aws

Panduan Developerr

AWS Data Pipeline

Versi API 2012-10-29

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Data Pipeline: Panduan Developerr

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

..... ix
Apa itu AWS Data Pipeline? 1
Migrasi beban kerja dari AWS Data Pipeline 2
Migrasi beban kerja ke AWS Glue 3
Migrasi beban kerja ke AWS Step Functions 4
Memigrasi beban kerja ke Amazon MWAA 5
Memetakan konsep 6
Sampel 7
Layanan terkait 8
Mengakses AWS Data Pipeline 9
Harga 10
Tipe Instans yang Didukung untuk Aktivitas Kerja Alur 10
EC2 Instans Amazon default menurut Wilayah AWS 11
EC2Instans Amazon Tambahan yang Didukung 12
EC2 Instans Amazon yang Didukung untuk Cluster EMR Amazon 13
AWS Data Pipeline Konsep 15
Definisi Alur 15
Komponen Alur, Instans, dan Upaya 16
Runner Tugas 18
Simpul Data 19
Basis Data 20
Aktivitas 20
Prasyarat 21
Prasyarat yang Dikelola Sistem 22
Prasyarat Dikelola Pengguna 22
Sumber daya 22
Batasan sumber daya 23
Platform yang Didukung 23
Instans Amazon EC2 Spot dengan Cluster EMR Amazon dan AWS Data Pipeline 24
Tindakan 25
Pemantauan Alur Proaktif 25
Mengatur 26
Mendaftar untuk AWS 26
Mendaftar untuk Akun AWS 26

Buat pengguna dengan akses administratif ... 27
Buat Peran IAM untuk AWS Data Pipeline dan Pipeline Resources ... 28
Izinkan IAM utama (Pengguna dan Grup) untuk Melakukan Tindakan yang Diperlukan ... 28
Memberikan akses terprogram ... 30
Memulai dengan AWS Data Pipeline ... 33
Buat Alur ... 34
Memantau Alur Berjalan ... 35
Lihat Output ... 36
Hapus Alur ... 36
Bekerja dengan jaringan pipa ... 37
Membuat pipa ... 37
Buat pipeline dari template Data Pipeline menggunakan CLI ... 38
Melihat Alur Anda ... 57
Menafsirkan Kode Status Alur ... 57
Menafsirkan Alur dan Status Kondisi Komponen ... 59
Melihat Definisi Alur Anda ... 61
Melihat Detail Instans Alur ... 61
Melihat Log Alur ... 62
Mengedit Alur Anda ... 64
Batasan ... 64
Mengedit Pipeline Menggunakan AWS CLI ... 65
Mengkloning Alur Anda ... 65
Menandai Alur Anda ... 66
Menonaktifkan Alur Anda ... 67
Menonaktifkan Alur Anda Menggunakan AWS CLI ... 67
Menghapus Alur Anda ... 68
Penahapan Data dan Tabel dengan Aktivitas ... 69
Pementasan Data dengan ShellCommandActivity ... 70
Penahapan Tabel dengan Hive dan Simpul Data yang Didukung Penahapan ... 71
Penahapan Tabel dengan Hive dan Simpul Data yang Tidak Didukung Penahapan ... 73
Menggunakan Sumber Daya di Beberapa Wilayah ... 74
Kegagalan dan tayangan ulang yang berulang ... 76
Aktivitas ... 77
Node dan prasyarat data ... 77
Sumber daya ... 77
Menjalankan kembali objek yang gagal bertingkat ... 78

Kegagalan bertingkat dan penimbunan ulang 78
Sintaks file definisi pipa 78
Struktur File 79
Bidang Alur 80
Bidang yang ditentukan pengguna 81
Bekerja dengan API 82
Pasang AWS SDK 82
Membuat Permintaan HTTP ke AWS Data Pipeline 83
Keamanan 88
Perlindungan Data 89
Identity and Access Management 90
Kebijakan IAM untuk AWS Data Pipeline 91
Contoh Kebijakan untuk AWS Data Pipeline 95
Peran IAM 98
Pembuatan Log dan Pemantauan 103
AWS Data Pipeline Informasi di CloudTrail 103
Memahami Entri File AWS Data Pipeline Log 104
Tanggapan Insiden 105
Validasi Kepatuhan 105
Ketahanan 105
Keamanan Infrastruktur 106
Konfigurasi dan Analisis Kerentanan di AWS Data Pipeline 106
Tutorial 107
Proses Data Menggunakan Amazon EMR dengan Hadoop Streaming 107
Sebelum Anda Memulai 108
Menggunakan CLI 108
Salin Data CSV dari Amazon S3 ke Amazon S3 112
Sebelum Anda Memulai 114
Menggunakan CLI 114
Ekspor Data MySQL ke Amazon S3 121
Sebelum Anda Memulai 122
Menggunakan CLI 123
Salin Data ke Amazon Redshift 133
Sebelum Anda Mulai: Mengonfigurasi Opsi COPY 133
Sebelum Anda Mulai: Mengatur Alur, Keamanan, dan Klaster 134
Menggunakan CLI 136

Ekspresi dan Fungsi Alur 146
Tipe Data Sederhana 146
DateTime 146
Numerik 146
Referensi Objek 146
Periode 147
String 147
Ekspresi 147
Mereferensikan Bidang dan Objek 148
Ekspresi Terinduk 149
Daftar 149
Ekspresi simpul 150
Evaluasi Ekspresi 151
Fungsi Matematika 151
Fungsi String 152
Fungsi Tanggal dan Waktu 153
Karakter khusus 162
Referensi Objek Alur 163
Simpul Data 164
Node Dinamo DBData 165
MySqlDataNode 172
RedshiftDataNode 179
S3 DataNode 187
SqlDataNode 195
Aktivitas 202
CopyActivity 202
EmrActivity 210
HadoopActivity 220
HiveActivity 231
HiveCopyActivity 241
PigActivity 251
RedshiftCopyActivity 265
ShellCommandActivity 279
SqlActivity 288
Sumber daya 297
Ec2Resource 297

EmrCluster ... 309
HttpProxy ... 340
Prasyarat ... 343
Dynamo Ada DBData ... 343
Dynamo Ada DBTable ... 347
Exists ... 351
S3 KeyExists ... 355
S3 PrefixNotEmpty ... 359
ShellCommandPrecondition ... 364
Basis Data ... 368
JdbcDatabase ... 369
RdsDatabase ... 371
RedshiftDatabase ... 373
Format Data ... 376
Format Data CSV ... 376
Format Data Kustom ... 378
Format Dinamo DBData ... 379
Dinamo DBExport DataFormat ... 382
RegEx Format Data ... 385
Format Data TSV ... 387
Tindakan ... 389
SnsAlarm ... 389
Mengakhiri ... 391
Jadwal ... 393
Contoh ... 393
Sintaksis ... 398
Utilitas ... 400
ShellScriptConfig ... 400
EmrConfiguration ... 402
Properti ... 407
Bekerja dengan Runner Tugas ... 410
Pelari Tugas pada Sumber Daya yang AWS Data Pipeline Dikelola ... 410
Menjalankan Pekerjaan pada Sumber Daya yang Ada Menggunakan Runner Tugas ... 412
Pemasangan Runner Tugas ... 414
(Opsional) Memberikan Akses Runner Tugas ke Amazon RDS ... 414
Memulai Runner Tugas ... 416

Memverifikasi Pencatatan Runner Tugas 417
Thread dan Prasyarat Runner Tugas 417
Opsi Konfigurasi Runner Tugas 417
Menggunakan Runner Tugas dengan Proxy 420
Pelari Tugas dan Kustom AMIs 420
Pemecahan Masalah 422
Menemukan Kesalahan dalam Alur 422
Mengidentifikasi Klaster Amazon EMR yang Melayani Alur Anda 423
Menafsirkan Detail Status Alur 423
Menemukan Log Kesalahan 425
Log Alur 425
Tugas Hadoop dan Log Langkah Amazon EMR 426
Menyelesaikan Masalah Umum 426
Alur Terjebak dalam Status Tertunda 427
Komponen Alur Terjebak dalam Menunggu Status Runner 427
Komponen Alur Terjebak dalam Status WAITING_ON_DEPENDENCIES 428
Jalankan Tidak Mulai Saat Dijadwalkan 429
Komponen Alur Berjalan dalam Urutan yang Salah 429
Klaster EMR Gagal Dengan Kesalahan: Token keamanan yang disertakan dalam permintaan tidak valid 430
Izin Tidak Memadai untuk Mengakses Sumber Daya 430
Kode Status: 400 Kode Kesalahan: PipelineNotFoundException 430
Membuat Alur Menyebabkan Kesalahan Token Keamanan 430
Tidak Dapat Melihat Detail Alur di Konsol Tersebut 431
Kesalahan dalam Kode Status runner jarak jauh: 404, Layanan AWS: Amazon S3 431
Akses Ditolak - Tidak Diotorisasi untuk Melakukan Fungsi datapipeline: 431
Amazon EMR Lama AMIs Dapat Membuat Data Palsu untuk File CSV Besar 432
Meningkatkan AWS Data Pipeline Batas 432
Batas 433
Batasan Akun 433
Batas Panggilan Layanan Web 434
Pertimbangan Penskalaan 436
AWS Data Pipeline Sumber Daya 437
Riwayat Dokumen 439

AWS Data Pipeline tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS Data Pipeline dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.

Apa itu AWS Data Pipeline?

> **Note**
>
> AWS Data Pipeline layanan dalam mode pemeliharaan dan tidak ada fitur baru atau perluasan wilayah yang direncanakan. Untuk mempelajari lebih lanjut dan mengetahui cara memigrasi beban kerja yang ada, lihat. Migrasi beban kerja dari AWS Data Pipeline

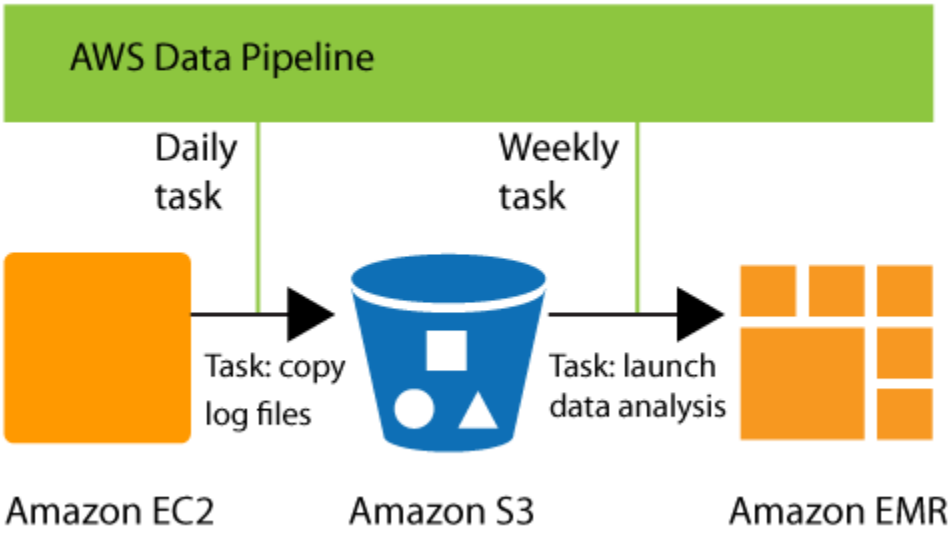
AWS Data Pipeline adalah layanan web yang dapat Anda gunakan untuk mengotomatiskan pergerakan dan transformasi data. Dengan AWS Data Pipeline, Anda dapat menentukan alur kerja berbasis data, sehingga tugas dapat bergantung pada keberhasilan penyelesaian tugas sebelumnya. Anda menentukan parameter transformasi data Anda dan AWS Data Pipeline menerapkan logika yang telah Anda atur.

Komponen berikut AWS Data Pipeline bekerja sama untuk mengelola data Anda:

- Sebuah definisi alur menentukan logika bisnis manajemen data Anda. Untuk informasi selengkapnya, lihat Sintaks file definisi pipa.
- Pipeline menjadwalkan dan menjalankan tugas dengan membuat EC2 instans Amazon untuk melakukan aktivitas kerja yang ditentukan. Anda mengunggah definisi alur Anda ke alur, dan kemudian mengaktifkan alur. Anda dapat mengedit definisi alur untuk alur berjalan dan mengaktifkan alur kembali agar alur dapat bekerja. Anda dapat menonaktifkan alur, memodifikasi sumber data, dan kemudian mengaktifkan alur kembali. Setelah Anda selesai dengan alur, Anda dapat menghapusnya.
- Task Runner membuat poling untuk tugas lalu melakukan tugas tersebut. Misalnya, Task Runner bisa menyalin berkas log ke Amazon S3 dan meluncurkan klaster Amazon EMR. Task Runner diinstal dan berjalan secara otomatis pada sumber daya yang dibuat oleh definisi alur Anda. Anda dapat menulis aplikasi task runner kustom, atau Anda dapat menggunakan aplikasi Task Runner yang disediakan oleh. AWS Data Pipeline Untuk informasi selengkapnya, lihat Runner Tugas.

Misalnya, Anda dapat menggunakannya AWS Data Pipeline untuk mengarsipkan log server web Anda ke Amazon Simple Storage Service (Amazon S3) setiap hari dan kemudian menjalankan klaster Amazon EMR (Amazon EMR) mingguan di atas log tersebut untuk menghasilkan laporan lalu lintas. AWS Data Pipeline menjadwalkan tugas harian untuk menyalin data dan tugas mingguan untuk meluncurkan cluster EMR Amazon. AWS Data Pipeline juga memastikan bahwa Amazon EMR

menunggu data hari terakhir diunggah ke Amazon S3 sebelum memulai analisisnya, bahkan jika ada penundaan yang tidak terduga dalam mengunggah log.



Daftar Isi

- Migrasi beban kerja dari AWS Data Pipeline
- Layanan terkait
- Mengakses AWS Data Pipeline
- Harga
- Tipe Instans yang Didukung untuk Aktivitas Kerja Alur

Migrasi beban kerja dari AWS Data Pipeline

AWS meluncurkan AWS Data Pipeline layanan pada tahun 2012. Pada saat itu, pelanggan mencari layanan untuk membantu mereka memindahkan data dengan andal antara sumber data yang berbeda menggunakan berbagai opsi komputasi. Sekarang, ada layanan lain yang menawarkan pelanggan pengalaman yang lebih baik. Misalnya, Anda dapat menggunakan AWS Glue to untuk menjalankan dan mengatur aplikasi Apache Spark, Step Functions AWS untuk membantu mengatur komponen AWS layanan, atau Amazon Managed Workflow for Apache Airflow (Amazon MWAA) untuk membantu mengelola orkestrasi alur kerja untuk Apache Airflow.

