguente:

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

L'esempio seguente modifica una proprietà specifica di Hive per un cluster EMR:

```
{
    "objects": [
        {
            "name": "hivesite",
            "id": "hivesite",
            "type": "EmrConfiguration",
            "classification": "hive-site",
            "property": [
                {
                    "ref": "hive-client-timeout"
                }
            ]
        },
        {
```

```
            "name": "hive-client-timeout",
            "id": "hive-client-timeout",
            "type": "Property",
            "key": "hive.metastore.client.socket.timeout",
            "value": "2400s"
        }
    ]
}
```

Sintassi

Questo oggetto include i campi seguenti.

Campi obbligatori	Description	Tipo di slot
classificazione	Classificazione della configurazione.	Stringa

Campi opzionali	Description	Tipo di slot
configurazione	Sottoconfigurazione per questa configurazione.	Oggetto di riferimento, ad esempio «configur ation»: {"ref»:» Id "} myEmrConfiguration
parent	Padre dell'oggetto corrente da cui saranno ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}
property	Proprietà di configurazione.	Oggetto di riferimento, ad esempio «property »: {"ref»:» myPropert yId «}

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato	Stringa
@pipelineId	L'id della pipeline a cui appartiene questo oggetto	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects	Stringa

Vedi anche

- EmrCluster
- Proprietà
- Guida al rilascio di Amazon EMR

Proprietà

Una singola proprietà chiave-valore da utilizzare con un EmrConfiguration oggetto.

Esempio

La seguente definizione della pipeline mostra un EmrConfiguration oggetto e gli oggetti Property corrispondenti per lanciare un: 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"
    }
  ]
}
```

Sintassi

Questo oggetto include i campi seguenti.

Campi obbligatori	Description	Tipo di slot
Chiave	key	Stringa
value	value	Stringa

Campi opzionali	Description	Tipo di slot
parent	Padre dell'oggetto corrente da cui vengono ereditati gli slot.	Oggetto di riferimento, ad esempio «parent»: {"ref»:» myBaseObj ect Id "}

Campi Runtime	Description	Tipo di slot
@version	Versione della pipeline con cui l'oggetto è stato creato.	Stringa

Campi di sistema	Description	Tipo di slot
@error	Errore che descrive il formato oggetto errato.	Stringa
@pipelineId	L'ID della pipeline a cui appartiene questo oggetto.	Stringa
@sphere	La sfera di un oggetto indica la propria posizione nel ciclo di vita: i Component Objects generano Instance Objects che eseguono Attempt Objects.	Stringa

Vedi anche

- EmrCluster
- EmrConfiguration
- Guida al rilascio di Amazon EMR

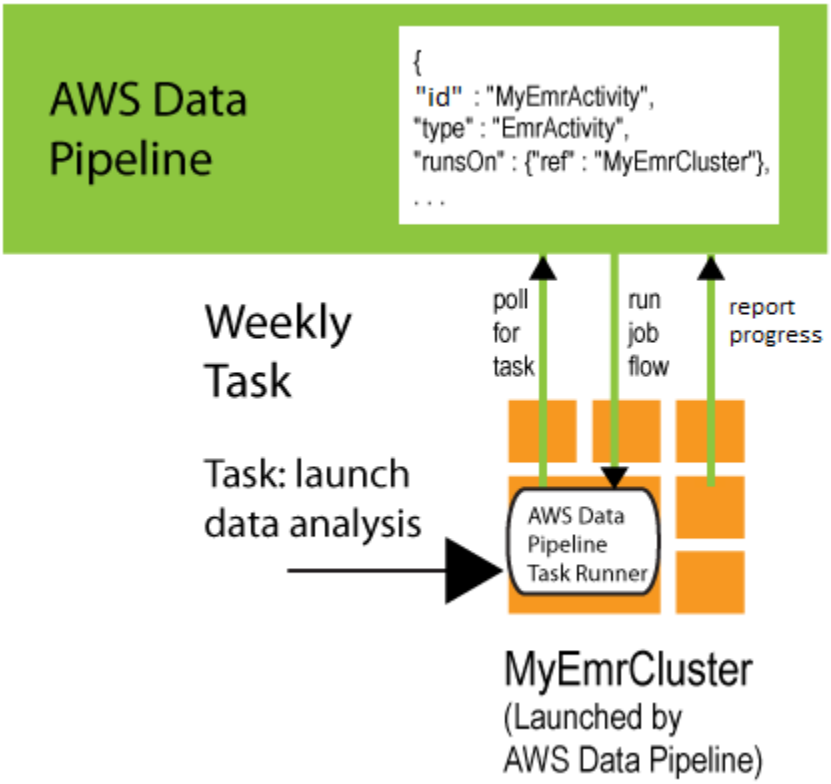
Lavorare con Task Runner

Task Runner è un'applicazione task agent che analizza AWS Data Pipeline le attività pianificate e le esegue su istanze Amazon EC2, cluster Amazon EMR o altre risorse di calcolo, segnalando lo stato in corso. A seconda dell'applicazione, è possibile scegliere di:

- Consenti di installare e gestire AWS Data Pipeline una o più applicazioni Task Runner per te. Quando viene attivata una pipeline, viene creato automaticamente il valore predefinito `Ec2Instance` o `EmrCluster` l'oggetto a cui fa riferimento un campo RunSon di attività. AWS Data Pipeline si occupa dell'installazione di Task Runner su un'istanza EC2 o sul nodo master di un cluster EMR. In questo modello, AWS Data Pipeline può occuparsi della maggior parte della gestione dell'istanza o del cluster per te.
- Eseguire tutta o parte di una pipeline su risorse gestite dall'utente. Le risorse potenziali includono un'istanza Amazon EC2 a lunga durata, un cluster Amazon EMR o un server fisico. È possibile installare un task runner (che può essere Task Runner o un task agent personalizzato del proprio dispositivo) quasi ovunque, a condizione che sia in grado di comunicare con il servizio Web. AWS Data Pipeline In questo modello, si assume il controllo quasi completo su quali risorse vengono utilizzate e su come vengono gestite, ed è necessario installare e configurare manualmente Task Runner. Per eseguire questa operazione, utilizzare le procedure di questa sezione, come descritto in Esecuzione del lavoro su risorse esistenti utilizzando Task Runner.

Task Runner su AWS Data Pipeline-Managed Resources

Quando una risorsa viene avviata e gestita da AWS Data Pipeline, il servizio Web installa automaticamente Task Runner su tale risorsa per elaborare le attività nella pipeline. È necessario specificare una risorsa di calcolo (un'istanza Amazon EC2 o un cluster Amazon EMR) per `runs0n` il campo di un oggetto di attività. Quando AWS Data Pipeline avvia questa risorsa, installa Task Runner su quella risorsa e la configura per elaborare tutti gli oggetti di attività il cui campo è impostato su quella risorsa. `runs0n` Quando AWS Data Pipeline termina la risorsa, i log di Task Runner vengono pubblicati in una posizione Amazon S3 prima della chiusura.



Ad esempio, se si utilizza la `EmrActivity` in una pipeline e si specifica una risorsa `EmrCluster` nel campo `runsOn`. Quando AWS Data Pipeline elabora tale attività, avvia un cluster Amazon EMR e installa Task Runner sul nodo master. Questo Task Runner elabora quindi le attività per le attività il cui `runsOn` campo è impostato su quell'oggetto. `EmrCluster` Il seguente estratto da una definizione di pipeline mostra questa relazione tra due oggetti.