Topik ini menjelaskan cara bermigrasi dari AWS Data Pipeline ke opsi alternatif. Opsi yang Anda pilih tergantung pada beban kerja Anda saat ini. AWS Data Pipeline Anda dapat memigrasikan kasus

penggunaan umum AWS Data Pipeline ke salah satu AWS Glue, AWS Step Functions, atau Amazon MWAA.

Migrasi beban kerja ke AWS Glue

AWS Glueadalah layanan integrasi data tanpa server yang memudahkan pengguna analitik untuk menemukan, menyiapkan, memindahkan, dan mengintegrasikan data dari berbagai sumber. Ini termasuk perkakas untuk menulis, menjalankan pekerjaan, dan mengatur alur kerja. Dengan AWS Glue, Anda dapat menemukan dan terhubung ke lebih dari 70 sumber data yang beragam dan mengelola data Anda dalam katalog data terpusat. Anda dapat membuat, menjalankan, dan memantau pipeline ekstrak, mengubah, dan memuat (ETL) secara visual untuk memuat data ke dalam data lake Anda. Selain itu, Anda dapat segera mencari dan menanyakan data katalog menggunakan Amazon Athena, Amazon EMR, dan Amazon Redshift Spectrum.

Kami merekomendasikan untuk memigrasikan AWS Data Pipeline beban kerja Anda ke AWS Glue saat:

- Anda mencari layanan integrasi data tanpa server yang mendukung berbagai sumber data, antarmuka penulisan termasuk editor visual dan notebook, dan kemampuan manajemen data tingkat lanjut seperti kualitas data dan deteksi data sensitif.
- Beban kerja Anda dapat dimigrasikan ke AWS Glue alur kerja, pekerjaan (dengan Python atau Apache Spark) dan crawler (misalnya, pipeline yang ada dibangun di atas Apache Spark).
- Anda memerlukan satu platform yang dapat menangani semua aspek pipeline data Anda, termasuk konsumsi, pemrosesan, transfer, pengujian integritas, dan pemeriksaan kualitas.
- Pipeline yang ada dibuat dari template yang telah ditentukan sebelumnya di AWS Data Pipeline konsol, seperti mengekspor tabel DynamoDB ke Amazon S3, dan Anda mencari template tujuan yang sama.
- Beban kerja Anda tidak bergantung pada aplikasi ekosistem Hadoop tertentu seperti Apache Hive.
- Beban kerja Anda tidak memerlukan orkestrasi server lokal.

AWS membebankan tarif per jam, ditagih per detik, untuk crawler (menemukan data) dan pekerjaan ETL (memproses dan memuat data). AWS Glue Studio adalah mesin orkestrasi bawaan untuk AWS Glue sumber daya, dan ditawarkan tanpa biaya tambahan. Pelajari lebih lanjut tentang harga di AWS Glue Harga.

Migrasi beban kerja ke AWS Step Functions

AWS Step Functions adalah layanan orkestrasi tanpa server yang memungkinkan Anda membangun alur kerja untuk aplikasi penting bisnis Anda. Dengan Step Functions, Anda menggunakan editor visual untuk membangun alur kerja dan mengintegrasikan secara langsung dengan lebih dari 11.000 tindakan untuk lebih dari 250 AWS layanan, seperti AWS Lambda, Amazon EMR, DynamoDB, dan lainnya. Anda dapat menggunakan Step Functions untuk mengatur pipeline pemrosesan data, menangani kesalahan, dan bekerja dengan batas pembatasan pada layanan yang mendasarinya. AWS Anda dapat membuat alur kerja yang memproses dan mempublikasikan model pembelajaran mesin, mengatur layanan mikro, serta AWS layanan kontrol, seperti, untuk membuat alur kerja ekstrak, transformasi AWS Glue, dan beban (ETL). Anda juga dapat membuat alur kerja otomatis yang berjalan lama untuk aplikasi yang memerlukan interaksi manusia.

Demikian pula dengan AWS Data Pipeline, AWS Step Functions adalah layanan yang dikelola sepenuhnya yang disediakan oleh AWS. Anda tidak akan diminta untuk mengelola infrastruktur, menambal pekerja, mengelola pembaruan versi OS atau yang serupa.

Sebaiknya migrasi AWS Data Pipeline beban kerja Anda ke AWS Step Functions saat:

- Anda mencari layanan orkestrasi alur kerja tanpa server dan sangat tersedia.
- Anda sedang mencari solusi hemat biaya yang mengenakan biaya pada perincian pelaksanaan tugas tunggal.
- Beban kerja Anda mengatur tugas untuk beberapa AWS layanan lain, seperti Amazon EMR, Lambda, atau DynamoDB. AWS Glue
- Anda sedang mencari solusi low-code yang dilengkapi dengan desainer drag-and-drop visual untuk pembuatan alur kerja dan tidak memerlukan pembelajaran konsep pemrograman baru.
- Anda mencari layanan yang menyediakan integrasi dengan lebih dari 250 AWS layanan lain yang mencakup lebih dari 11.000 tindakan out-of-the-box, serta memungkinkan integrasi dengan AWS non-layanan dan aktivitas khusus.

Keduanya AWS Data Pipeline dan Step Functions menggunakan format JSON untuk menentukan alur kerja. Ini memungkinkan untuk menyimpan alur kerja Anda dalam kontrol sumber, mengelola versi, mengontrol akses, dan mengotomatisasi dengan CI/CD. Step Functions menggunakan sintaks yang disebut Amazon State Language yang sepenuhnya didasarkan pada JSON, dan memungkinkan transisi yang mulus antara representasi tekstual dan visual dari alur kerja.

Dengan Step Functions, Anda dapat memilih versi EMR Amazon yang sama dengan yang Anda gunakan saat ini. AWS Data Pipeline

Untuk memigrasi aktivitas pada sumber daya AWS Data Pipeline terkelola, Anda dapat menggunakan integrasi layanan AWS SDK pada Step Functions untuk mengotomatiskan penyediaan dan pembersihan sumber daya.

Untuk memigrasikan aktivitas di server lokal, EC2 instance yang dikelola pengguna, atau kluster EMR yang dikelola pengguna, Anda dapat menginstal agen SSM ke instans. Anda dapat memulai perintah melalui AWS Systems Manager Run Command dari Step Functions. Anda juga dapat memulai mesin status dari jadwal yang ditentukan di Amazon EventBridge.

AWS Step Functions memiliki dua jenis alur kerja: Alur Kerja Standar dan Alur Kerja Ekspres. Untuk Alur Kerja Standar, Anda dikenakan biaya berdasarkan jumlah transisi status yang diperlukan untuk menjalankan aplikasi Anda. Untuk Alur Kerja Ekspres, Anda dikenakan biaya berdasarkan jumlah permintaan untuk alur kerja dan durasinya. Pelajari lebih lanjut tentang penetapan harga di AWS Step Functions Pricing.

Memigrasi beban kerja ke Amazon MWAA

Amazon MWAA (Managed Workflows for Apache Airflow) adalah layanan orkestrasi terkelola untuk Apache Airflow yang memudahkan penyiapan dan pengoperasian jalur data di cloud dalam skala besar. end-to-end Apache Airflow adalah alat sumber terbuka yang digunakan untuk secara terprogram membuat, menjadwalkan, dan memantau urutan proses dan tugas yang disebut sebagai “alur kerja”. Dengan Amazon MWAA, Anda dapat menggunakan bahasa pemrograman Airflow dan Python untuk membuat alur kerja tanpa harus mengelola infrastruktur yang mendasarinya untuk skalabilitas, ketersediaan, dan keamanan. Amazon MWAA secara otomatis menskalakan kapasitas eksekusi alur kerjanya untuk memenuhi kebutuhan Anda, dan terintegrasi dengan layanan AWS keamanan untuk membantu memberi Anda akses cepat dan aman ke data Anda.

Demikian pula dengan AWS Data Pipeline, Amazon MWAA adalah layanan yang dikelola sepenuhnya yang disediakan oleh. AWS Meskipun Anda perlu mempelajari beberapa konsep baru khusus untuk layanan ini, Anda tidak diharuskan untuk mengelola infrastruktur, pekerja patch, mengelola pembaruan versi OS atau yang serupa.

Kami merekomendasikan untuk memigrasikan AWS Data Pipeline beban kerja Anda ke Amazon MWAA saat:

- Anda mencari layanan terkelola dan sangat tersedia untuk mengatur alur kerja yang ditulis dengan Python.

- Anda ingin beralih ke teknologi open-source yang dikelola sepenuhnya dan diadopsi secara luas, Apache Airflow, untuk portabilitas maksimum.
- Anda memerlukan satu platform yang dapat menangani semua aspek pipeline data Anda, termasuk konsumsi, pemrosesan, transfer, pengujian integritas, dan pemeriksaan kualitas.
- Anda sedang mencari layanan yang dirancang untuk orkestrasi pipeline data dengan fitur seperti UI kaya untuk observabilitas, restart untuk alur kerja yang gagal, pengisian ulang, dan percobaan ulang untuk tugas.
- Anda sedang mencari layanan yang dilengkapi dengan lebih dari 800 operator dan sensor pra-bangun, mencakup AWS serta AWS non-layanan.

Alur kerja Amazon MWAA didefinisikan sebagai Directed Acyclic Graphs () DAGs menggunakan Python, sehingga Anda juga dapat memperlakukannya sebagai kode sumber. Kerangka kerja Python Airflow yang dapat diperluas memungkinkan Anda membangun alur kerja yang terhubung dengan hampir semua teknologi. Muncul dengan antarmuka pengguna yang kaya untuk melihat dan memantau alur kerja dan dapat dengan mudah diintegrasikan dengan sistem kontrol versi untuk mengotomatiskan proses CI/CD .

Dengan Amazon MWAA, Anda dapat memilih versi EMR Amazon yang sama dengan yang Anda gunakan saat ini. AWS Data Pipeline

AWS mengenakan biaya untuk waktu lingkungan Aliran Udara Anda berjalan ditambah penskalaan otomatis tambahan untuk memberikan lebih banyak kapasitas pekerja atau server web. Pelajari lebih lanjut tentang penetapan harga di Alur Kerja Terkelola Amazon untuk Harga Apache Airflow.

Memetakan konsep

Tabel berikut berisi pemetaan konsep utama yang digunakan oleh layanan. Ini akan membantu orang yang akrab dengan Data Pipeline untuk memahami Step Functions dan terminologi MWAA.

Data Pipeline	Glue	Step Functions	Amazon MWAA
Pipeline	Alur kerja	Alur kerja	Grafik asilat langsung
Definisi pipa JSON	Definisi alur kerja atau cetak biru berbasis Python	Bahasa Negara Bagian Amazon JSON	Berbasis Python

Data Pipeline	Glue	Step Functions	Amazon MWAA
Aktivitas	Lowongan	Negara dan Tugas	Tugas (Operator dan Sensor)
Contoh	Job berjalan	Eksekusi	DAG berjalan
Upaya	Coba lagi upaya	Penangkap dan retrier	Mencoba lagi
Jadwal pipa	Jadwal pemicu	EventBridge Tugas penjadwal	Cron, jadwal , sadar data
Ekspresi dan fungsi pipa	Pustaka cetak biru	Step Functions fungsi intrinsik dan Lambda AWS	Kerangka Python yang dapat diperluas

Sampel

Bagian berikut mencantumkan contoh publik yang dapat Anda rujuk untuk bermigrasi dari AWS Data Pipeline ke layanan individual. Anda dapat merujuknya sebagai contoh, dan membangun pipeline Anda sendiri pada layanan individual dengan memperbarui dan mengujinya berdasarkan kasus penggunaan Anda.

AWS Glue sampel

Daftar berikut berisi contoh implementasi untuk kasus AWS Data Pipeline penggunaan yang paling umum dengan. AWS Glue

- Lowongan kerja Running Spark
- Menyalin data dari JDBC ke Amazon S3 (termasuk Amazon Redshift)
- Menyalin data dari Amazon S3 ke JDBC (termasuk Amazon Redshift)
- Menyalin data dari Amazon S3 ke DynamoDB
- Memindahkan data ke dan dari Amazon Redshift
- Akses lintas akun Lintas wilayah ke tabel DynamoDB

AWS Sampel Step Functions

Daftar berikut berisi contoh implementasi untuk AWS Data Pipeline kasus penggunaan yang paling umum dengan Step Functions AWS .

- Mengelola pekerjaan EMR Amazon
- Menjalankan pekerjaan pemrosesan data di Amazon EMR Tanpa Server
- Menjalankan Hive/Pig/Hadoop pekerjaan
- Menanyakan kumpulan data besar (Amazon Athena, Amazon S3,) AWS Glue
- Menjalankan alur kerja ETL menggunakan Amazon Redshift
- Mengatur crawler AWS Glue

Lihat tutorial tambahan dan contoh proyek untuk menggunakan AWS Step Functions.

Sampel Amazon MWAA

Daftar berikut berisi contoh implementasi untuk kasus AWS Data Pipeline penggunaan yang paling umum dengan Amazon MWAA.

- Menjalankan pekerjaan EMR Amazon
- Membuat plugin khusus untuk Apache Hive dan Hadoop
- Menyalin data dari Amazon S3 ke Redshift
- Menjalankan skrip Shell pada instance jarak jauh EC2
- Mengatur alur kerja hybrid (on-prem)

Lihat tutorial tambahan dan contoh proyek untuk menggunakan Amazon MWAA.

Layanan terkait

AWS Data Pipeline bekerja dengan layanan berikut untuk menyimpan data.

- Amazon DynamoDB - Menyediakan basis data NoSQL terkelola penuh dengan performa yang cepat dengan biaya rendah. Untuk informasi selengkapnya, lihat Panduan Developer Amazon DynamoDB.

- Amazon RDS - Menyediakan basis data relasional terkelola penuh yang menskalakan untuk set data besar. Untuk informasi selengkapnya, lihat Panduan Developer Amazon Relational Database Service.
- Amazon Redshift - Menyediakan gudang data yang cepat, terkelola penuh, berskala-petabyte yang memudahkan dan hemat biaya untuk menganalisis sejumlah besar data. Untuk informasi selengkapnya, lihat Panduan Developer Basis Data Amazon Redshift.
- Amazon S3 — Menyediakan penyimpanan objek yang aman, tahan lama, dan dapat diskalakan. Untuk informasi selengkapnya, lihat Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

AWS Data Pipeline bekerja dengan layanan komputasi berikut untuk mengubah data.

- Amazon EC2 — Menyediakan kapasitas komputasi yang dapat diubah ukurannya — secara harfiah, server di pusat data Amazon — yang Anda gunakan untuk membangun dan meng-host sistem perangkat lunak Anda. Untuk informasi selengkapnya, lihat Panduan EC2 Pengguna Amazon.
- Amazon EMR — Memudahkan, cepat, dan hemat biaya bagi Anda untuk mendistribusikan dan memproses sejumlah besar data di seluruh EC2 server Amazon, menggunakan kerangka kerja seperti Apache Hadoop atau Apache Spark. Untuk informasi lebih lanjut, lihat Panduan Developer Amazon EMR.

Mengakses AWS Data Pipeline

Anda dapat membuat, mengakses, dan mengelola alur Anda menggunakan salah satu antarmuka berikut:

- AWS Management Console — Menyediakan antarmuka web yang dapat Anda gunakan untuk mengakses AWS Data Pipeline.
- AWS Command Line Interface (AWS CLI) — Menyediakan perintah untuk serangkaian layanan AWS yang luas, termasuk AWS Data Pipeline, dan didukung di Windows, macOS, dan Linux. Untuk informasi lebih lanjut tentang menginstal AWS CLI, lihat AWS Command Line Interface. Untuk daftar perintah AWS Data Pipeline, lihat datapipeline.
- AWS SDKs — Menyediakan khusus bahasa APIs dan menangani banyak detail koneksi, seperti menghitung tanda tangan, menangani percobaan ulang permintaan, dan penanganan kesalahan. Untuk informasi selengkapnya, lihat AWS SDKs.

- Query API - Menyediakan tingkat rendah APIs yang Anda panggil menggunakan permintaan HTTPS. Menggunakan API Kueri merupakan cara paling langsung untuk mengakses AWS Data Pipeline, tetapi mengharuskan aplikasi Anda menangani detail tingkat rendah seperti membuat hash untuk menandatangani permintaan, dan penanganan kesalahan. Untuk informasi lebih lanjut, lihat AWS Data Pipeline Referensi API.

Harga

Dengan Amazon Web Services, Anda hanya membayar untuk apa yang Anda gunakan. Untuk AWS Data Pipeline, Anda membayar pipa Anda berdasarkan seberapa sering aktivitas dan prasyarat Anda dijadwalkan untuk dijalankan dan di mana mereka berjalan. Untuk informasi selengkapnya, silakan lihat Harga AWS Data Pipeline.

Jika akun AWS Anda berusia kurang dari 12 bulan, Anda berhak untuk menggunakan tingkat gratis. Tingkat gratis mencakup tiga prasyarat frekuensi rendah dan lima aktivitas frekuensi rendah per bulan tanpa biaya. Untuk informasi selengkapnya, lihat AWS Tingkat Gratis.

Tipe Instans yang Didukung untuk Aktivitas Kerja Alur

Saat AWS Data Pipeline menjalankan pipeline, ia mengkompilasi komponen pipeline untuk membuat satu set instance Amazon EC2 yang dapat ditindaklanjuti. Setiap instans berisi semua informasi untuk melakukan tugas tertentu. Set lengkap instans adalah daftar yang harus dilakukan dari alur. AWS Data Pipeline menyerahkan instans ke runner tugas untuk diproses.

EC2 instance datang dalam konfigurasi yang berbeda, yang dikenal sebagai tipe instance. Setiap tipe instans memiliki CPU, input/output, dan kapasitas penyimpanan yang berbeda. Selain menentukan tipe instans untuk suatu aktivitas, Anda dapat memilih opsi pembelian yang berbeda. Tidak semua tipe instans yang tersedia di semua Wilayah AWS. Jika tipe instans tidak tersedia, alur Anda dapat gagal untuk penyediaan atau macet saat melakukan penyediaan. Untuk informasi tentang ketersediaan instans, lihat Halaman EC2 Harga Amazon. Buka tautan untuk opsi pembelian instans Anda dan filter berdasarkan Wilayah untuk melihat apakah jenis instans tersedia di Wilayah. Untuk informasi selengkapnya tentang jenis instans, keluarga, dan tipe virtualisasi ini, lihat EC2 Instans Amazon dan Matriks Jenis Instance AMI Amazon Linux.

Tabel berikut menjelaskan jenis instance yang AWS Data Pipeline mendukung. Anda dapat menggunakan AWS Data Pipeline untuk meluncurkan EC2 instans Amazon di Wilayah mana pun, termasuk Wilayah yang AWS Data Pipeline tidak didukung. Untuk informasi tentang Wilayah yang AWS Data Pipeline didukung, lihat Wilayah dan Titik Akhir AWS.

Daftar Isi

- EC2 Instans Amazon default menurut Wilayah AWS
- EC2Instans Amazon Tambahan yang Didukung
- EC2 Instans Amazon yang Didukung untuk Cluster EMR Amazon

EC2 Instans Amazon default menurut Wilayah AWS

Jika Anda tidak menentukan tipe instans dalam definisi alur Anda, AWS Data Pipeline meluncurkan sebuah instans secara default.

Tabel berikut mencantumkan EC2 instance Amazon yang AWS Data Pipeline digunakan secara default di Wilayah yang AWS Data Pipeline didukung.

Nama Wilayah	Region	Tipe Instans
US East (N. Virginia)	us-east-1	m1.small
US West (Oregon)	us-west-2	m1.small
Asia Pacific (Sydney)	ap-southeast-2	m1.small
Asia Pacific (Tokyo)	ap-northeast-1	m1.small
EU (Ireland)	eu-west-1	m1.small

Tabel berikut mencantumkan EC2 instance Amazon yang AWS Data Pipeline diluncurkan secara default di Wilayah yang tidak AWS Data Pipeline didukung.

Nama Wilayah	Region	Tipe Instans
US East (Ohio)	us-east-2	t2.small
US West (N. California)	us-west-1	m1.small
Asia Pacific (Mumbai)	ap-south-1	t2.small
Asia Pacific (Singapore)	ap-southeast-1	m1.small

Nama Wilayah	Region	Tipe Instans
Asia Pacific (Seoul)	ap-northeast-2	t2.small
Canada (Central)	ca-central-1	t2.small
EU (Frankfurt)	eu-central-1	t2.small
EU (London)	eu-west-2	t2.small
EU (Paris)	eu-west-3	t2.small
South America (São Paulo)	sa-east-1	m1.small

EC2Instans Amazon Tambahan yang Didukung

Selain instans default yang dibuat jika Anda tidak menentukan tipe instans dalam definisi alur Anda, instans berikut didukung.

Tabel berikut mencantumkan EC2 instance Amazon yang AWS Data Pipeline mendukung dan dapat membuat, jika ditentukan.

Kelas instans	Tipe instans
Tujuan umum	t2.nano | t2.micro | t2.small | t2.medium | t2.large
Komputasi yang dioptimalkan	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
Memori yang dioptimalkan	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

Kelas instans	Tipe instans
	r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge
Penyimpanan dioptimalkan	i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge | hs1.8xlarge | g2.2xlarge | g2.8xlarge | d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge

EC2 Instans Amazon yang Didukung untuk Cluster EMR Amazon

Tabel ini mencantumkan EC2 instans Amazon yang AWS Data Pipeline mendukung dan dapat membuat untuk klaster EMR Amazon, jika ditentukan. Untuk informasi selengkapnya, lihat Tipe instans yang didukung di Panduan Pengelolaan Amazon EMR.

Kelas instans	Tipe instans
Tujuan umum	m1.small | m1.medium | m1.large | m1.xlarge | m3.xlarge | m3.2xlarge
Komputasi yang dioptimalkan	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
Memori yang dioptimalkan	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
Penyimpanan dioptimalkan	h1.4xlarge | hs1.2xlarge | hs1.4xlarge| hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4large | i2.8xlarge | d2.xlarge | d2.2xlarge| d2.4xlarge | d2.8xlarge

Kelas instans	Tipe instans
Komputasi yang dipercepat	g2.2xlarge | cg1.4xlarge

AWS Data Pipeline Konsep

Sebelum Anda mulai, baca tentang konsep dan komponen kunci untuk AWS Data Pipeline.

Daftar Isi

- Definisi Alur
- Komponen Alur, Instans, dan Upaya
- Runner Tugas
- Simpul Data
- Basis Data
- Aktivitas
- Prasyarat
- Sumber daya
- Tindakan

Definisi Alur

Definisi pipeline adalah bagaimana Anda mengkomunikasikan logika bisnis Anda AWS Data Pipeline. Itu berisi informasi berikut:

- Nama, lokasi, dan format dari sumber data Anda
- Aktivitas yang mengubah data
- Jadwal untuk aktivitas tersebut
- Sumber daya yang menjalankan aktivitas dan prasyarat Anda
- Prasyarat yang harus dipenuhi sebelum aktivitas dapat dijadwalkan
- Cara untuk memberitahukan Anda dengan pembaruan status saat eksekusi alur berlangsung

Dari definisi pipeline Anda, AWS Data Pipeline tentukan tugas, jadwalkan, dan tetapkan tugas ke pelari tugas. Jika tugas tidak berhasil diselesaikan, AWS Data Pipeline coba ulang tugas sesuai dengan instruksi Anda dan, jika perlu, tetapkan kembali ke pelari tugas lain. Jika tugas gagal berulang kali, Anda dapat mengonfigurasi alur untuk memberitahu Anda.

Misalnya, dalam definisi alur, Anda dapat menentukan bahwa berkas log yang dihasilkan oleh aplikasi Anda diarsipkan setiap bulan pada tahun 2013 ke bucket Amazon S3. AWS Data Pipeline kemudian akan membuat 12 tugas, masing-masing menyalin lebih dari satu bulan data, terlepas dari apakah bulan tersebut berisi 30, 31, 28, atau 29 hari.

Anda dapat membuat definisi alur dengan cara berikut:

- Secara grafis, dengan menggunakan konsol AWS Data Pipeline
- Secara tekstual, dengan menulis file JSON dalam format yang digunakan oleh antarmuka baris perintah
- Secara terprogram, dengan memanggil layanan web dengan salah satu AWS SDKs atau API AWS Data Pipeline

Definisi alur dapat berisi jenis komponen berikut.

Komponen Alur

Simpul Data

Lokasi input data untuk tugas atau lokasi di mana data output akan disimpan.

Aktivitas

Definisi pekerjaan yang harus dilakukan terjadwal menggunakan sumber daya komputasi dan biasanya simpul data input dan output.

Prasyarat

Pernyataan bersyarat yang harus betul sebelum suatu tindakan dapat dijalankan.

Sumber daya

Sumber daya komputasi yang melakukan pekerjaan yang ditentukan oleh alur.

Tindakan

Tindakan yang terpicu saat kondisi tertentu terpenuhi, seperti kegagalan aktivitas.

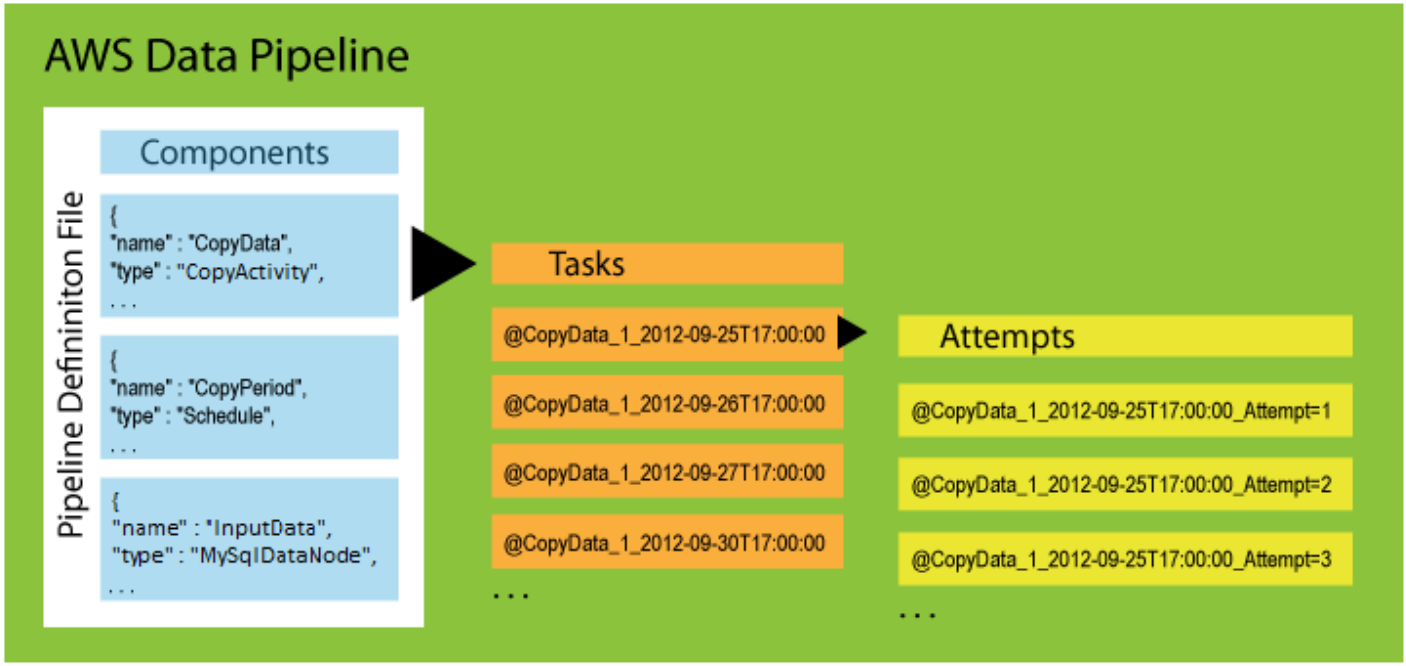
Untuk informasi selengkapnya, lihat Sintaks file definisi pipa.

Komponen Alur, Instans, dan Upaya

Ada tiga jenis item yang terkait dengan alur terjadwal:

- Komponen Alur— Komponen alur mewakili logika bisnis alur dan diwakili oleh bagian yang berbeda dari definisi alur. Komponen alur menentukan sumber data, aktivitas, jadwal, dan prasyarat alur kerja. Mereka dapat mewarisi sifat dari komponen induk. Hubungan antara komponen didefinisikan oleh referensi. Komponen alur menentukan aturan pengelolaan data.
- Instance — Saat AWS Data Pipeline menjalankan pipeline, ia mengkompilasi komponen pipeline untuk membuat satu set instance yang dapat ditindaklanjuti. Setiap instans berisi semua informasi untuk melakukan tugas tertentu. Set lengkap instance adalah daftar tugas dari pipeline. AWS Data Pipeline menyerahkan instance ke pelari tugas untuk diproses.
- Upaya — Untuk menyediakan pengelolaan data yang tangguh, AWS Data Pipeline mencoba ulang operasi yang gagal. Itu terus melakukannya hingga tugas mencapai jumlah maksimum upaya coba lagi yang diizinkan. Objek percobaan melacak berbagai upaya, hasil, dan alasan kegagalan jika dapat diaplikasikan. Pada dasarnya, ini adalah contoh dengan penghitung. AWS Data Pipeline melakukan percobaan ulang menggunakan sumber daya yang sama dari upaya sebelumnya, seperti cluster EC2 dan instance EMR Amazon.