```
{
  "id" : "MyEmrActivity",
  "name" : "Work to perform on my data",
  "type" : "EmrActivity",
  "runsOn" : {"ref" : "MyEmrCluster"},
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : "s3://amzn-s3-demo-bucket/myPath/myStep.jar,firstArg,secondArg",
  "step" : "s3://amzn-s3-demo-bucket/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"
}
```

Per informazioni ed esempi di esecuzione di questa attività, vedi EmrActivity.

Se in una pipeline sono presenti più risorse AWS Data Pipeline gestite, Task Runner viene installato su ognuna di esse e tutte controllano le attività da elaborare AWS Data Pipeline .

Esecuzione del lavoro su risorse esistenti utilizzando Task Runner

Puoi installare Task Runner su risorse di calcolo che gestisci, come un'istanza Amazon EC2 o un server o una workstation fisica. Task Runner può essere installato ovunque, su qualsiasi hardware o sistema operativo compatibile, a condizione che possa comunicare con il servizio web. AWS Data Pipeline

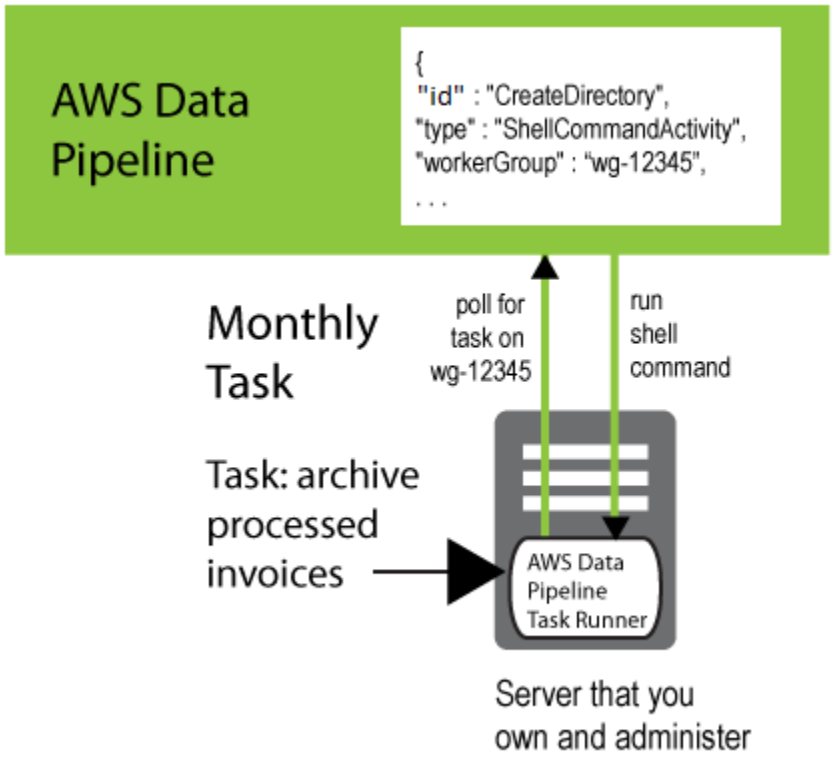
Questo approccio può essere utile quando, ad esempio, si desidera utilizzare per AWS Data Pipeline elaborare i dati archiviati all'interno del firewall dell'organizzazione. Installando Task Runner su un server della rete locale, è possibile accedere al database locale in modo sicuro e quindi eseguire il polling AWS Data Pipeline per l'operazione successiva da eseguire. Quando AWS Data Pipeline termina l'elaborazione o elimina la pipeline, l'istanza di Task Runner rimane in esecuzione sulla risorsa di calcolo fino a quando non viene chiusa manualmente. I log di Task Runner persistono dopo il completamento dell'esecuzione della pipeline.

Per utilizzare Task Runner su una risorsa gestita dall'utente, è necessario innanzitutto scaricare Task Runner e quindi installarlo sulla risorsa di calcolo utilizzando le procedure descritte in questa sezione.

> **Note**
>
> Puoi installare Task Runner solo su Linux, UNIX o macOS. Task Runner non è supportato nel sistema operativo Windows.

Per utilizzare Task Runner 2.0, la versione minima di Java richiesta è 1.7.

Per connettere un Task Runner che avete installato alle attività della pipeline che deve elaborare, aggiungete un `workerGroup` campo all'oggetto e configurate Task Runner per verificare il valore del gruppo di lavoro. Puoi farlo passando la stringa del gruppo di lavoro come parametro (ad esempio, `--workerGroup=wg-12345`) quando esegui il file JAR di Task Runner.




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

Installazione di Task Runner

Questa sezione spiega come installare e configurare Task Runner e i relativi prerequisiti. L'installazione è un semplice processo manuale.

Per installare Task Runner

1. Task Runner richiede le versioni Java 1.6 o 1.8. Per determinare se Java è installato e la versione in esecuzione, utilizzare il comando seguente:

   ```
   java -version
   ```

   Se sul computer non è installato Java 1.6 o 1.8, scaricate una di queste versioni da http://www.oracle. com/technetwork/java/index.html. Scaricare e installare Java, quindi procedere con il passaggio successivo.
2. Scarica `TaskRunner-1.0.jar` da https://s3.amazonaws.com/datapipeline-us-east-1/us-east-1/software/latest/TaskRunner/TaskRunner -1.0.jar e poi copialo in una cartella sulla risorsa di calcolo di destinazione. Per i cluster Amazon EMR che eseguono `EmrActivity` attività, installa Task Runner sul nodo master del cluster.
3. Quando utilizzano Task Runner per connettersi al servizio AWS Data Pipeline Web per elaborare i comandi, gli utenti devono accedere programmaticamente a un ruolo che dispone delle autorizzazioni per creare o gestire pipeline di dati. Per ulteriori informazioni, consulta Concessione dell'accesso programmatico.
4. Task Runner si connette al servizio Web tramite HTTPS AWS Data Pipeline . Se utilizzi una AWS risorsa, assicurati che HTTPS sia abilitato nella tabella di routing e nell'ACL di sottorete appropriati. Se si sta usando un firewall o un proxy, assicurarsi che la porta 443 sia aperta.

(Facoltativo) Concessione dell'accesso a Task Runner ad Amazon RDS

Amazon RDS ti consente di controllare l'accesso alle tue istanze DB utilizzando gruppi di sicurezza del database (gruppi di sicurezza DB). Un gruppo di sicurezza DB si comporta come un firewall, controllando l'accesso di rete all'istanza database. Per impostazione predefinita, l'accesso alla rete è disattivato per le istanze database. È necessario modificare i gruppi di sicurezza del database per consentire a Task Runner di accedere alle istanze Amazon RDS. Task Runner ottiene l'accesso ad Amazon RDS dall'istanza su cui viene eseguita, quindi gli account e i gruppi di sicurezza che aggiungi all'istanza Amazon RDS dipendono da dove installi Task Runner.

Per concedere l'accesso a Task Runner in EC2-Classic

1. Apri la console Amazon RDS.
2. Nel riquadro di navigazione scegliere Instances (Istanze) e quindi selezionare l'istanza database.

3. In Security and Network (Sicurezza e Network), selezionare il gruppo di sicurezza che apre la pagina relativa ai Security Groups (Gruppi di sicurezza) con questo gruppo di sicurezza di database selezionato. Selezionare l'icona dei dettagli per il gruppo di sicurezza DB.
4. In Security Group Details (Dettagli gruppo di sicurezza), creare una regola con il Connection Type (Tipo di connessione) e i Details (Dettagli) appropriati. Questi campi dipendono dalla posizione in cui è in esecuzione Task Runner, come descritto di seguito:

   - `Ec2Resource`
     - Connection Type (Tipo di connessione): `EC2 Security Group`

       Dettagli: *`my-security-group-name`* (il nome del gruppo di sicurezza che hai creato per l'istanza EC2)
   - `EmrResource`
     - Connection Type (Tipo di connessione): `EC2 Security Group`

       Details (Dettagli): `ElasticMapReduce-master`
     - Connection Type (Tipo di connessione): `EC2 Security Group`

       Details (Dettagli): `ElasticMapReduce-slave`
   - Ambiente locale (in locale)
     - Connection Type (Tipo di connessione): `CIDR/IP`:

       Dettagli: *`my-ip-address`* (l'indirizzo IP del computer o l'intervallo di indirizzi IP della rete, se il computer è protetto da un firewall)
5. Fare clic su Add (Aggiungi).

Per concedere l'accesso a Task Runner in EC2-VPC

1. Apri la console Amazon RDS.
2. Nel riquadro di navigazione, scegliere Instances (Istanze).
3. Selezionare l'icona dei dettagli per l'istanza database. In Sicurezza e rete, apri il link al gruppo di sicurezza, che ti porta alla console Amazon EC2. Se si utilizza il vecchio progetto della console per i gruppi di sicurezza, passare al nuovo progetto della console selezionando l'icona visualizzata nella parte superiore della pagina della console.

4. Nella scheda Inbound (In entrata), scegli Edit (Modifica), Add Rule (Aggiungi regola). Specificare la porta del database utilizzata quando è stata avviata l'istanza database. L'origine dipende da dove è in esecuzione Task Runner, come descritto qui:

   - `Ec2Resource`
     - *`my-security-group-id`*(l'ID del gruppo di sicurezza che hai creato per l'istanza EC2)
   - `EmrResource`
     - *`master-security-group-id`*(l'ID del gruppo di `ElasticMapReduce-master` sicurezza)
     - *`slave-security-group-id`*(l'ID del gruppo `ElasticMapReduce-slave` di sicurezza)
   - Ambiente locale (in locale)
     - *`ip-address`*(l'indirizzo IP del computer o l'intervallo di indirizzi IP della rete, se il computer è protetto da un firewall)

5. Fai clic su Salva.

Avvio di Task Runner

In una nuova finestra del prompt dei comandi impostata sulla directory in cui è installato Task Runner, avvia Task Runner con il comando seguente.

```
java -jar TaskRunner-1.0.jar --config ~/credentials.json --workerGroup=myWorkerGroup --
region=MyRegion --logUri=s3://amzn-s3-demo-bucket/foldername
```

L'opzione `--config` punta al file delle credenziali.

L'opzione `--workerGroup` specifica il nome del gruppo di lavoratori, che deve essere lo stesso valore specificato nella pipeline per le attività da elaborare.

L'opzione `--region` specifica la regione del servizio da cui prendere le operazioni da eseguire.

L'`--logUri`opzione viene utilizzata per inviare i log compressi in una posizione in Amazon S3.

Quando Task Runner è attivo, stampa il percorso in cui vengono scritti i file di registro nella finestra del terminale. Di seguito è riportato un esempio di :

```
Logging to /Computer_Name/.../output/logs
```

Task Runner deve essere eseguito non collegato alla shell di login. Se si sta usando un'applicazione terminale per connettersi al computer, potrebbe essere necessario utilizzare una utility come nohup o schermo per evitare di uscire dall'applicazione Task Runner al momento della disconnessione. Per ulteriori informazioni sulle opzioni delle righe di comando, consulta Opzioni di configurazione di Task Runner.

Verifica della registrazione di Task Runner

Il modo più semplice per verificare che Task Runner funzioni è verificare se sta scrivendo file di registro. Task Runner scrive i file di registro ogni ora nella directory`output/logs`, nella directory in cui è installato Task Runner. Il nome del file è `Task Runner.log.YYYY-MM-DD-HH`, dove HH viene eseguito da mezzanotte alle 23:00, in UDT. Per risparmiare spazio di archiviazione, tutti i file di registro più vecchi di otto ore vengono compressi con. GZip

Thread e precondizioni di Task Runner

Task Runner utilizza un pool di thread per ciascuna attività, attività e condizione preliminare. L'impostazione predefinita per `--tasks` è 2, il che significa che ci sono due thread assegnati dal pool di attività e ogni thread interroga il servizio per individuare nuove attività. AWS Data Pipeline Pertanto, `--tasks` è un attributo di ottimizzazione delle prestazioni che può essere utilizzato per ottimizzare il throughput della pipeline.

La logica di ripetizione dei tentativi della pipeline per le precondizioni si verifica in Task Runner. Due thread di precondizione vengono assegnati al sondaggio per gli oggetti di precondizione. AWS Data Pipeline Task Runner rispetta i campi retryDelay e preconditionTimeout dell'oggetto di precondizione definiti in base alle precondizioni.

In molti casi, diminuendo il timeout del polling della precondizione e il numero di nuovi tentativi è possibile migliorare le prestazioni dell'applicazione. Analogamente, le applicazioni con precondizioni di lunga durata potrebbero avere bisogno di un aumento dei valori relativi a timeout e nuovi tentativi. Per ulteriori informazioni sugli oggetti delle precondizioni, consulta Precondizioni.

Opzioni di configurazione di Task Runner

Queste sono le opzioni di configurazione disponibili dalla riga di comando all'avvio di Task Runner.

Parametri della riga di comando	Description
`--help`	Informazioni di aiuto della riga di comando. Ad esempio: `Java -jar TaskRunner-1.0.jar --help`
`--config`	Percorso e nome del file `credentials.json` .
`--accessId`	L'ID della chiave di AWS accesso che Task Runner può utilizzare per effettuare richieste. Le `--secretKey` opzioni `--accessID` and forniscono un'alternativa all'utilizzo di un file credentials.json. Se viene fornito anche un file `credentials.json` , le opzioni `--accessID` e `--secretKey` hanno la precedenza.
`--secretKey`	La chiave AWS segreta che Task Runner può utilizzare per effettuare richieste. Per ulteriori informazioni, consulta `--accessID` .
`--endpoint`	Un endpoint è un URL che rappresenta il punto di partenza per un servizio Web. L'endpoint del AWS Data Pipeline servizio nella regione in cui si effettuano le richieste. Opzionale. In generale, è sufficiente specificare una regione e non è necessario impostare l'endpoint. Per un elenco di AWS Data Pipeline regioni ed endpoint, consulta AWS Data Pipeline Regions and Endpoints nel. Riferimenti generali di AWS
`--workerGroup`	Il nome del gruppo di lavoro per il quale Task Runner recupera il lavoro. Obbligatorio. Quando Task Runner esegue il polling del servizio Web, utilizza le credenziali fornite e il valore di `workerGroup` per seleziona

Parametri della riga di comando	Description
	re quali (se presenti) attività recuperare. È possibile utilizzare qualsiasi nome significativo per l'utente; l'unico requisito è che la stringa corrisponda tra Task Runner e le attività della pipeline corrispondenti. Il nome del gruppo di lavoratori è associato a una regione. Anche se esistono nomi di gruppi di lavoro identici in altre regioni, Task Runner ottiene sempre le attività dalla regione specificata in. `--region`
`--taskrunnerId`	L'ID del runner dell'attività da utilizzare per segnalare l'avanzamento. Opzionale.
`--output`	La directory Task Runner per i file di output dei log. Opzionale. I file di log vengono archiviati in una directory locale fino a quando non vengono inviati ad Amazon S3. Questa opzione sostituis ce la directory di default.
`--region`	La regione da utilizzare. L'impostazione della regione è facoltativa, ma è sempre consiglia ta. Se non si specifica la regione, Task Runner recupera le attività dalla regione di servizio predefinita,. `us-east-1` Altre regioni supportate sono: `eu-west-1` , `ap-northeast-1` , `ap-southeast-2` , `us-west-2` .
`--logUri`	Il percorso di destinazione di Amazon S3 per Task Runner su cui eseguire il backup dei file di registro ogni ora. Quando Task Runner termina, i log attivi nella directory locale vengono inviati alla cartella di destinazione di Amazon S3.

Parametri della riga di comando	Description
`--proxyHost`	L'host del proxy utilizzato dai client Task Runner per connettersi ai servizi AWS.
`--proxyPort`	Porta dell'host proxy utilizzata dai client Task Runner per connettersi ai servizi AWS.
`--proxyUsername`	Il nome utente per il proxy.
`--proxyPassword`	La password per il proxy.
`--proxyDomain`	Il nome del dominio di Windows per il proxy NTLM.
`--proxyWorkstation`	Il nome della workstation di Windows per il proxy NTLM.

Utilizzo di Task Runner con un proxy

Se si sta usando un host proxy, è possibile specificarne la configurazione quando si richiama Task Runner o si imposta la variabile di ambiente, HTTPS_PROXY. La variabile di ambiente usata con Task Runner accetta la stessa configurazione utilizzata per l'interfaccia a riga di comando di AWS.

Task Runner e Custom AMIs

Quando specifichi un `Ec2Resource` oggetto per la tua pipeline, AWS Data Pipeline crea un'istanza EC2 per te, utilizzando un'AMI che installa e configura Task Runner per te. Un tipo di istanza compatibile con PV è obbligatorio in questo caso. In alternativa, puoi creare un'AMI personalizzata con Task Runner e quindi specificare l'ID di questa AMI utilizzando il `imageId` campo dell'`Ec2Resource`oggetto. Per ulteriori informazioni, consulta Ec2Resource.

Un'AMI personalizzata deve soddisfare i seguenti requisiti per poterla AWS Data Pipeline utilizzare correttamente per Task Runner:

- Creazione dell'AMI nella stessa regione in cui saranno eseguite le istanze. Per ulteriori informazioni, consulta Creating Your Own AMI nella Guida per l'utente di Amazon EC2.

- Assicurarsi che il tipo di virtualizzazione dell'AMI sia supportato dal tipo di istanza che si intende utilizzare. Ad esempio, i tipi di istanze I2 e G2 richiedono un'AMI HVM e T1, C1, M1 e i tipi di istanza M2 richiedono un'AMI PV. Per ulteriori informazioni, consulta la sezione Tipi di virtualizzazione delle AMI Linux nella Guida per l'utente di Amazon EC2.