> **Note**
>
> Mencoba kembali tugas yang gagal adalah bagian penting dari strategi toleransi kesalahan, dan definisi AWS Data Pipeline memberikan kondisi dan ambang batas untuk mengendalikan percobaan ulang. Namun, terlalu banyak percobaan ulang dapat menunda deteksi kegagalan yang tidak dapat dipulihkan karena AWS Data Pipeline tidak melaporkan kegagalan hingga semua percobaan ulang yang Anda tentukan telah habis. Percobaan ulang ekstra dapat dikenakan biaya tambahan jika dijalankan pada sumber daya AWS. Akibatnya, pertimbangkan dengan cermat kapan tepat untuk melampaui pengaturan AWS Data Pipeline default yang Anda gunakan untuk mengontrol percobaan ulang dan pengaturan terkait.

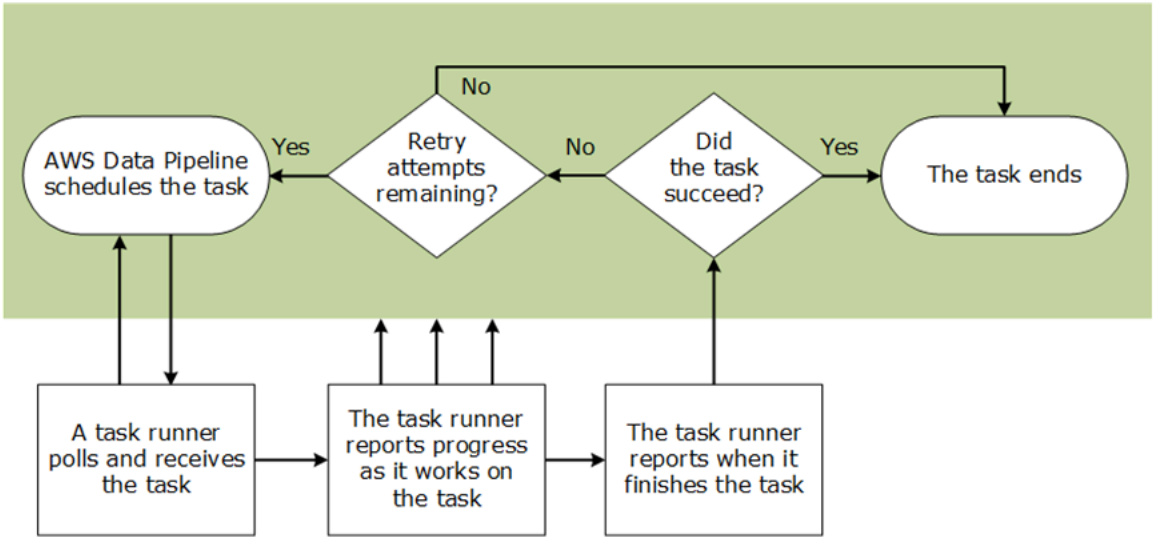


Runner Tugas

Task runner adalah aplikasi yang melakukan polling AWS Data Pipeline untuk tugas dan kemudian melakukan tugas-tugas tersebut.

Runner Tugas adalah implementasi default dari runner tugas yang disediakan oleh AWS Data Pipeline. Ketika Task Runner diinstal dan dikonfigurasi, ia melakukan polling AWS Data Pipeline untuk tugas-tugas yang terkait dengan pipeline yang telah Anda aktifkan. Saat tugas ditetapkan ke Runner Tugas, ia melakukan tugas itu dan melaporkan statusnya kembali ke AWS Data Pipeline.

Diagram berikut menggambarkan bagaimana AWS Data Pipeline dan task runner berinteraksi untuk memproses tugas terjadwal. Tugas adalah unit kerja terpisah yang dibagikan AWS Data Pipeline layanan dengan pelari tugas. Ini berbeda dari alur yang merupakan definisi umum dari aktivitas dan sumber daya yang biasanya menghasilkan beberapa tugas.



Ada dua cara Anda dapat menggunakan Runner Tugas untuk mengolah alur Anda:

- AWS Data Pipeline menginstal Task Runner untuk Anda pada sumber daya yang diluncurkan dan dikelola oleh layanan AWS Data Pipeline web.
- Anda menginstal Task Runner pada sumber daya komputasi yang Anda kelola, seperti EC2 instans yang berjalan lama, atau server lokal.

Untuk informasi lebih lanjut tentang bekerja dengan Runner Tugas, lihat Bekerja dengan Runner Tugas.

Simpul Data

Dalam AWS Data Pipeline, node data mendefinisikan lokasi dan jenis data yang digunakan aktivitas pipeline sebagai input atau output. AWS Data Pipeline mendukung jenis node data berikut:

Node Dinamo DBData

Tabel DynamoDB yang berisi data untuk HiveActivity atau EmrActivity untuk digunakan.

SqlDataNode

Tabel SQL dan kueri basis data yang mewakili data untuk aktivitas alur yang akan digunakan.

> **Note**
>
> Sebelumnya, MySqlDataNode digunakan. Gunakan SqlDataNode sebagai gantinya.

RedshiftDataNode

Tabel Amazon Redshift yang berisi data untuk digunakan RedshiftCopyActivity.

S3 DataNode

Lokasi Amazon S3 yang berisi satu atau beberapa file untuk digunakan aktivitas alur.

Basis Data

AWS Data Pipeline mendukung jenis database berikut:

JdbcDatabase

Sebuah basis data JDBC.

RdsDatabase

Sebuah basis data Amazon RDS.

RedshiftDatabase

Sebuah basis data Amazon Redshift.

Aktivitas

Dalam AWS Data Pipeline, aktivitas adalah komponen pipa yang mendefinisikan pekerjaan yang akan dilakukan. AWS Data Pipeline menyediakan beberapa aktivitas pra-paket yang mengakomodasi skenario umum, seperti memindahkan data dari satu lokasi ke lokasi lain, menjalankan kueri Hive, dan sebagainya. Aktivitas dapat diperluas, sehingga Anda dapat menjalankan skrip kustom Anda sendiri untuk mendukung kombinasi tanpa akhir.

AWS Data Pipeline mendukung jenis kegiatan berikut:

CopyActivity

Menyalin data dari satu lokasi ke lokasi lain.

EmrActivity

Menjalankan klaster Amazon EMR.

HiveActivity

Menjalankan kueri Hive pada klaster Amazon EMR.

HiveCopyActivity

Menjalankan kueri Hive di klaster Amazon EMR dengan dukungan untuk pemfilteran data tingkat lanjut dan dukungan untuk S3 DataNode dan Node Dinamo DBData.

PigActivity

Menjalankan skrip Pig di klaster Amazon EMR.

RedshiftCopyActivity

Menyalin data ke dan dari tabel Amazon Redshift.

ShellCommandActivity

Menjalankan perintah UNIX/Linux shell kustom sebagai aktivitas.

SqlActivity

Menjalankan kueri SQL pada basis data.

Beberapa aktivitas memiliki dukungan khusus untuk menyiapkan data dan tabel basis data. Untuk informasi selengkapnya, lihat Penahapan Data dan Tabel dengan Aktivitas Alur.

Prasyarat

Dalam AWS Data Pipeline, prasyarat adalah komponen pipeline yang berisi pernyataan bersyarat yang harus benar sebelum aktivitas dapat dijalankan. Misalnya, prasyarat dapat memeriksa apakah data sumber ada sebelum aktivitas pipeline mencoba menyalinnya. AWS Data Pipeline menyediakan beberapa prasyarat pra-paket yang mengakomodasi skenario umum, seperti apakah tabel database ada, apakah kunci Amazon S3 ada, dan sebagainya. Namun, prasyarat dapat diperluas dan memungkinkan Anda menjalankan skrip khusus Anda sendiri untuk mendukung kombinasi tanpa akhir.

Ada dua jenis prakondisi: prakondisi yang terkelola sistem dan prakondisi yang dikelola pengguna. Prasyarat yang dikelola sistem dijalankan oleh layanan AWS Data Pipeline web atas nama Anda dan tidak memerlukan sumber daya komputasi. Prasyarat yang dikelola pengguna hanya berjalan pada

sumber daya komputasi yang Anda tentukan menggunakan bidang `runsOn` atau `workerGroup`. Sumber daya `workerGroup` berasal dari aktivitas yang menggunakan prasyarat.

Prasyarat yang Dikelola Sistem

Dynamo Ada DBData

Periksa apakah data ada dalam tabel DynamoDB tertentu.

Dynamo Ada DBTable

Periksa apakah tabel DynamoDB ada.

S3 KeyExists

Periksa apakah kunci Amazon S3 ada.

S3 PrefixNotEmpty

Periksa apakah prefiks Amazon S3 kosong.

Prasyarat Dikelola Pengguna

Exists

Periksa apakah simpul data ada.

ShellCommandPrecondition

Menjalankan perintah Unix/Linux shell kustom sebagai prasyarat.

Sumber daya

Dalam AWS Data Pipeline, sumber daya adalah sumber daya komputasi yang melakukan pekerjaan yang ditentukan oleh aktivitas pipa. AWS Data Pipeline mendukung jenis sumber daya berikut:

Ec2Resource

EC2 Instance yang melakukan pekerjaan yang ditentukan oleh aktivitas pipeline.

EmrCluster

Klaster Amazon EMR yang melakukan pekerjaan yang ditentukan oleh aktivitas alur, seperti EmrActivity.

Resource dapat berjalan di wilayah yang sama dengan set data kerjanya, bahkan wilayah yang berbeda dari AWS Data Pipeline. Untuk informasi selengkapnya, lihat Menggunakan Alur dengan Sumber Daya di Beberapa Wilayah.

Batasan sumber daya

AWS Data Pipeline skala untuk mengakomodasi sejumlah besar tugas bersamaan dan Anda dapat mengonfigurasinya untuk secara otomatis membuat sumber daya yang diperlukan untuk menangani beban kerja yang besar. Sumber daya yang dibuat secara otomatis ini berada di bawah kendali Anda dan memperhitungkan batas sumber daya akun AWS Anda. Misalnya, jika Anda mengonfigurasi AWS Data Pipeline untuk membuat klaster EMR Amazon 20-node secara otomatis untuk memproses data dan akun AWS Anda memiliki batas instans EC2 yang disetel ke 20, Anda mungkin secara tidak sengaja menghabiskan sumber daya pengisian ulang yang tersedia. Sebagai hasilnya, pertimbangkan pembatasan sumber daya ini dalam desain Anda atau tingkatkan batas akun Anda dengan sesuai. Untuk informasi selengkapnya tentang kuota layanan, lihat Kuota Layanan AWS di Referensi Umum AWS.

> **Note**
>
> Batasnya adalah satu instans per objek komponen `Ec2Resource`.

Platform yang Didukung

Alur dapat meluncurkan sumber daya Anda ke platform berikut:

EC2-Klasik

Sumber daya Anda berjalan dalam satu jaringan datar tunggal yang Anda bagikan dengan pelanggan lain.

EC2-VPC

Sumber daya Anda berjalan di virtual private cloud (VPC) yang secara logis diisolasi ke akun AWS Anda.

Akun AWS Anda dapat meluncurkan sumber daya baik ke kedua platform atau hanya ke EC2 -VPC, berdasarkan wilayah menurut wilayah. Untuk informasi selengkapnya, lihat Platform yang Didukung di Panduan EC2 Pengguna Amazon.

Jika akun AWS Anda hanya mendukung EC2 -VPC, kami membuat VPC default untuk Anda di setiap Wilayah AWS. Secara default, kami meluncurkan sumber daya Anda ke subnet default VPC default Anda. Atau, Anda dapat membuat VPC non-default dan menentukan salah satu subnetnya saat Anda mengonfigurasi sumber daya, lalu kami meluncurkan sumber daya Anda ke subnet tertentu dari VPC non-default.

Saat Anda meluncurkan instans ke VPC, Anda harus menentukan grup keamanan yang dibuat khusus untuk VPC tersebut. Anda tidak dapat menentukan grup keamanan yang Anda buat untuk EC2 -Classic saat meluncurkan instance ke VPC. Selain itu, Anda harus menggunakan ID grup keamanan dan bukan nama grup keamanan untuk mengidentifikasi grup keamanan untuk VPC.

Instans Amazon EC2 Spot dengan Cluster EMR Amazon dan AWS Data Pipeline

Pipeline dapat menggunakan Instans EC2 Spot Amazon untuk node tugas di sumber daya kluster Amazon EMR mereka. Secara default, alur menggunakan Instans Sesuai Permintaan. Instans Spot memungkinkan Anda menggunakan EC2 instance cadangan dan menjalankannya. Model harga Instans Spot melengkapi model harga Instans Cadangan dan Sesuai Permintaan, yang berpotensi memberikan opsi paling hemat biaya untuk memperoleh kapasitas komputasi, bergantung pada aplikasi Anda. Untuk informasi selengkapnya, lihat halaman produk Instans EC2 Spot Amazon.

Saat menggunakan Instans Spot, AWS Data Pipeline kirimkan harga maksimum Instans Spot ke EMR Amazon saat klaster diluncurkan. Secara otomatis mengalokasikan pekerjaan cluster ke jumlah node tugas Spot Instance yang Anda tentukan menggunakan bidang. `taskInstanceCount` AWS Data Pipeline membatasi Instans Spot untuk node tugas untuk memastikan bahwa node inti sesuai permintaan tersedia untuk menjalankan pipeline Anda.

Anda dapat mengedit instans sumber daya alur yang gagal atau selesai untuk menambahkan Instans Spot. Saat alur meluncurkan klaster kembali, ia menggunakan Instans Spot untuk simpul tugas.

Pertimbangan Instans Spot

Saat Anda menggunakan Instans Spot dengan AWS Data Pipeline, pertimbangan berikut berlaku:

- Instans Spot Anda dapat berakhir ketika harga Instans Spot melebihi harga maksimum untuk instans tersebut, atau karena alasan EC2 kapasitas Amazon. Namun, Anda tidak kehilangan data karena AWS Data Pipeline menggunakan cluster dengan node inti yang selalu Instans Sesuai Permintaan dan tidak tunduk pada penghentian.