- Installare il seguente software:
  - Linux
  - Bash
  - wget
  - unzip
  - Java 1.6 o 1.8
  - cloud-init
- Crea e configura un utente denominato. `ec2-user`

Risoluzione dei problemi

Quando si verifica un problema con AWS Data Pipeline, il sintomo più comune è il mancato funzionamento di una pipeline. È possibile utilizzare i dati che la console e l'interfaccia a riga di comando forniscono per individuare il problema e trovare una soluzione.

Indice

Individuazione di errori nelle pipeline

La AWS Data Pipeline console è uno strumento utile per monitorare visivamente lo stato delle pipeline e individuare facilmente eventuali errori relativi a esecuzioni di pipeline non riuscite o incomplete.

Individuare gli errori su esecuzioni non riuscite o incomplete con la console

1. Nella pagina List Pipelines (Elenca pipeline), se la colonna Status (Stato) di una delle istanze della pipeline mostra uno stato diverso da FINISHED (FINITO), la pipeline sta aspettando una precondizione da soddisfare o ha fallito ed è necessario intervenire per risolvere il problema.
2. Nella pagina List Pipelines (Elenca pipeline), individuare la pipeline dell'istanza e selezionare il triangolo a sinistra, per espandere i dettagli.
3. Alla fine di questo pannello, scegliere View execution details (Visualizza dettagli esecuzione); si pare il pannello Instance summary (Riepilogo istanza) per mostrare i dettagli dell'istanza selezionata.
4. Nel pannello Instance summary (Riepilogo istanza), selezionare il triangolo accanto all'istanza per visualizzare ulteriori dettagli dell'istanza e scegliere Details (Dettagli), More... (Ulteriori informazioni...) Se lo stato della tua istanza selezionata è FAILED (NON RIUSCITO), la casella dettagli dispone di voci per il messaggio di errore, il `errorStackTrace` e altre informazioni. È possibile salvare le informazioni in un file. Scegli OK.

5. Nel riquadro Instance summary (Riepilogo istanze, scegliere Attempts (Tentativi) per vedere i dettagli per ogni riga di tentativo.
6. Per intervenire sull'istanza non completa o non riuscita, selezionare la casella di controllo accanto all'istanza. In questo modo vengono attivate le operazioni. Quindi, selezionare un'operazione (`Rerun|Cancel|Mark Finished`).

Identificazione del cluster Amazon EMR che serve la tua pipeline

Se un `EMRCluster` OR `EMRActivity` fallisce e le informazioni sull'errore fornite dalla AWS Data Pipeline console non sono chiare, puoi identificare il cluster Amazon EMR che serve la tua pipeline utilizzando la console Amazon EMR. Questo ti aiuta a individuare i log forniti da Amazon EMR per ottenere maggiori dettagli sugli errori che si verificano.

Per visualizzare informazioni più dettagliate sugli errori di Amazon EMR

1. Nella AWS Data Pipeline console, seleziona il triangolo accanto all'istanza della pipeline per espandere i dettagli dell'istanza.
2. Scegliere View execution details (Visualizza dettagli esecuzione) e selezionare il triangolo accanto al componente.
3. Nella colonna Details (Dettagli), scegliere More... (Altro...). Si apre la schermata di informazioni che fornisce un elenco di dettagli del componente. Individuare e copiare il valore instanceParent dallo schermo, ad esempio: `@EmrActivityId_xiFDD_2017-09-30T21:40:13`
4. Accedi alla console Amazon EMR, cerca un cluster con il valore InstanceParent corrispondente nel nome, quindi scegli Debug.

   > **Note**
   >
   > Affinché il pulsante Debug funzioni, la definizione della pipeline deve aver impostato l' EmrActivity `enableDebugging`opzione su `true` e l'`EmrLogUri`opzione su un percorso valido.

5. Ora che sai quale cluster Amazon EMR contiene l'errore che causa il fallimento della pipeline, segui i suggerimenti per la risoluzione dei problemi nella Amazon EMR Developer Guide.

Interpretazione dei dettagli sullo stato della pipeline

I vari livelli di stato visualizzati nella AWS Data Pipeline console e nella CLI indicano le condizioni di una pipeline e dei suoi componenti. Lo stato della pipeline è semplicemente una panoramica di una pipeline; per visualizzare ulteriori informazioni, visualizzare lo stato dei singoli componenti della pipeline. È possibile eseguire questa operazione facendo clic sulla pipeline nella console o recuperando i dettagli del componente della pipeline tramite l'interfaccia a riga di comando.

Codici di stato

`ACTIVATING`

Il componente o la risorsa è in fase di avvio, ad esempio un'istanza EC2.

`CANCELED`

Il componente è stato annullato da un utente o AWS Data Pipeline prima che potesse essere eseguito. Ciò può avvenire automaticamente quando si verifica un errore in un altro componente o risorsa da cui dipende questo componente.

`CASCADE_FAILED`

Il componente o la risorsa è stato annullato a causa di un errore a cascata dovuto a una delle sue dipendenze, ma probabilmente non era il componente all'origine dell'errore.

`DEACTIVATING`

La pipeline viene disattivata.

`FAILED`

Il componente o la risorsa ha riscontrato un errore e ha smesso di funzionare. Quando un componente o una risorsa si guasta, possono verificarsi annullamenti ed errori che si ripercuotono su altri componenti che dipendono da esso.

`FINISHED`

Il componente ha completato il lavoro assegnato.

`INACTIVE`

La pipeline è stata disattivata.

`PAUSED`

Il componente è stato messo in pausa e attualmente non sta funzionando.

`PENDING`

La pipeline è pronta per essere attivata per la prima volta.

`RUNNING`

La risorsa è in esecuzione e pronta per ricevere lavoro.

`SCHEDULED`

L'esecuzione della risorsa è pianificata.

`SHUTTING_DOWN`

La risorsa si spegne dopo aver completato con successo il suo lavoro.

`SKIPPED`

Il componente ha saltato gli intervalli di esecuzione dopo l'attivazione della pipeline utilizzando un timestamp successivo alla pianificazione corrente.

`TIMEDOUT`

La risorsa ha superato la `terminateAfter` soglia ed è stata interrotta da. AWS Data Pipeline Dopo che la risorsa ha raggiunto questo stato, AWS Data Pipeline ignora i `actionOnResourceFailure` `retryTimeout` valori e e per quella risorsa. `retryDelay` Questo stato si applica solo alle risorse.

`VALIDATING`

La definizione della pipeline viene convalidata da. AWS Data Pipeline

`WAITING_FOR_RUNNER`

Il componente è in attesa che il suo client di lavoro recuperi un elemento di lavoro. La relazione tra componente e cliente-lavoratore è controllata dai `workerGroup` campi `runsOn` o dai campi definiti da quel componente.

`WAITING_ON_DEPENDENCIES`

Il componente sta verificando che le precondizioni predefinite e configurate dall'utente siano soddisfatte prima di eseguire il suo lavoro.

Individuazione dei log di errore

Questa sezione spiega come trovare i vari registri di AWS Data Pipeline scrittura, che è possibile utilizzare per determinare l'origine di determinati errori ed errori.

Log della pipeline

Ti consigliamo di configurare le pipeline per creare file di registro in una posizione persistente, come nell'esempio seguente in cui utilizzi il `pipelineLogUri` campo sull'`Default`oggetto di una pipeline per fare in modo che tutti i componenti della pipeline utilizzino una posizione di registro Amazon S3 per impostazione predefinita (puoi sovrascrivere questa impostazione configurando una posizione di registro in un componente specifico della pipeline).

> **Note**
>
> Per impostazione predefinita, Task Runner archivia i registri in una posizione diversa, che potrebbe non essere disponibile al termine della pipeline e all'interruzione dell'istanza che esegue Task Runner. Per ulteriori informazioni, consulta Verifica della registrazione di Task Runner.

Per configurare la posizione del registro utilizzando la AWS Data Pipeline CLI in un file JSON della pipeline, inizia il file della pipeline con il testo seguente:

```
{ "objects": [
{
  "id":"Default",
  "pipelineLogUri":"s3://amzn-s3-demo-bucket/error_logs"
},
...
```

Dopo aver configurato una directory di log della pipeline, Task Runner crea una copia dei log nella directory, con la stessa formattazione e gli stessi nomi di file descritti nella sezione precedente sui log di Task Runner.

Registri dei passaggi di Hadoop Job e Amazon EMR

Con qualsiasi attività basata su Hadoop, ad esempio HadoopActivityHiveActivity, è PigActivity possibile visualizzare i job log di Hadoop nella posizione restituita nello slot di runtime,. hadoopJobLog EmrActivitydispone di funzionalità di registrazione proprie e tali log vengono archiviati utilizzando la posizione scelta da Amazon EMR e restituita dallo slot di runtime,. emrStepLog Per ulteriori informazioni, consulta View Log Files nella Amazon EMR Developer Guide.

Risoluzione dei problemi più comuni

In questo argomento vengono descritti vari sintomi AWS Data Pipeline relativi ai problemi e i passaggi consigliati per risolverli.

Indice

Pipeline bloccata in stato Pending

Una pipeline che sembra bloccata in stato PENDING indica che una pipeline non è stata ancora attivata o l'attivazione non è riuscita a causa di un errore nella definizione della pipeline. Assicurati di non aver ricevuto errori quando hai inviato la pipeline utilizzando la AWS Data Pipeline CLI o quando hai tentato di salvare o attivare la pipeline utilizzando la console. AWS Data Pipeline Inoltre, controllare che la pipeline abbia una definizione valida.

Per visualizzare la definizione della pipeline sullo schermo utilizzando l'interfaccia a riga di comando:

```
aws datapipeline --get-pipeline-definition --pipeline-id df-EXAMPLE_PIPELINE_ID
```

Verificare che la definizione della pipeline sia completa, controllare le parentesi di chiusura e le virgole obbligatorie, verificare eventuali riferimenti mancanti e altri errori di sintassi. È consigliabile utilizzare un editor di testo in grado di convalidare visivamente la sintassi dei file JSON.

Componente della pipeline bloccato nello stato Waiting for Runner

Se la pipeline è nello stato SCHEDULED e una o più operazioni appaiono bloccate nello stato WAITING_FOR_RUNNER, verificare di aver impostato un valore valido per i campi runsOn o workerGroup per tali attività. Se entrambi i valori sono vuoti o mancanti, l'attività non può iniziare perché non vi è alcuna associazione tra l'attività e il lavoratore per eseguire l'attività. In questo caso, hai definito il lavoro ma non il computer che deve eseguire il lavoro. Se applicabile, verificate che il valore WorkerGroup assegnato al componente pipeline abbia esattamente lo stesso nome e maiuscole del valore WorkerGroup configurato per Task Runner.

> **Note**
>
> Se si fornisce un valore runsOn e workerGroup esiste, workerGroup verrà ignorato.

Un'altra possibile causa di questo problema è che l'endpoint e la chiave di accesso forniti a Task Runner non sono gli stessi della AWS Data Pipeline console o del computer su cui sono installati gli strumenti AWS Data Pipeline CLI. È possibile che abbiate creato nuove pipeline senza errori visibili, ma Task Runner esegue il polling nella posizione sbagliata a causa della differenza di credenziali o esegue il polling nella posizione corretta con autorizzazioni insufficienti per identificare ed eseguire il lavoro specificato dalla definizione della pipeline.

Componente della pipeline bloccato nello stato WAITING_ON_DEPENDENCIES

Se la pipeline è nello stato `SCHEDULED` e una o più attività appaiono bloccate nello stato `WAITING_ON_DEPENDENCIES`, verificare che le precondizioni iniziali della pipeline siano state soddisfatte. Se le precondizioni del primo oggetto nella catena logica non sono state soddisfatte, nessuno degli oggetti che dipendono da quel primo oggetto possono essere spostati dallo stato `WAITING_ON_DEPENDENCIES`.

Ad esempio, considerare il seguente estratto da una definizione di pipeline. In questo caso, l'`InputData`oggetto presenta una condizione preliminare «Pronto» che specifica che i dati devono esistere prima che l'oggetto sia completo. InputData Se i dati non esistono, l' InputData oggetto

rimane WAITING_ON_DEPENDENCIES nello stato, in attesa che i dati specificati dal campo del percorso diventino disponibili. Tutti gli oggetti che dipendono da InputData similmente rimangono in uno WAITING_ON_DEPENDENCIES stato in attesa che l' InputData oggetto raggiunga lo FINISHED stato.

```
{
    "id": "InputData",
    "type": "S3DataNode",
    "filePath": "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
    "schedule":{"ref":"MySchedule"},
    "precondition": "Ready"
},
{
    "id": "Ready",
    "type": "Exists"
...
```

Inoltre, controllare che gli oggetti abbiano le autorizzazioni richieste per accedere ai dati. Nell'esempio precedente, se le informazioni nel campo delle credenziali non disponevano delle autorizzazioni per accedere ai dati specificati nel campo del percorso, l' InputData oggetto rimarrebbe bloccato in uno WAITING_ON_DEPENDENCIES stato perché non può accedere ai dati specificati dal campo del percorso, anche se tali dati esistono.

È anche possibile che una risorsa che comunica con Amazon S3 non abbia un indirizzo IP pubblico associato. Ad esempio, un Ec2Resource in una sottorete pubblica deve disporre di un indirizzo IP pubblico associato.

Infine, in determinate condizioni, le istanze relative a risorse possono raggiungere lo stato WAITING_ON_DEPENDENCIES molto prima rispetto a quando è programmato l'avvio delle attività associate, dando l'impressione che la risorsa o l'attività stia fallendo.

L'esecuzione non inizia quando è stata programmata

Controllare di aver scelto il tipo di pianificazione corretta che stabilisce se l'attività viene avviata all'inizio dell'intervallo di pianificazione (Tipo di pianificazione stile Cron) o al termine dell'intervallo di pianificazione (Tipo di pianificazione serie temporali).

Inoltre, verifica di aver specificato correttamente le date negli oggetti di pianificazione e che i endDateTime valori startDateTime e siano in formato UTC, come nell'esempio seguente:

```
{
```

```
    "id": "MySchedule",
    "startDateTime": "2012-11-12T19:30:00",
    "endDateTime":"2012-11-12T20:30:00",
    "period": "1 Hour",
    "type": "Schedule"
},
```

Componenti della pipeline eseguiti in ordine errato

È possibile notare che gli orari di inizio e di fine dei componenti della pipeline vengano eseguiti nell'ordine errato o in un'altra sequenza rispetto a quella prevista. È importante capire che l'esecuzione dei componenti della pipeline può iniziare contemporaneamente se le loro precondizioni sono soddisfatte al momento dell'avvio. In altre parole, i componenti della pipeline non vengono eseguiti in sequenza per impostazione predefinita; se è necessario un determinato ordine di esecuzione, è necessario controllare l'ordine di esecuzione con le precondizioni e i campi `dependsOn`.

Verificare che si sta utilizzando il campo `dependsOn` popolato con un riferimento ai componenti della pipeline dei prerequisiti corretti e che tutti i puntatori necessari tra componenti siano presenti per ottenere l'ordine richiesto.

Il cluster EMR ha esito negativo con l'errore: "The security token included in the request is invalid" ("Il token di sicurezza incluso nella richiesta non è valido")

Verifica i ruoli, le policy e le relazioni di trust di IAM come descritto in[Ruoli IAM per AWS Data Pipeline](#).

Autorizzazioni insufficienti per accedere alle risorse

Le autorizzazioni impostate sui ruoli IAM determinano se è AWS Data Pipeline possibile accedere ai cluster EMR e alle istanze EC2 per eseguire le pipeline. Inoltre, IAM fornisce il concetto di relazioni di fiducia che vanno oltre per consentire la creazione di risorse per tuo conto. Ad esempio, quando crei una pipeline che utilizza un'istanza EC2 per eseguire un comando per spostare i dati, AWS Data Pipeline puoi fornire questa istanza EC2 per te. Se riscontri problemi, in particolare quelli che riguardano risorse a cui puoi accedere manualmente ma AWS Data Pipeline non puoi, verifica i ruoli, le policy e le relazioni di trust di IAM come descritto in. Ruoli IAM per AWS Data Pipeline

Codice di stato: 400 Codice di errore: PipelineNotFoundException

Questo errore indica che i ruoli predefiniti di IAM potrebbero non disporre delle autorizzazioni necessarie AWS Data Pipeline per funzionare correttamente. Per ulteriori informazioni, consulta Ruoli IAM per AWS Data Pipeline.

La creazione di una pipeline provoca un errore relativo al Security Token

Il seguente messaggio di errore viene ricevuto quando si tenta di creare una pipeline:

Creazione della pipeline con 'pipeline_name' non riuscita. Errore: UnrecognizedClientException - Il token di sicurezza incluso nella richiesta non è valido.

Impossibile visualizzare i dettagli della pipeline nella console

Il filtro della pipeline della AWS Data Pipeline console si applica alla data di inizio pianificata per una pipeline, indipendentemente da quando la pipeline è stata inviata. È possibile inviare una nuova pipeline utilizzando una data di inizio programmata che si verifica nel passato, che il filtro di data predefinito potrebbe non mostrare. Per vedere i dettagli della pipeline, modificare il filtro della data per avere la garanzia che la data di inizio programmata della pipeline cada all'interno del filtro dell'intervallo di date.

Errore in remote runner Codice stato: 404, AWS Service: Amazon S3

Questo errore indica che Task Runner non è riuscito ad accedere ai tuoi file in Amazon S3. Verificare che:

- Le credenziali siano state impostate correttamente
- Il bucket Amazon S3 a cui stai tentando di accedere esiste
- Sei autorizzato ad accedere al bucket Amazon S3

Accesso negato - Non autorizzato per eseguire la funzione datapipeline:

Nei log di Task Runner, potresti visualizzare un errore simile al seguente:

- Codice stato ERRORE: 403
- Servizio AWS: DataPipeline
- Codice di errore AWS: AccessDenied

- Messaggio di errore AWS: User: arn:aws:sts: :XXXXXXXXXX:Federated-User/I-XXXXXXXX non è autorizzato a eseguire: datapipeline:. PollForTask

> **Note**
>
> In questo PollForTask messaggio di errore, può essere sostituito con nomi di altre AWS Data Pipeline autorizzazioni.

Questo messaggio di errore indica che il ruolo IAM specificato richiede le autorizzazioni aggiuntive necessarie per interagire con. AWS Data Pipeline Assicurati che la tua policy sui ruoli IAM contenga le seguenti righe, dove PollForTask viene sostituita dal nome dell'autorizzazione che desideri aggiungere (usa* per concedere tutte le autorizzazioni). Per ulteriori informazioni su come creare un nuovo ruolo IAM e applicarvi una policy, consulta Managing IAM Policies nella guida Using IAM.

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

Le versioni precedenti di Amazon EMR AMIs possono creare dati falsi per file CSV di grandi dimensioni

Su Amazon EMR, le versioni AMIs precedenti alla versione 3.9 (3.8 e precedenti) AWS Data Pipeline utilizzavano un file CSV personalizzato InputFormat per leggere e scrivere file CSV da utilizzare con i lavori. MapReduce Viene utilizzato quando il servizio imposta le tabelle da e verso Amazon S3. InputFormat È stato scoperto un problema a causa del quale la lettura di record da file CSV di grandi dimensioni può comportare la produzione di tabelle che non vengono copiate correttamente. Questo problema è stato risolto nelle versioni successive di Amazon EMR. Utilizza l'AMI Amazon EMR 3.9 o una versione Amazon EMR 4.0.0 o successiva.

Limiti AWS Data Pipeline crescenti

Occasionalmente, è possibile superare i limiti specifici AWS Data Pipeline del sistema. Ad esempio, il limite predefinito della pipeline è di 20 pipeline con 50 oggetti in ognuna. Se si scopre che si ha bisogno di più pipeline rispetto al limite, considerare l'unione di più pipeline per creare un numero

minore di pipeline con più oggetti in ognuna. Per ulteriori informazioni sui limiti di AWS Data Pipeline , consulta AWS Data Pipeline Limiti. Tuttavia, se non è possibile risolvere il problema dei limiti tramite tecnica di unione delle pipeline, è opportuno richiedere un aumento della capacità utilizzando questo modulo: Aumento limiti della pipeline dei dati.

AWS Data Pipeline Limiti

Per garantire la capacità di tutti gli utenti, AWS Data Pipeline impone limiti alle risorse che è possibile allocare e alla velocità con cui è possibile allocare le risorse.

Indice

Limiti dell'account

I seguenti limiti si applicano a un singolo account. AWS Se hai bisogno di capacità aggiuntiva, puoi utilizzare il modulo di richiesta dell'Amazon Web Services Support Center per aumentare la tua capacità.

Attributo	Limite	Regolabile
Numero di pipeline	100	Sì
Numero di oggetti per pipeline	100	Sì
Numero di istanze attive per oggetto	5	Sì
Numero di campi per oggetto	50	No
Numero di UTF8 byte per nome di campo o identificatore	256	No
Numero di UTF8 byte per campo	10,240	No

Attributo	Limite	Regolabile
Numero di UTF8 byte per oggetto	15.360 (inclusi i nomi campo)	No
Velocità di creazione di un'istanza da un oggetto	1 ogni 5 minuti	No
Nuovi tentativi per un'attività di pipeline	5 per attività	No
Intervallo minimo tra i tentativi	2 minuti	No
Intervallo di pianificazione minimo	15 minuti	No
Numero massimo di roll-up in un singolo oggetto	32	No
Numero massimo di EC2 istanze per oggetto Ec2Resource	1	No

Limiti chiamata del servizio Web

AWS Data Pipeline limita la velocità con cui è possibile chiamare l'API del servizio Web. Questi limiti si applicano anche agli AWS Data Pipeline agenti che chiamano l'API del servizio Web per tuo conto, come la console, la CLI e Task Runner.

I seguenti limiti si applicano a un singolo AWS account. Questo significa che l'utilizzo totale dell'account, incluso quello degli utenti , non può superare tali limiti.

La velocità di burst consente di risparmiare le chiamate del servizio Web durante i periodi di inattività e impiegarle tutte in un breve periodo di tempo. Ad esempio, CreatePipeline ha una frequenza normale di una chiamata ogni cinque secondi. Se non chiami il servizio per 30 secondi, hai sei

chiamate salvate. È quindi possibile chiamare il servizio Web sei volte in un secondo. Dal momento che questo numero è al di sotto del limite di aumento delle prestazioni e mantiene il limite di chiamate medio alla tariffa ordinaria, le chiamate non vengono limitate.

Se si supera il limite di frequenza e il limite di aumento delle prestazioni, la chiamata al servizio Web non va a buon fine e restituisce un'eccezione di throttling. L'implementazione predefinita di un worker, Task Runner, riprova automaticamente le chiamate API che hanno esito negativo con un'eccezione di limitazione. Task Runner dispone di un sistema di backup, in modo che i tentativi successivi di chiamata all'API avvengano a intervalli sempre più lunghi. Se si scrive un lavoratore, consigliamo di implementare una logica di ripetizione analoga.

Questi limiti vengono applicati a un singolo AWS account.

"Hello, World!"	Limite frequenza regolare	Limite di aumento delle prestazioni
ActivatePipeline	1 chiamata al secondo	100 chiamate
CreatePipeline	1 chiamata al secondo	100 chiamate
DeletePipeline	1 chiamata al secondo	100 chiamate
DescribeObjects	2 chiamate al secondo	100 chiamate
DescribePipelines	1 chiamata al secondo	100 chiamate
GetPipelineDefinition	1 chiamata al secondo	100 chiamate
PollForTask	2 chiamate al secondo	100 chiamate
ListPipelines	1 chiamata al secondo	100 chiamate
PutPipelineDefinition	1 chiamata al secondo	100 chiamate
QueryObjects	2 chiamate al secondo	100 chiamate
ReportTaskProgress	10 chiamate al secondo	100 chiamate
SetTaskStatus	10 chiamate al secondo	100 chiamate
SetStatus	1 chiamata al secondo	100 chiamate

"Hello, World!"	Limite frequenza regolare	Limite di aumento delle prestazioni
ReportTaskRunnerHeartbeat	1 chiamata al secondo	100 chiamate
ValidatePipelineDefinition	1 chiamata al secondo	100 chiamate

Considerazioni su dimensionamento