- Instans Spot dapat memerlukan lebih banyak waktu untuk memulai karena mereka memenuhi kapasitas secara asinkron. Oleh karena itu, alur Instans Spot dapat berjalan lebih lambat daripada alur Instans Sesuai Permintaan yang setara.
- Klaster Anda mungkin tidak berjalan jika Anda tidak menerima Instans Spot, seperti saat harga maksimum Anda terlalu rendah.

Tindakan

AWS Data Pipeline Tindakan adalah langkah-langkah yang diambil komponen pipeline ketika peristiwa tertentu terjadi, seperti keberhasilan, kegagalan, atau aktivitas yang terlambat. Bidang peristiwa dari suatu aktivitas mengambil referensi pada suatu tindakan, seperti referensi ke `snsalarm` di bidang `onLateAction` dari `EmrActivity`.

AWS Data Pipeline bergantung pada notifikasi Amazon SNS sebagai cara utama untuk menunjukkan status jaringan pipa dan komponennya dengan cara yang tidak dijaga. Untuk informasi selengkapnya, lihat Amazon SNS. Selain pemberitahuan SNS, Anda dapat menggunakan AWS Data Pipeline konsol dan CLI untuk mendapatkan informasi status pipeline.

AWS Data Pipeline mendukung tindakan berikut:

SnsAlarm

Tindakan yang mengirimkan notifikasi SNS ke topik berdasarkan peristiwa `onSuccess`, `OnFail`, dan `onLateAction`.

Mengakhiri

Tindakan yang memicu pembatalan aktivitas, sumber daya, atau simpul data yang tertunda atau belum selesai. Anda tidak dapat mengakhiri tindakan yang menyertakan `onSuccess`, `OnFail`, atau `onLateAction`.

Pemantauan Alur Proaktif

Cara terbaik untuk mendeteksi masalah adalah dengan memantau alur Anda secara proaktif sejak awal. Anda dapat mengonfigurasi komponen pipeline untuk memberi tahu Anda tentang situasi atau peristiwa tertentu, seperti ketika komponen pipeline gagal atau tidak dimulai dengan waktu mulai yang dijadwalkan. AWS Data Pipeline memudahkan untuk mengonfigurasi notifikasi dengan menyediakan bidang peristiwa pada komponen pipeline yang dapat Anda kaitkan dengan notifikasi Amazon SNS, seperti, `onSuccessOnFail`, dan. `onLateAction`

Menyiapkan untuk AWS Data Pipeline

Sebelum Anda menggunakan AWS Data Pipeline untuk pertama kalinya, selesaikan tugas-tugas berikut.

Tugas

- Mendaftar untuk AWS
- Buat Peran IAM untuk AWS Data Pipeline dan Pipeline Resources
- Izinkan IAM utama (Pengguna dan Grup) untuk Melakukan Tindakan yang Diperlukan
- Memberikan akses terprogram

Setelah Anda menyelesaikan tugas-tugas ini, Anda dapat mulai menggunakan AWS Data Pipeline. Untuk tutorial, basic Memulai dengan AWS Data Pipeline.

Mendaftar untuk AWS

Saat Anda mendaftar ke Amazon Web Services (AWS), akun AWS Anda secara otomatis mendaftar untuk semua layanan di AWS, termasuk AWS Data Pipeline. Anda hanya membayar biaya layanan yang Anda gunakan. Untuk informasi selengkapnya tentang tingkat AWS Data Pipeline penggunaan, lihat AWS Data Pipeline.

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.
2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Buat Peran IAM untuk AWS Data Pipeline dan Pipeline Resources

AWS Data Pipeline memerlukan peran IAM yang menentukan izin untuk melakukan tindakan dan mengakses AWS sumber daya. Peran pipeline menentukan izin yang AWS Data Pipeline dimiliki, dan peran sumber daya menentukan izin yang dimiliki aplikasi yang berjalan pada sumber daya pipeline, seperti EC2 instance. Anda menentukan peran ini saat Anda membuat alur. Meskipun Anda tidak menentukan peran khusus dan menggunakan peran default `DataPipelineDefaultRole` dan `DataPipelineDefaultResourceRole`, Anda harus terlebih dahulu membuat peran dan melampirkan kebijakan izin. Untuk informasi selengkapnya, lihat Peran IAM untuk AWS Data Pipeline.

Izinkan IAM utama (Pengguna dan Grup) untuk Melakukan Tindakan yang Diperlukan

Untuk bekerja dengan alur, IAM utama (pengguna atau grup) di akun Anda harus diizinkan untuk melakukan tindakan AWS Data Pipeline yang diperlukan dan tindakan untuk layanan lain seperti yang ditentukan oleh alur Anda.

Untuk menyederhanakan izin, kebijakan AWSDataPipeline_FullAccessterkelola tersedia bagi Anda untuk dilampirkan ke prinsipal IAM. Kebijakan terkelola ini memungkinkan prinsipal untuk melakukan semua tindakan yang diperlukan pengguna dan `iam:PassRole` tindakan pada peran default yang digunakan AWS Data Pipeline saat peran kustom tidak ditentukan.

Kami sangat merekomendasikan agar Anda mengevaluasi kebijakan terkelola ini dengan cermat dan membatasi izin hanya untuk izin yang diperlukan pengguna Anda. Jika perlu, gunakan kebijakan ini sebagai titik awal, lalu hapus izin untuk membuat kebijakan izin sebaris yang lebih ketat yang dapat Anda lampirkan ke IAM utama. Untuk informasi selengkapnya dan contoh kebijakan izin, lihat Contoh Kebijakan untuk AWS Data Pipeline

Pernyataan kebijakan yang mirip dengan contoh berikut harus disertakan dalam kebijakan yang dilampirkan ke setiap IAM utama yang menggunakan alur. Pernyataan ini memungkinkan IAM utama untuk melakukan tindakan `PassRole` pada peran yang digunakan alur. Jika Anda tidak menggunakan peran default, ganti *`MyPipelineRole`* dan *`MyResourceRole`* dengan peran khusus yang Anda buat.

JSON

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

Prosedur berikut menunjukkan cara membuat grup IAM, melampirkan kebijakan AWSDataPipeline_FullAccessterkelola ke grup, dan kemudian menambahkan pengguna ke grup. Anda dapat menggunakan prosedur ini untuk kebijakan sebaris apa pun

Untuk membuat grup pengguna **`DataPipelineDevelopers`** dan melampirkan AWSDataPipeline_FullAccesskebijakan

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Dalam panel navigasi, pilih Grup, Buat Grup Baru.

3. Masukkan Nama grup, sebagai contoh, **`DataPipelineDevelopers`**, dan kemudian pilih Langkah Selanjutnya.
4. Masukkan **`AWSDataPipeline_FullAccess`** untuk Filter dan kemudian pilih itu dari daftar.
5. Pilih Langkah Berikutnya dan pilih Buat Grup.
6. Untuk menambahkan pengguna ke grup:
   a. Pilih grup yang Anda buat dari daftar grup.
   b. Pilih Tindakan Grup, Tambahkan Pengguna ke Grup.
   c. Pilih pengguna yang ingin Anda tambahkan dari daftar, lalu pilih Tambahkan Pengguna ke Grup.

Memberikan akses terprogram

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. AWS Management Console Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Disarankan) Gunakan kredenal konsol sebagai kredenial sementara untuk menandatangani permintaan terprogram ke,, atau. AWS CLI AWS SDKs AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk itu AWS CLI, lihat Login untuk pengembangan AWS lokal di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, lihat Login untuk pengembangan AWS lokal di Panduan

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
		Referensi Alat AWS SDKs dan Alat.
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, alat, dan AWS APIs, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi Alat AWS SDKs dan Alat.
IAM	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk dalam Menggunakan kredensil sementara dengan AWS sumber daya di Panduan Pengguna IAM.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandata ngani permintaan terprogra m ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi menggunakan kredensil pengguna IAM di Panduan Pengguna.AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensi jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Memulai dengan AWS Data Pipeline

AWS Data Pipeline membantu Anda mengurutkan, menjadwalkan, menjalankan, dan mengelola beban kerja pemrosesan data berulang dengan andal dan hemat biaya. Layanan ini memudahkan Anda untuk merancang aktivitas extract-transform-load (ETL) menggunakan data terstruktur dan tidak terstruktur, baik lokal maupun di cloud, berdasarkan logika bisnis Anda.

Untuk menggunakannya AWS Data Pipeline, Anda membuat definisi pipeline yang menentukan logika bisnis untuk pemrosesan data Anda. Definisi pipa tipikal terdiri dari aktivitas yang menentukan pekerjaan yang akan dilakukan, dan node data yang menentukan lokasi dan jenis data input dan output.

Dalam tutorial ini, Anda menjalankan skrip perintah shell yang menghitung jumlah permintaan GET di log server web Apache. Alur ini berjalan setiap 15 menit selama satu jam, dan menulis output ke Amazon S3 pada setiap iterasi.

Prasyarat

Sebelum Anda memulai, selesaikan tugas di Menyiapkan untuk AWS Data Pipeline.

Objek Alur

Alur menggunakan objek berikut:

ShellCommandActivity

Membaca berkas log input dan menghitung jumlah kesalahan.

S3 DataNode (input)

Bucket S3 yang berisi berkas log input.

S3 DataNode (output)

Bucket S3 untuk output.

Ec2Resource

Sumber daya komputasi yang AWS Data Pipeline digunakan untuk melakukan aktivitas.

Perhatikan bahwa jika Anda memiliki sejumlah besar data file log, Anda dapat mengonfigurasi pipeline Anda untuk menggunakan klaster EMR untuk memproses file, bukan instance EC2 .

Jadwal

Mendefinisikan bahwa aktivitas tersebut dilakukan setiap 15 menit selama satu jam.

Tugas

- Buat Alur
- Memantau Alur Berjalan
- Lihat Output
- Hapus Alur

Buat Alur

Cara tercepat untuk memulai AWS Data Pipeline adalah dengan menggunakan definisi pipeline yang disebut template.

Untuk membuat alur

1. Buka AWS Data Pipeline konsol di https://console.aws.amazon.com/datapipeline/.
2. Dari bilah navigasi, pilih wilayah. Anda dapat memilih wilayah mana pun yang tersedia untuk Anda, di mana pun lokasi Anda. Banyak sumber daya AWS khusus untuk suatu wilayah, tetapi AWS Data Pipeline memungkinkan Anda untuk menggunakan sumber daya yang berada di wilayah yang berbeda dari pipeline.
3. Layar pertama yang Anda lihat bergantung pada apakah Anda telah membuat alur di wilayah saat ini.
   a. Jika Anda belum membuat alur di wilayah ini, konsol tersebut akan menampilkan layar perkenalan. Pilih Mulai Sekarang.
   b. Jika Anda telah membuat alur di wilayah ini, konsol akan menampilkan halaman yang mencantumkan alur Anda untuk wilayah tersebut. Pilih Buat alur baru.
4. Di Nama, masukkan nama untuk alur Anda.
5. (Opsional) Di Deskripsi, masukkan deskripsi untuk alur Anda.
6. Untuk Sumber, pilih Bangun menggunakan templat, lalu pilih templat berikut: Memulai menggunakan ShellCommandActivity.

7. Di bawah bagian Parameter, yang terbuka saat Anda memilih templat, biarkan folder input S3 dan perintah Shell untuk dijalankan dengan nilai defaultnya. Klik ikon folder di sebelah folder output S3, pilih salah satu bucket atau folder Anda, lalu klik Pilih.
8. Di bawah Jadwal, biarkan nilai default. Saat Anda mengaktifkan alur, alur mulai berjalan, dan kemudian lanjutkan setiap 15 menit selama satu jam.

   Jika mau, Anda dapat memilih Jalankan sekali pada aktivasi alur sebagai gantinya.
9. Di bawah Konfigurasi Alur, biarkan pencatatan diaktifkan. Pilih ikon folder di bawah lokasi S3 untuk log, pilih salah satu bucket atau folder Anda, lalu pilih Pilih.

   Jika mau, Anda dapat menonaktifkan pencatatan sebagai gantinya.
10. Di bawah Keamanan/Akses, biarkan IAM role diatur ke Default.
11. Klik Aktifkan.

    Jika Anda mau, Anda dapat memilih Edit di Arsitek untuk memodifikasi alur ini. Misalnya, Anda dapat menambahkan prasyarat.

Memantau Alur Berjalan

Setelah Anda mengaktifkan alur Anda, Anda akan dibawa ke halaman Detail eksekusi di mana Anda dapat memantau kemajuan alur Anda.

Untuk memantau kemajuan alur Anda

1. Klik Perbarui atau tekan F5 untuk memperbarui status yang ditampilkan.

   > **Tip**
   >
   > Jika tidak ada proses berjalan yang terdaftar, pastikan bahwa Mulai (dalam UTC) dan Akhir (dalam UTC) mencakup awal dan akhir yang dijadwalkan dari alur Anda, lalu klik Perbarui.
2. Ketika status setiap objek dalam alur Anda adalah `FINISHED`, alur Anda telah berhasil menyelesaikan tugas yang dijadwalkan.
3. Jika alur Anda tidak berhasil diselesaikan, periksa pengaturan alur Anda untuk masalah. Untuk informasi selengkapnya tentang pemecahan masalah yang gagal atau tidak lengkapnya proses instans dari alur Anda, lihat Menyelesaikan Masalah Umum.

Lihat Output

Buka konsol Amazon S3 dan navigasikan ke bucket Anda. Jika Anda menjalankan alur Anda setiap 15 menit selama satu jam, Anda akan melihat empat subfolder yang diberi stempel waktu. Setiap subfolder berisi output dalam file dengan nama `output.txt`. Karena kami menjalankan skrip pada file input yang sama setiap kali, file output menjadi identik.

Hapus Alur

Untuk berhenti dikenakan biaya, hapus alur Anda. Menghapus alur Anda akan menghapus definisi alur dan semua objek terkait.

Untuk menghapus alur Anda

1. Pada halaman Daftar Alur, pilih alur Anda.
2. Klik Tindakan, lalu pilih Hapus.
3. Saat diminta konfirmasi, pilih Hapus.

Jika Anda selesai dengan output dari tutorial ini, hapus folder output dari bucket Amazon S3 Anda.

Bekerja dengan jaringan pipa

Anda dapat mengelola, membuat, dan memodifikasi pipeline menggunakan antarmuka baris perintah (CLI) atau SDK. AWS Bagian berikut memperkenalkan konsep AWS Data Pipeline mendasar dan menunjukkan cara bekerja dengan alur.

> **Important**
>
> Sebelum Anda memulai, lihat Menyiapkan untuk AWS Data Pipeline.