AWS Data Pipeline si adatta a un numero enorme di attività simultanee ed è possibile configurarlo per creare automaticamente le risorse necessarie per gestire carichi di lavoro di grandi dimensioni. Queste risorse create automaticamente sono sotto il controllo dell'utente e vengono conteggiate ai fini dei limiti delle risorse dell'account AWS . Ad esempio, se configuri per AWS Data Pipeline creare automaticamente un cluster Amazon EMR a 20 nodi per elaborare i dati e AWS il tuo account ha EC2 un limite di istanze impostato su 20, potresti inavvertitamente esaurire le risorse di backfill disponibili. Di conseguenza, è necessario considerare queste limitazioni in termini di risorse nel progetto oppure aumentare i limiti dell'account in base alle necessità.

Se hai bisogno di capacità aggiuntiva, puoi utilizzare il modulo di richiesta dell'Amazon Web Services Support Center per aumentare la tua capacità.

AWS Data Pipeline Risorse

Le seguenti sono risorse che ti aiutano a utilizzare AWS Data Pipeline.

- AWS Data Pipeline Informazioni sul prodotto: la pagina Web principale per informazioni su AWS Data Pipeline.
- AWS Data Pipeline Domande frequenti tecniche: contiene le 20 domande principali poste dagli sviluppatori su questo prodotto.
- Note di rilascio: forniscono una panoramica di alto livello della versione corrente. In particolare, vengono indicate tutte le nuove funzioni, le correzioni e tutti i problemi noti.
- Forum di discussione di AWS Data Pipeline: un forum basato sulla community per gli sviluppatori per discutere di questioni tecniche relative ad Amazon Web Services.

- Corsi e workshop: collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS
- AWS Developer Center: esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti di sviluppo SDKs, toolkit IDE e strumenti da riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttivo: scopri come configurare Account AWS, entrare a far parte della AWS community e lanciare la tua prima applicazione.
- Tutorial pratici: segui i tutorial per avviare la step-by-step tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- Supporto AWS Center: l'hub per la creazione e la gestione dei casi. Supporto AWS Include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.
- Supporto— La pagina web principale per informazioni su Supporto one-on-one, un canale di supporto a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- Contattaci: un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .
- AWS Termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

Cronologia dei documenti

La presente documentazione è associata alla versione 2012-10-29 di. AWS Data Pipeline

Modifica	Description	Data di rilascio
AWS Data Pipeline non è più disponibile per i nuovi clienti	AWS Data Pipeline non è più disponibile per i nuovi clienti. I clienti esistenti di AWS Data Pipeline possono continuare a utilizzare il servizio normalmente. Ulteriori informazioni	25 luglio 2025
È stata aggiunta la documentazione per l'esecuzione di determinate procedure utilizzan do la AWS CLI. Sono state rimosse le procedure relative alla AWS Data Pipeline console.	Per ulteriori informazioni, consultare Clonazione della pipeline, Visualizza log pipeline e Crea una pipeline dai modelli di Data Pipeline utilizzando la CLI.	26 maggio 2023
Sono stati aggiunti altri contenuti ed esempi per la migrazione AWS Data Pipeline da altri servizi alternativi.	È stato aggiornato l'argomento per la migrazione AWS Data Pipeline a AWS Step Functions o Amazon MWAA con ulteriori informazioni su ciascuna alternati va, mappature concettuali tra i servizi ed esempi. AWS Glue Per ulteriori informazioni, consulta Migrazione dei carichi di lavoro da AWS Data Pipeline.	31 marzo 2023
Sono state aggiunte informazioni sul AWS Data Pipeline supporto di IMDSv2.	AWS Data Pipeline supporta IMDSv2 le risorse Amazon EMR e Amazon EC2 . Per ulteriori informazi oni, consultare Protezione dei dati in AWS Data Pipeline, EmrCluster e Ec2Resource.	16 dicembre 2022
È stato aggiunto un argomento per la migrazione AWS	Ora esistono altri AWS servizi che offrono ai clienti una migliore esperienza di integrazione dei dati. Puoi migrare i casi d'uso tipici AWS Data Pipeline verso	16 dicembre 2022

Modifica	Description	Data di rilascio
Data Pipeline da altri servizi alternativi.	AWS Step Functions o Amazon MWAA. AWS Glue Per ulteriori informazioni, consulta Migrazione dei carichi di lavoro da AWS Data Pipeline.	
Sono stati aggiornati gli elenchi delle istanze Amazon EC2 e Amazon EMR supportate. È stato aggiornato l'elenco IDs delle HVM (Hardware Virtual Machine) AMIs utilizzate per le istanze.	Sono stati aggiornati gli elenchi delle istanze Amazon EC2 e Amazon EMR supportate. Per ulteriori informazioni, consulta Tipi di istanza supportati per attività di lavoro delle pipeline. È stato aggiornato l'elenco IDs delle HVM (Hardware Virtual Machine) AMIs utilizzate per le istanze. Per ulteriori informazioni, consulta Sintassi e cerca `imageId`.	9 novembre 2018

Modifica	Description	Data di rilascio
È stata aggiunta una configurazione per collegare i volumi Amazon EBS ai nodi del cluster e per avviare un cluster Amazon EMR in una sottorete privata.	Aggiunte le opzioni di configurazione a un oggetto `EMRcluster` . Puoi utilizzare queste opzioni nelle pipeline che utilizzano cluster Amazon EMR. Utilizza i `TaskEbsConfiguration` campi `coreEbsConfiguration` `masterEbs Configuration` , e per configurare l'allegato dei volumi Amazon EBS ai nodi core, master e task nel cluster Amazon EMR. Per ulteriori informazioni, consulta Collegare i volumi EBS ai nodi del cluster. Utilizza i `ServiceAccessSecurityGroupI d` campi `emrManagedMasterSecurityGro upId` `emrManagedSlaveSecurityGroupId` , e per configurare un cluster Amazon EMR in una sottorete privata. Per ulteriori informazioni, consulta Configurare un cluster Amazon EMR in una sottorete privata. Per ulteriori informazioni sulla sintassi `EMRcluster` , consulta EmrCluster.	19 aprile 2018
È stato aggiunto l'elenco delle istanze Amazon EC2 e Amazon EMR supportate.	È stato aggiunto l'elenco delle istanze che vengono AWS Data Pipeline create per impostazione predefini ta, se non si specifica un tipo di istanza nella definizio ne della pipeline. È stato aggiunto un elenco di istanze Amazon EC2 e Amazon EMR supportate. Per ulteriori informazioni, consulta Tipi di istanza supportati per attività di lavoro delle pipeline.	22 marzo 2018
Aggiunto il supporto per pipeline on demand.	• Aggiunto il supporto per pipeline on demand, che consente di eseguire nuovamente una pipeline attivandola di nuovo.	22 febbraio 2016

Modifica	Description	Data di rilascio
Supporto aggiuntivo per database RDS	• Aggiunti `rdsInstanceId` , `region` e `jdbcDriverJarUri` a RdsDatabase. • Aggiornato `database` in SqlActivity per supportare anche `RdsDatabase` .	17 agosto 2015
Ulteriore supporto di JDBC	• Aggiornato `database` in SqlActivity per supportare anche `JdbcDatabase` . • È stato aggiunto `jdbcDriverJarUri` a JdbcDatabase. • Aggiunto `initTimeout` a Ec2Resource e EmrCluster. • Aggiunto `runAsUser` a Ec2Resource.	7 luglio 2015
HadoopActivity, Zona di disponibilità e Spot Support	• Aggiunto il supporto per l'invio di lavoro parallelo Al cluster Hadoop. Per ulteriori informazioni, consulta HadoopActivity. • Aggiunta la possibilità di richiedere le istanze Spot con Ec2Resource e EmrCluster. • Aggiunta la possibilità di avviare risorse `EmrCluster` in una determinata zona di disponibilità.	1 giugno 2015
Disattivazione pipeline	Aggiunto il supporto per la disattivazione delle pipeline attive. Per ulteriori informazioni, consulta Disattivazione pipeline.	7 aprile 2015
Aggiornati modelli e console	Aggiunti nuovi modelli. È stato aggiornato il capitolo Guida introduttiva per utilizzare il ShellCommandActivity modello Guida introduttiva con. Per ulteriori informazioni, consulta Crea una pipeline dai modelli di Data Pipeline utilizzando la CLI.	25 novembre 2014

Modifica	Description	Data di rilascio
Supporto per VPC	Aggiunto il supporto per avviare le risorse in un cloud privato virtuale (VPC).	12 marzo 2014
Supporto delle Regioni	Aggiunto il supporto per diverse regioni del servizio. Inoltre`us-east-1` , AWS Data Pipeline è supportato anche in`eu-west-1` , `ap-northeast-1` `ap-southeast-2` , `eus-west-2` .	20 febbraio 2014
Supporto di Amazon Redshift	È stato aggiunto il supporto per Amazon Redshift in AWS Data Pipeline, incluso un nuovo modello di console (Copy to Redshift) e un tutorial per illustrare il modello. Per ulteriori informazioni, consulta Copia i dati su Amazon Redshift utilizzando AWS Data Pipeline, RedshiftDataNode, RedshiftDatabase e RedshiftCopyActivity.	6 novembre 2013
PigActivity	Aggiunto PigActivity, che fornisce supporto nativo per Pig. Per ulteriori informazioni, consulta PigActivity.	15 ottobre 2013
Nuovo modello di console, attività e formato di dati	È stato aggiunto il nuovo CrossRegion modello di console DynamoDB Copy, che include il HiveCopyActivity nuovo modello e Dynamo. DBExport DataFormat	21 agosto 2013
Guasti di una delle dipendenze e riesecuzioni	Sono state aggiunte informazioni sugli errori a AWS Data Pipeline cascata e sul comportamento di riesecuzione. Per ulteriori informazioni, consulta Guasti di una delle dipendenze e riesecuzioni.	8 agosto 2013
Risoluzione dei problemi video	È stato aggiunto il video sulla risoluzione dei AWS Data Pipeline problemi di base. Per ulteriori informazioni, consulta Risoluzione dei problemi.	17 luglio 2013
Modifica delle pipeline attive	Aggiunte ulteriori informazioni sulla modifica delle pipeline attive e sulla riesecuzione dei componenti della pipeline. Per ulteriori informazioni, consulta Modifica della pipeline.	17 luglio 2013

Modifica	Description	Data di rilascio
Utilizza le risorse in regioni diverse	Aggiunte ulteriori informazioni sull'utilizzo delle risorse in regioni diverse. Per ulteriori informazioni, consulta Utilizzo di una pipeline con risorse in più regioni.	17 giugno 2013
Stato WAITING_ON_DEPENDENCIES	Stato CHECKING_PRECONDITIONS modificato in WAITING_ON_DEPENDENCIES e aggiunto il campo runtime @waitingOn per gli oggetti delle pipeline.	20 maggio 2013
Formato Dynamo DBData	È stato aggiunto il modello Dynamo FormatDBData.	23 aprile 2013
Video log Web dei processi e supporto delle istanze Spot	Ha presentato il video «Elaborazione dei log Web con AWS Data Pipeline, Amazon EMR e Hive» e EC2 il supporto delle istanze Amazon Spot.	21 febbraio 2013
	La versione iniziale della Developer Guide. AWS Data Pipeline	20 dicembre 2012