Daftar Isi

- Membuat pipa
- Melihat Alur Anda
- Mengedit Alur Anda
- Mengkloning Alur Anda
- Menandai Alur Anda
- Menonaktifkan Alur Anda
- Menghapus Alur Anda
- Penahapan Data dan Tabel dengan Aktivitas Alur
- Menggunakan Alur dengan Sumber Daya di Beberapa Wilayah
- Kegagalan dan tayangan ulang yang berulang
- Sintaks file definisi pipa
- Bekerja dengan API

Membuat pipa

AWS Data Pipeline menyediakan beberapa cara bagi Anda untuk membuat saluran pipa:

- Gunakan AWS Command Line Interface (CLI) dengan template yang disediakan untuk kenyamanan Anda. Untuk informasi selengkapnya, lihat Buat pipeline dari template Data Pipeline menggunakan CLI.

- Gunakan AWS Command Line Interface (CLI) dengan file definisi pipeline dalam format JSON.
- Menggunakan AWS SDK dengan API spesifik bahasa. Untuk informasi selengkapnya, lihat Bekerja dengan API.

Buat pipeline dari template Data Pipeline menggunakan CLI

Data Pipeline menyediakan beberapa definisi pipeline yang telah dikonfigurasi sebelumnya, yang dikenal sebagai template. Anda dapat menggunakan template untuk memulai dengan AWS Data Pipeline cepat. Template ini tersedia dalam ember publik di lokasi Amazon S3:. `s3://datapipeline-us-east-1/templates/` Template yang telah ditentukan ini dibuat untuk mencapai kasus penggunaan tertentu dan dapat digunakan untuk membuat saluran pipa. Anda dapat menggunakan `aws s3 ls --recursive "s3://datapipeline-us-east-1/templates/"` untuk daftar semua template yang tersedia.

Buat pipeline dari template menggunakan CLI

Misalnya Anda ingin membuat pipeline yang mengekspor tabel DynamoDB ke Amazon S3. Template yang akan digunakan dalam kasus ini dapat ditemukan di:`s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json`.

Untuk mengunduh template JSON dan membuat pipeline menggunakan CLI

1. Unduh template menggunakan `aws s3 cp` CLI atau curl. Contoh:

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

2. Buat perubahan pada template yang diunduh sesuai kebutuhan. Misalnya, untuk menggunakan versi rilis EMR terbaru, ubah `releaseLabel` bidang dalam `EmrClusterForBackup` objek, ubah jenis instance master dan inti, dan ubah nilai default parameter dalam templat.
3. Buat pipeline menggunakan `create-pipeline` CLI. Contoh:

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

4. Perhatikan ID pipeline yang dibuat.
5. Gunakan `put-pipeline-definition` untuk mengunggah definisi. Berikan nilai parameter yang nilai defaultnya ingin Anda timpa menggunakan `--parameter-values` opsi.

Untuk informasi lebih lanjut tentang templat, lihat Pilih templat.

Pilih templat

Template berikut tersedia untuk diunduh dari bucket Amazon S3:. `s3://datapipeline-us-east-1/templates/`

Template

- Memulai menggunakan ShellCommandActivity
- Jalankan AWS perintah CLI
- Ekspor tabel DynamoDB ke S3
- Impor data cadangan DynamoDB dari S3
- Jalankan pekerjaan di klaster EMR Amazon
- Salinan lengkap Amazon RDS MySQL Table ke Amazon S3
- Salinan tambahan tabel Amazon RDS MySQL ke Amazon S3
- Muat data S3 ke tabel Amazon RDS MySQL
- Salinan lengkap tabel Amazon RDS MySQL ke Amazon Redshift
- Salinan tambahan tabel MySQL Amazon RDS ke Amazon Redshift
- Muat data dari Amazon S3 ke Amazon Redshift

Memulai menggunakan ShellCommandActivity

ShellCommandActivityTemplate Getting Started menggunakan menjalankan skrip perintah shell untuk menghitung jumlah permintaan GET dalam file log. Keluarannya ditulis di lokasi Amazon S3 yang diberi stempel waktu pada setiap penjalanan alur yang dijadwalkan.

Templat menggunakan objek alur berikut:

- ShellCommandActivity
- S3 InputNode
- S3 OutputNode
- Ec2Resource

Jalankan AWS perintah CLI

Template ini menjalankan AWS CLI perintah yang ditentukan pengguna pada interval terjadwal.

Ekspor tabel DynamoDB ke S3

Templat Ekspor tabel DynamoDB ke S3 menjadwalkan klaster Amazon EMR untuk mengekspor data dari tabel DynamoDB ke bucket Amazon S3. Templat ini menggunakan klaster Amazon EMR, yang ukurannya proporsional dengan nilai throughput yang tersedia untuk tabel DynamoDB. Meskipun Anda dapat meningkatkan IOPs di atas meja, ini dapat menimbulkan biaya tambahan saat mengimpor dan mengekspor. Sebelumnya, ekspor menggunakan HiveActivity tetapi sekarang menggunakan asli MapReduce.

Templat menggunakan objek alur berikut:

- EmrActivity
- EmrCluster
- Node Dinamo DBData
- S3 DataNode

Impor data cadangan DynamoDB dari S3

Templat Impor data cadangan DynamoDB dari S3 menjadwalkan klaster Amazon EMR untuk memuat cadangan DynamoDB yang dibuat sebelumnya di Amazon S3 ke tabel DynamoDB. Item yang ada dalam tabel DynamoDB diperbarui dengan item dari data cadangan dan item baru ditambahkan ke tabel. Templat ini menggunakan klaster Amazon EMR, yang ukurannya proporsional dengan nilai throughput yang tersedia untuk tabel DynamoDB. Meskipun Anda dapat meningkatkan IOPs di atas meja, ini dapat menimbulkan biaya tambahan saat mengimpor dan mengekspor. Sebelumnya, import menggunakan a HiveActivity tetapi sekarang menggunakan native MapReduce.

Templat menggunakan objek alur berikut:

- EmrActivity
- EmrCluster
- Node Dinamo DBData
- S3 DataNode
- S3 PrefixNotEmpty

Jalankan pekerjaan di klaster EMR Amazon

Template Run Job pada Elastic MapReduce Cluster meluncurkan klaster EMR Amazon berdasarkan parameter yang disediakan dan mulai menjalankan langkah-langkah berdasarkan jadwal yang ditentukan. Setelah tugas selesai, klaster EMR dihentikan. Tindakan bootstrap opsional dapat ditentukan untuk menginstal perangkat lunak tambahan atau untuk mengubah konfigurasi aplikasi pada klaster.

Templat menggunakan objek alur berikut:

- EmrActivity
- EmrCluster

Salinan lengkap Amazon RDS MySQL Table ke Amazon S3

Templat Salinan Lengkap Tabel MySQL Amazon RDS ke Amazon S3 menyalin seluruh tabel MySQL Amazon RDS dan menyimpan keluaran di lokasi Amazon S3. Keluaran disimpan sebagai file CSV dalam subfolder yang diberi stempel waktu di bawah lokasi Amazon S3 yang ditentukan.

Templat menggunakan objek alur berikut:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Salinan tambahan tabel Amazon RDS MySQL ke Amazon S3

Templat Salinan Tambahan Tabel MySQL Amazon RDS ke Amazon S3 melakukan penyalinan tambahan data dari tabel MySQL Amazon RDS dan menyimpan keluaran di lokasi Amazon S3. Tabel MySQL Amazon RDS harus memiliki kolom Terakhir Dimodifikasi.

Templat ini menyalin perubahan yang dibuat ke tabel di antara interval terjadwal mulai dari waktu mulai terjadwal. Jenis jadwal adalah deret waktu jadi jika salinan dijadwalkan untuk jam tertentu, AWS Data Pipeline salin baris tabel yang memiliki cap waktu Modifikasi Terakhir yang jatuh dalam satu jam. Penghapusan fisik ke tabel tidak disalin. Keluarannya ditulis dalam subfolder yang diberi stempel waktu di bawah lokasi Amazon S3 pada setiap penjalanan yang dijadwalkan.

Templat menggunakan objek alur berikut:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Muat data S3 ke tabel Amazon RDS MySQL

Template Load S3 Data ke RDS MySQL Table menjadwalkan EC2 instance Amazon untuk menyalin file CSV dari jalur file Amazon S3 yang ditentukan di bawah ini ke tabel Amazon RDS MySQL. File CSV tidak boleh memiliki baris header. Templat memperbarui entri yang ada di tabel MySQL Amazon RDS dengan entri di data Amazon S3 dan menambahkan entri baru dari data Amazon S3 ke tabel MySQL Amazon RDS. Anda dapat memuat data ke dalam tabel yang sudah ada atau menyediakan kueri SQL untuk membuat tabel baru.

Templat menggunakan objek alur berikut:

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Templat Amazon RDS ke Amazon Redshift

Dua templat berikut menyalin tabel dari MySQL Amazon RDS ke Amazon Redshift menggunakan skrip terjemahan, yang membuat tabel Amazon Redshift menggunakan skema tabel sumber dengan peringatan berikut:

- Jika kunci distribusi tidak ditentukan, kunci primer pertama dari tabel Amazon RDS ditetapkan sebagai kunci distribusi.
- Anda tidak dapat melewati kolom yang ada di tabel MySQL Amazon RDS saat Anda melakukan penyalinan ke Amazon Redshift.
- (Opsional) Anda dapat menyediakan pemetaan tipe data kolom MySQL Amazon RDS ke Amazon Redshift sebagai salah satu parameter dalam templat. Jika ini ditentukan, skrip akan menggunakan ini untuk membuat tabel Amazon Redshift.

Jika mode penyisipan Amazon Redshift `Overwrite_Existing` sedang digunakan:

- Jika kunci distribusi tidak disediakan, kunci primer pada tabel MySQL Amazon RDS akan digunakan.
- Jika ada kunci primer komposit pada tabel, yang pertama digunakan sebagai kunci distribusi jika kunci distribusi tidak disediakan. Hanya kunci komposit pertama yang ditetapkan sebagai kunci primer di tabel Amazon Redshift.
- Jika kunci distribusi tidak disediakan dan tidak ada kunci primer pada tabel MySQL Amazon RDS, operasi penyalinan gagal.

Untuk informasi selengkapnya tentang Amazon Redshift, lihat topik berikut:

- Cluster Amazon Redshift
- SALINAN Amazon Redshift
- Gaya distribusi dan contoh DISTKEY
- Sortir Tombol

Tabel berikut menjelaskan bagaimana skrip menerjemahkan tipe data:

Terjemahan tipe data antara MySQL dan Amazon Redshift

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
TINYINT, TINYINT (ukuran)	SMALLINT	MySQL: -128 hingga 127. Jumlah digit maksimum dapat ditentukan dalam tanda kurung. Amazon Redshift:. INT2 Bilangan bulat dua byte bertanda
TINYINT UNSIGNED, TINYINT (ukuran) UNSIGNED	SMALLINT	MySQL: 0 hingga 255 UNSIGNED. Jumlah digit maksimum dapat ditentukan dalam tanda kurung.

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
		Amazon Redshift:. INT2 Bilangan bulat dua byte bertanda
SMALLINT, SMALLINT(ukuran)	SMALLINT	MySQL: -32768 hingga 32767 normal. Jumlah digit maksimum dapat ditentukan dalam tanda kurung. Amazon Redshift:. INT2 Bilangan bulat dua byte bertanda
SMALLINT UNSIGNED, SMALLINT(ukuran) UNSIGNED,	INTEGER	MySQL: 0 hingga 65535 UNSIGNED*. Jumlah digit maksimum dapat ditentukan dalam tanda kurung Amazon Redshift:. INT4 Bilangan bulat empat byte bertanda
MEDIUMINT, MEDIUMINT(ukuran)	INTEGER	MySQL: 388608 hingga 8388607. Jumlah digit maksimum dapat ditentukan dalam tanda kurung Amazon Redshift:. INT4 Bilangan bulat empat byte bertanda

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
MEDIUMINT UNSIGNED, MEDIUMINT(ukuran) UNSIGNED	INTEGER	MySQL: 0 hingga 16777215. Jumlah digit maksimum dapat ditentukan dalam tanda kurung Amazon Redshift:. INT4 Bilangan bulat empat byte bertanda
INT, INT(ukuran)	INTEGER	MySQL: 147483648 hingga 2147483647 Amazon Redshift:. INT4 Bilangan bulat empat byte bertanda
INT UNSIGNED, INT(ukuran) UNSIGNED	BIGINT	MySQL: 0 hingga 4294967295 Amazon Redshift:. INT8 Bilangan bulat delapan byte bertanda
BIGINT BIGINT(ukuran)	BIGINT	Amazon Redshift:. INT8 Bilangan bulat delapan byte bertanda
BIGINT UNSIGNED BIGINT(ukuran) UNSIGNED	VARCHAR(20*4)	MySQL: 0 hingga 18446744073709551615 Amazon Redshift: Tidak ada padanan asli, sehingga menggunakan array char.

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
FLOAT FLOAT(ukuran,d) FLOAT(ukuran,d) UNSIGNED	REAL	Jumlah digit maksimum dapat ditentukan dalam parameter ukuran. Jumlah digit maksimum di sebelah kanan titik desimal ditentukan dalam parameter d. Pergeseran Merah Amazon: FLOAT4
DOUBLE(ukuran,d)	DOUBLE PRECISION	Jumlah digit maksimum dapat ditentukan dalam parameter ukuran. Jumlah digit maksimum di sebelah kanan titik desimal ditentukan dalam parameter d. Pergeseran Merah Amazon: FLOAT8
DECIMAL(ukuran,d)	DECIMAL(ukuran,d)	DOUBLE disimpan sebagai string, memungkinkan untuk titik desimal tetap. Jumlah digit maksimum dapat ditentuka n dalam parameter ukuran. Jumlah digit maksimum di sebelah kanan titik desimal ditentukan dalam parameter d. Amazon Redshift: Tidak ada padanan asli.

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
CHAR(ukuran)	VARCHAR(ukuran*4)	Memegang string dengan panjang tetap, yang dapat berisi huruf, angka, dan karakter khusus. Ukuran tetap ditentukan sebagai parameter dalam tanda kurung. Dapat menyimpan hingga 255 karakter. Kanan diisi dengan spasi. Amazon Redshift: Tipe data CHAR tidak mendukung karakter multibyte sehingga VARCHAR digunakan. Jumlah maksimum byte per karakter adalah 4 menurut RFC3629, yang membatasi tabel karakter ke U+10FFFF.
VARCHAR(ukuran)	VARCHAR(ukuran*4)	Dapat menyimpan hingga 255 karakter. VARCHAR tidak mendukung poin kode UTF-8 yang tidak valid berikut: 0xD800-0xDFFF, (Urutan byte: ED A0 80- ED BF BF), 0xFDD0-0xFDEF, 0xFFFE, dan 0xFFFF, (Urutan byte: EF B7 90- EF B7 AF , EF BF BE, dan EF BF BF)

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
TINYTEXT	VARCHAR(255*4)	Memegang string dengan panjang maksimum 255 karakter
TEXT	VARCHAR(max)	Memegang string dengan panjang maksimum 65.535 karakter.
MEDIUMTEXT	VARCHAR(max)	0 hingga 16.777.215 Karakter
LONGTEXT	VARCHAR(max)	0 hingga 4.294.967.295 Karakter
BOOLEAN BOOL TINYINT(1)	BOOLEAN	MySQL: Jenis ini adalah sinonim untuk TINYINT(1). Nilai nol dianggap salah. Nilai bukan nol dianggap benar.
BINARY[(M)]	varchar(255)	M adalah 0 hingga 255 byte, TETAP
VARBINARY(M)	VARCHAR(max)	0 hingga 65.535 byte
TINYBLOB	VARCHAR(255)	0 hingga 255 byte
BLOB	VARCHAR(max)	0 hingga 65.535 byte
MEDIUMBLOB	VARCHAR(max)	0 hingga 16.777.215 byte
LONGBLOB	VARCHAR(max)	0 hingga 4.294.967.295 byte
ENUM	VARCHAR(255*2)	Batasnya bukan pada panjang string enum literal, melainkan pada definisi tabel untuk jumlah nilai enum.
SET	VARCHAR(255*2)	Seperti enum.

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
DATE	DATE	(YYYY-MM-DD) "1000-01-01" hingga "9999-12-31"
TIME	VARCHAR(10*4)	(hh:mm:ss) "-838:59:59" hingga "838:59:59"
DATETIME	TIMESTAMP	(YYYY-MM-DD hh: mm: ss) 1000-01-01 00:00:00" hingga "9999-12-31 23:59:59"
TIMESTAMP	TIMESTAMP	(YYYYMMDDhhmmss) 19700101000000 hingga 2037+
YEAR	VARCHAR(4*4)	(YYYY) 1900 hingga 2155
kolom SERIAL	Generasi ID / Atribut ini tidak diperlukan untuk gudang data OLAP karena kolom ini disalin. Kata kunci SERIAL tidak ditambahkan saat menerjemahkan.	SERIAL sebenarnya adalah entitas bernama SEQUENCE. Ini ada secara independen di sisa tabel Anda. kolom GENERATED BY DEFAULT setara dengan: nama CREATE SEQUENCE; tabel CREATE TABLE(kolom INTEGER NOT NULL DEFAULT nextval(nama));

Tipe Data MySQL	Tipe Data Amazon Redshift	Catatan
kolom BIGINT UNSIGNED NOT NULL AUTO_INCR EMENT UNIQUE	Generasi ID / Atribut ini tidak diperlukan untuk gudang data OLAP karena kolom ini disalin. Jadi kata kunci SERIAL tidak ditambahkan saat menerjema hkan.	SERIAL sebenarnya adalah entitas bernama SEQUENCE. Ini ada secara independen di sisa tabel Anda. kolom GENERATED BY DEFAULT setara dengan: nama CREATE SEQUENCE; tabel CREATE TABLE(kolom INTEGER NOT NULL DEFAULT nextval(nama));
ZEROFILL	Kata kunci ZEROFILL tidak ditambahkan saat menerjema hkan.	INT UNSIGNED ZEROFILL NOT NULL ZEROFILL mengisi nilai bidang yang ditampilkan dengan nol hingga lebar tampilan yang ditentukan dalam definisi kolom. Nilai yang lebih panjang dari lebar tampilan tidak terpotong. Perhatikan bahwa penggunaa n ZEROFILL juga menyiratkan UNSIGNED.

Salinan lengkap tabel Amazon RDS MySQL ke Amazon Redshift

Templat Salinan lengkap tabel MySQL Amazon RDS ke Amazon Redshift menyalin seluruh tabel MySQL Amazon RDS ke tabel Amazon Redshift dengan menyusun data dalam folder Amazon S3. Folder penyusunan Amazon S3 harus berada di wilayah yang sama dengan klaster Amazon Redshift. Tabel Amazon Redshift dibuat dengan skema yang sama seperti tabel MySQL Amazon RDS sumber

jika belum ada. Harap berikan penimpaan tipe data kolom MySQL Amazon RDS ke Amazon Redshift yang ingin Anda terapkan selama pembuatan tabel Amazon Redshift.

Templat menggunakan objek alur berikut:

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

Salinan tambahan tabel MySQL Amazon RDS ke Amazon Redshift

Templat Salinan tambahan tabel MySQL Amazon RDS ke Amazon Redshift menyalin data dari tabel MySQL Amazon RDS ke tabel Amazon Redshift dengan menyusun data dalam folder Amazon S3.

Folder penyusunan Amazon S3 harus berada di wilayah yang sama dengan klaster Amazon Redshift.

AWS Data Pipeline menggunakan skrip terjemahan untuk membuat tabel Amazon Redshift dengan skema yang sama dengan tabel MySQL Amazon RDS sumber jika belum ada. Anda harus memberikan penimpaan tipe data kolom MySQL Amazon RDS ke Amazon Redshift yang ingin Anda terapkan selama pembuatan tabel Amazon Redshift.

Templat ini menyalin perubahan yang dibuat ke tabel MySQL Amazon RDS di antara interval terjadwal, mulai dari waktu mulai terjadwal. Penghapusan fisik ke tabel MySQL Amazon RDS tidak disalin. Anda harus memberikan nama kolom yang menyimpan nilai waktu terakhir yang diubah.

Ketika Anda menggunakan templat default untuk membuat alur untuk salinan Amazon RDS tambahan, aktivitas dengan nama default `RDSToS3CopyActivity` dibuat. Anda dapat mengganti namanya.

Templat menggunakan objek alur berikut:

- CopyActivity
- RedshiftCopyActivity
- S3 DataNode

- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

Muat data dari Amazon S3 ke Amazon Redshift

Templat Muat data dari Amazon S3 ke Amazon Redshift menyalin data dari folder Amazon S3 ke dalam tabel Amazon Redshift. Anda dapat memuat data ke dalam tabel yang sudah ada atau menyediakan kueri SQL untuk membuat tabel.

Data disalin berdasarkan opsi `COPY` Amazon Redshift. Tabel Amazon Redshift harus memiliki skema yang sama dengan data di Amazon S3. Untuk pilihan `COPY`, lihat COPY di Panduan Developer Basis Data Amazon Redshift.

Templat menggunakan objek alur berikut:

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

Membuat pipeline Menggunakan template parametrized

Anda dapat menggunakan templat parametris untuk menyesuaikan definisi alur. Hal ini memungkinkan Anda untuk membuat definisi alur yang umum tetapi memberikan parameter yang berbeda saat Anda menambahkan definisi alur ke alur baru.

Daftar Isi

- Tambahkan MyVariables ke definisi pipeline
- Tentukan objek parameter
- Menentukan Nilai Parameter
- Mengirimkan definisi pipa

Tambahkan MyVariables ke definisi pipeline

Saat Anda membuat file definisi pipeline, tentukan variabel menggunakan sintaks berikut: # {my*Variable*}. Diperlukan agar variabel menggunakan prefiks `my`. Misalnya, file definisi pipeline berikut,`pipeline-definition.json`, mencakup variabel-variabel berikut:*myShellCmd*,*myS3InputLoc*, dan*myS3OutputLoc*.

> **Note**
>
> Definisi alur memiliki batas atas 50 parameter.

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

Tentukan objek parameter

Anda dapat membuat file terpisah dengan objek parameter yang mendefinisikan variabel dalam definisi alur Anda. Misalnya, file JSON berikut,`parameters.json`, berisi objek parameter untuk*myShellCmd*,*myS3InputLoc*, dan *myS3OutputLoc* variabel dari definisi pipeline contoh di atas.

```
{
  "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**
>
> Anda dapat menambahkan objek ini langsung ke file definisi alur alih-alih menggunakan file terpisah.

Tabel berikut menjelaskan atribut untuk objek parameter.

Atribut parameter

Atribut	Tipe	Deskripsi
`id`	String	Pengidentifikasi unik dari parameter. Untuk menutupi nilai saat diketik atau ditampilkan, tambahkan tanda bintang ('*') sebagai prefiks. Misalnya, `*myVariable` -. Perhatikan bahwa ini juga mengenkripsi nilai sebelum disimpan oleh AWS Data Pipeline.
deskripsi	String	Deskripsi parameter.

Atribut	Tipe	Deskripsi
jenis	String, Integer, Ganda, atau AWS::S3::ObjectKey	Jenis parameter yang menentukan rentang nilai input dan aturan validasi yang diizinkan. Default-nya adalah String.
opsional	Boolean	Menunjukkan apakah parameter adalah opsional atau diperlukan. Nilai default-nya `false`.
allowedValues	Daftar String	Menghitung semua nilai yang diizinkan untuk parameter.
default	String	Nilai default untuk parameter . Jika Anda menentukan nilai untuk parameter ini menggunakan nilai parameter, nilai default akan ditimpa.
isArray	Boolean	Menunjukkan apakah parameter adalah array.

Menentukan Nilai Parameter

Anda dapat membuat file terpisah untuk mendefinisikan variabel Anda menggunakan nilai parameter. Misalnya, file JSON berikut,`file://values.json`, berisi nilai untuk *`myS3OutputLoc`* variabel dari definisi pipeline contoh di atas.

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

Mengirimkan definisi pipa

Ketika Anda mengirimkan definisi alur Anda, Anda dapat menentukan parameter, objek parameter, dan nilai-nilai parameter. Misalnya, Anda dapat menggunakan put-pipeline-definition AWS CLI perintah sebagai berikut:

```
$ 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**
>
> Definisi alur memiliki batas atas 50 parameter. Ukuran file untuk `parameter-values-uri` memiliki batas atas 15 KB.

Melihat Alur Anda

Anda dapat melihat pipeline Anda menggunakan antarmuka baris perintah (CLI).

Untuk melihat saluran pipa Anda menggunakan AWS CLI

- Gunakan perintah list-pipelines berikut untuk melihat daftar alur Anda:

  ```
  aws datapipeline list-pipelines
  ```

Menafsirkan Kode Status Alur

Tingkat status yang ditampilkan di AWS Data Pipeline konsol dan CLI menunjukkan kondisi pipa dan komponennya. Status alur hanyalah gambaran umum dari alur; untuk melihat informasi lebih lanjut, lihat status masing-masing komponen alur.

Sebuah alur memiliki status `SCHEDULED` jika sudah siap (definisi alur lulus validasi), sedang melakukan pekerjaan, atau selesai melakukan pekerjaan. Sebuah alur memiliki status `PENDING` jika tidak diaktifkan atau tidak dapat melakukan pekerjaan (misalnya, definisi alur gagal validasi.)

Sebuah alur dianggap tidak aktif jika statusnya `PENDING`, `INACTIVE`, atau `FINISHED`. Alur yang tidak aktif dikenakan biaya (untuk informasi lebih lanjut, lihat Harga).

Kode Status

`ACTIVATING`

Komponen atau sumber daya sedang dimulai, seperti EC2 instance.

`CANCELED`

Komponen dibatalkan oleh pengguna atau AWS Data Pipeline sebelum dapat dijalankan. Hal ini dapat terjadi secara otomatis ketika terjadi kegagalan pada komponen atau sumber daya yang berbeda yang bergantung pada komponen ini.

`CASCADE_FAILED`

Komponen atau sumber daya dibatalkan sebagai akibat dari kegagalan kaskade dari salah satu dependensinya, tetapi komponen tersebut mungkin bukan sumber asli kegagalan.

`DEACTIVATING`

Alur sedang dinonaktifkan.

`FAILED`

Komponen atau sumber daya mengalami kesalahan dan berhenti bekerja. Ketika komponen atau sumber daya gagal, itu dapat menyebabkan pembatalan dan kegagalan untuk mengalir ke komponen lain yang bergantung padanya.

`FINISHED`

Komponen menyelesaikan pekerjaan yang ditugaskan.

`INACTIVE`

Alur dinonaktifkan.

`PAUSED`

Komponen dijeda dan saat ini tidak menjalankan tugasnya.

`PENDING`

Alur siap untuk diaktifkan untuk pertama kalinya.

`RUNNING`

Sumber daya sedang berjalan dan siap menerima pekerjaan.

`SCHEDULED`

Sumber daya dijadwalkan untuk berjalan.

`SHUTTING_DOWN`

Sumber daya dimatikan setelah berhasil menyelesaikan pekerjaannya.

`SKIPPED`

Komponen melewatkan interval eksekusi setelah alur diaktifkan menggunakan stempel waktu yang lebih lambat dari jadwal saat ini.

`TIMEDOUT`

Sumber daya melebihi `terminateAfter` ambang batas dan dihentikan. AWS Data Pipeline Setelah sumber daya mencapai status ini, AWS Data Pipeline mengabaikan nilai `actionOnResourceFailure`, `retryDelay`, dan `retryTimeout` untuk sumber daya tersebut. Status ini hanya berlaku untuk sumber daya.

`VALIDATING`

Definisi pipeline sedang divalidasi oleh AWS Data Pipeline.

`WAITING_FOR_RUNNER`

Komponen sedang menunggu klien pekerjanya untuk mengambil item pekerjaan. Hubungan klien komponen dan pekerja dikendalikan oleh bidang `runsOn` atau `workerGroup` yang ditentukan oleh komponen tersebut.

`WAITING_ON_DEPENDENCIES`

Komponen sedang memverifikasi bahwa prakondisi default dan yang dikonfigurasi pengguna terpenuhi sebelum melakukan pekerjaannya.

Menafsirkan Alur dan Status Kondisi Komponen

Setiap alur dan komponen dalam alur tersebut mengembalikan status kondisi `HEALTHY`, `ERROR`, `"-"`, `No Completed Executions`, atau `No Health Information Available`. Alur hanya memiliki status kondisi setelah komponen alur menyelesaikan eksekusi pertamanya atau jika prasyarat komponen gagal. Status kondisi untuk komponen digabungkan ke dalam status kondisi alur dalam status kesalahan itu terlihat pertama kali saat Anda melihat detail eksekusi alur Anda.

Status Kondisi Alur

`HEALTHY`

Status kondisi gabungan dari semua komponen adalah `HEALTHY`. Ini berarti setidaknya satu komponen harus berhasil diselesaikan. Anda dapat mengeklik status `HEALTHY` untuk melihat instans komponen alur yang paling terbaru yang berhasil diselesaikan di halaman Detail Eksekusi.

`ERROR`

Setidaknya satu komponen dalam alur memiliki status kondisi `ERROR`. Anda dapat mengeklik status `ERROR` untuk melihat instans komponen alur yang paling terbaru yang gagal di halaman Detail Eksekusi.

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

Tidak ada status kondisi yang dilaporkan untuk alur ini.

> **Note**
>
> Meskipun komponen segera memperbarui status kondisinya, mungkin diperlukan waktu hingga lima menit untuk memperbarui status kondisi alur.

Status Kondisi Komponen

`HEALTHY`

Komponen (`Activity` atau `DataNode`) memiliki status kondisi `HEALTHY` jika telah menyelesaikan eksekusi yang sukses di mana ia ditandai dengan status `FINISHED` atau `MARK_FINISHED`. Anda dapat mengeklik nama komponen atau status `HEALTHY` untuk melihat instans komponen alur yang paling terbaru yang berhasil diselesaikan di halaman Detail Eksekusi.

`ERROR`

Terjadi kesalahan pada tingkat komponen atau salah satu prasyaratnya gagal. Status `FAILED`, `TIMEOUT`, atau `CANCELED` memicu kesalahan ini. Anda dapat mengeklik nama komponen atau status `ERROR` untuk melihat instans komponen alur yang paling terbaru yang gagal di halaman Detail Eksekusi.

`No Completed Executions` atau `No Health Information Available`

Tidak ada status kondisi yang dilaporkan untuk komponen ini.

Melihat Definisi Alur Anda

Gunakan antarmuka baris perintah (CLI) untuk melihat definisi pipeline Anda. CLI mencetak file definisi pipeline, dalam format JSON. Untuk informasi tentang sintaks dan penggunaan file definisi alur, lihat Sintaks file definisi pipa.

Saat menggunakan CLI, ada baiknya untuk mengambil definisi pipeline sebelum Anda mengirimkan modifikasi, karena ada kemungkinan pengguna atau proses lain mengubah definisi pipeline setelah Anda terakhir kali bekerja dengannya. Dengan mengunduh salinan definisi saat ini dan menggunakannya sebagai dasar untuk modifikasi Anda, Anda dapat yakin bahwa Anda bekerja dengan definisi alur paling terbaru. Ini juga merupakan ide yang baik untuk mengambil definisi alur lagi setelah Anda memodifikasinya, sehingga Anda dapat memastikan bahwa pembaruan berhasil.

Saat menggunakan CLI, Anda bisa mendapatkan dua versi pipa yang berbeda. Versi `active` adalah alur yang saat ini berjalan. Versi `latest` adalah salinan yang dibuat saat Anda mengedit alur yang saat ini berjalan. Ketika Anda mengunggah alur yang diedit, itu menjadi versi `active` dan versi `active` sebelumnya tidak lagi tersedia.

Untuk mendapatkan definisi pipeline menggunakan AWS CLI

Untuk mendapatkan definisi pipeline lengkap, gunakan get-pipeline-definitionperintah. Definisi alur dicetak ke output standar (stdout).

Contoh berikut mendapatkan definisi alur untuk alur yang ditentukan.

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

Untuk mengambil versi alur tertentu, gunakan opsi `--version`. Contoh berikut mengambil versi `active` dari alur yang ditentukan.

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

Melihat Detail Instans Alur

Anda dapat memantau kemajuan alur Anda. Untuk informasi lebih lanjut tentang status instans, lihat Menafsirkan Detail Status Alur. Untuk informasi selengkapnya tentang pemecahan masalah yang gagal atau tidak lengkapnya proses instans dari alur Anda, lihat Menyelesaikan Masalah Umum.

Untuk memantau kemajuan pipa menggunakan AWS CLI

Untuk mengambil detail instans alur, seperti riwayat waktu alur telah berjalan, gunakan perintah list-runs. Perintah ini memungkinkan Anda untuk memfilter daftar penjalanan yang dikembalikan berdasarkan status mereka saat ini atau rentang tanggal di mana mereka diluncurkan. Memfilter hasil berguna karena, bergantung pada usia dan penjadwalan alur, riwayat penjalanan bisa besar.

Contoh berikut mengambil informasi untuk semua penjalanan.

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

Contoh berikut mengambil informasi untuk semua penjalanan yang telah selesai.

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

Contoh berikut mengambil informasi untuk semua penjalanan yang diluncurkan dalam kerangka waktu yang ditentukan.

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

Melihat Log Alur

Pencatatan log tingkat alur didukung pada pembuatan alur dengan menentukan lokasi Amazon S3 di konsol atau dengan `pipelineLogUri` dalam objek default dalam SDK/CLI. Struktur direktori untuk setiap alur di dalam URI tersebut adalah seperti berikut:

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

Untuk alur, `df-00123456ABC7DEF8HIJK`, struktur direktori terlihat seperti:

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

Untuk `ShellCommandActivity`, log untuk `stderr` dan `stdout` terkait dengan aktivitas ini disimpan di direktori untuk setiap upaya.

Untuk sumber daya seperti, `EmrCluster`, di mana `emrLogUri` diatur, nilai tersebut diutamakan. Jika tidak, sumber daya (termasuk TaskRunner log untuk sumber daya tersebut) mengikuti struktur pencatatan pipa di atas.

Untuk melihat log untuk menjalankan pipeline tertentu:

1. Ambil `ObjectId` dengan memanggil `query-objects` untuk mendapatkan ID objek yang tepat. Contoh:

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

   `query-objects`adalah CLI paginasi dan dapat mengembalikan token pagination jika ada lebih banyak eksekusi untuk yang diberikan. `pipeline-id` Anda dapat menggunakan token untuk melalui semua upaya sampai Anda menemukan objek yang diharapkan. Misalnya, yang dikembalikan ObjectId akan terlihat seperti:`@TableBackupActivity_2023-05-020T18:05:18_Attempt=1`.
2. Menggunakan ObjectId, ambil lokasi log menggunakan:

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

Pesan galat dari aktivitas yang gagal

Untuk mendapatkan pesan kesalahan, pertama-tama ObjectId gunakan`query-objects`.

Setelah mengambil yang gagal ObjectId, gunakan `describe-objects` CLI untuk mendapatkan pesan kesalahan yang sebenarnya.

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

Batalkan atau jalankan kembali atau tandai sebagai objek yang sudah selesai

Gunakan `set-status` CLI untuk membatalkan objek yang sedang berjalan, atau menjalankan kembali objek yang gagal atau menandai objek yang sedang berjalan sebagai Selesai.

Pertama, dapatkan ID objek menggunakan `query-objects` CLI. Contoh:

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

Gunakan `set-status` CLI untuk mengubah status objek yang diinginkan. Contoh:

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

Mengedit Alur Anda

Untuk mengubah beberapa aspek dari salah satu alur Anda, Anda dapat memperbarui definisi alurnya. Setelah Anda mengubah alur yang sedang berjalan, Anda harus mengaktifkan kembali alur agar perubahan Anda diterapkan. Selain itu, Anda dapat menjalankan kembali satu atau beberapa komponen alur.

Daftar Isi

- Batasan
- Mengedit Pipeline Menggunakan AWS CLI

Batasan

Saat alur dalam status `PENDING` dan tidak diaktifkan, Anda tidak dapat membuat perubahan apa pun pada alur itu. Setelah Anda mengaktifkan alur Anda dapat mengedit alur dengan batasan berikut. Perubahan yang Anda buat berlaku untuk menjalankan objek alur baru setelah Anda menyimpannya dan kemudian mengaktifkan alur lagi.

- Anda tidak dapat menghapus objek
- Anda tidak dapat mengubah periode jadwal objek yang sudah ada
- Anda tidak dapat menambahkan, menghapus, atau memodifikasi bidang referensi di objek yang sudah ada
- Anda tidak dapat mereferensikan objek yang sudah ada di bidang output dari objek baru
- Anda tidak dapat mengubah tanggal mulai terjadwal suatu objek (sebagai gantinya, aktifkan alur dengan tanggal dan waktu tertentu)

Mengedit Pipeline Menggunakan AWS CLI

Anda dapat mengedit alur menggunakan alat baris perintah.

Pertama, unduh salinan definisi pipeline saat ini menggunakan get-pipeline-definitionperintah. Dengan melakukan ini, Anda dapat yakin bahwa Anda sedang memodifikasi definisi alur terbaru. Contoh berikut menggunakan mencetak definisi alur ke output standar (stdout).

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

Simpan definisi alur ke file dan edit sesuai kebutuhan. Perbarui definisi pipeline Anda menggunakan put-pipeline-definitionperintah. Contoh berikut mengunggah file definisi alur yang diperbarui.

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

Anda dapat mengambil kembali definisi alur menggunakan perintah `get-pipeline-definition` untuk memastikan bahwa pembaruan berhasil. Untuk mengaktifkan alur, gunakan perintah activate-pipeline berikut:

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

Jika Anda mau, Anda dapat mengaktifkan alur dari tanggal dan waktu tertentu, menggunakan opsi `--start-timestamp` sebagai berikut:

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

Untuk menjalankan ulang satu atau beberapa komponen alur, gunakan perintah set-status.

Mengkloning Alur Anda

Pengkloningan membuat salinan alur dan memungkinkan Anda menentukan nama untuk alur baru. Anda dapat mengkloning alur yang dalam status apa pun, meskipun memiliki kesalahan; namun, alur baru tetap dalam status `PENDING` sampai Anda mengaktifkannya secara manual. Untuk alur baru, operasi kloning menggunakan versi terbaru dari definisi alur asli daripada versi aktif. Dalam operasi kloning, jadwal lengkap dari alur asli tidak disalin ke alur baru, hanya pengaturan periode.

Untuk mengkloning pipeline menggunakan AWS CLI:

1. Buat pipeline baru dengan nama baru dan ID unik. Perhatikan ID pipa yang dikembalikan.
2. Gunakan `get-pipeline-definition` CLI untuk mendapatkan definisi pipeline dari pipeline yang ada untuk dikloning dan tulis ke file sementara. Perhatikan jalur absolut dari file tersebut.
3. Gunakan `put-pipeline-definition` CLI untuk menyalin definisi pipeline dari pipeline yang ada ke pipeline baru.
4. Gunakan `get-pipeline-definition` CLI untuk mendapatkan definisi pipeline baru untuk memverifikasi definisi 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
```

Menandai Alur Anda

Tag adalah pasangan nilai kunci peka huruf besar/kecil yang terdiri dari kunci dan nilai opsional, keduanya ditentukan oleh pengguna. Anda dapat menerapkan hingga sepuluh tag ke setiap alur. Kunci tag harus unik untuk setiap alur. Jika Anda menambahkan tag dengan kunci yang sudah terkait dengan alur, maka nilai tag tersebut akan diperbarui.

Menerapkan tag ke pipeline juga menyebarkan tag ke sumber daya dasarnya (misalnya, klaster EMR Amazon dan instans EC2 Amazon). Namun, itu tidak menerapkan tag ini ke sumber daya yang dalam status `FINISHED` atau sebaliknya diakhiri. Anda dapat menggunakan CLI untuk menerapkan tag ke sumber daya ini, jika diperlukan.

Setelah selesai dengan tag, Anda dapat menghapusnya dari alur Anda.

Untuk memberi tag alur Anda menggunakan AWS CLI

Untuk menambahkan tag ke alur baru, tambahkan opsi `--tags` ke perintah create-pipeline. Misalnya, opsi berikut membuat alur dengan dua tag, tag `environment` dengan nilai `production`, dan tag `owner` dengan nilai `sales`.

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

Untuk menambahkan tag ke alur yang ada, gunakan perintah add-tags sebagai berikut:

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

Untuk menghapus tag dari alur yang ada, gunakan perintah remove-tags sebagai berikut:

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

Menonaktifkan Alur Anda

Menonaktifkan alur yang sedang berjalan akan menjeda eksekusi alur. Untuk melanjutkan eksekusi alur, Anda dapat mengaktifkan alur. Ini memungkinkan Anda untuk membuat perubahan. Misalnya, jika Anda menulis data ke basis data yang dijadwalkan untuk menjalani pemeliharaan, Anda dapat menonaktifkan alur, menunggu hingga pemeliharaan selesai, lalu mengaktifkan alur.

Saat Anda menonaktifkan alur, Anda dapat menentukan apa yang terjadi pada aktivitas yang sedang berjalan. Secara default, aktivitas ini segera dibatalkan. Atau, Anda dapat meminta AWS Data Pipeline menunggu hingga aktivitas selesai sebelum menonaktifkan alur.

Saat mengaktifkan alur yang dinonaktifkan, Anda dapat menentukan kapan alur dilanjutkan. Menggunakan AWS CLI atau API, pipeline dilanjutkan dari eksekusi terakhir yang diselesaikan secara default, atau Anda dapat menentukan tanggal dan waktu untuk melanjutkan pipeline.

Menonaktifkan Alur Anda Menggunakan AWS CLI

Gunakan perintah deactivate-pipeline berikut untuk menonaktifkan alur:

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

Untuk menonaktifkan alur hanya setelah semua aktivitas yang berjalan selesai, tambahkan opsi `--no-cancel-active`, sebagai berikut:

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

Saat Anda siap, Anda dapat melanjutkan eksekusi alur di mana ia tinggalkan menggunakan perintah activate-pipeline berikut:

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

Untuk memulai alur dari tanggal dan waktu tertentu, tambahkan opsi `--start-timestamp`, sebagai berikut:

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

Menghapus Alur Anda

Ketika Anda tidak lagi memerlukan alur, seperti alur yang dibuat selama pengujian aplikasi, Anda harus menghapusnya untuk menyingkirkannya dari penggunaan aktif. Menghapus alur akan membuatnya dalam status menghapus. Ketika alur dalam status terhapus, definisi alur dan riwayat penjalanannya akan hilang. Oleh karena itu, Anda tidak dapat lagi melakukan operasi pada alur, termasuk mendeskripsikannya.

> **Important**
>
> Anda tidak dapat memulihkan alur setelah Anda menghapusnya, jadi pastikan Anda tidak memerlukan alur di masa mendatang sebelum Anda menghapusnya.

Untuk menghapus pipa menggunakan AWS CLI

Untuk menghapus alur, gunakan perintah delete-pipeline. Perintah berikut menghapus alur yang ditentukan.

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

Penahapan Data dan Tabel dengan Aktivitas Alur

AWS Data Pipeline dapat melakukan input dan output data dalam pipeline Anda untuk mempermudah penggunaan aktivitas tertentu, seperti `ShellCommandActivity` dan`HiveActivity`.

Penahapan data memungkinkan Anda untuk menyalin data dari simpul data input ke sumber daya yang menjalankan aktivitas, dan, sama halnya, dari sumber daya ke simpul data output.

Data bertahap di Amazon EMR atau sumber daya EC2 Amazon tersedia dengan menggunakan variabel khusus dalam perintah shell aktivitas atau skrip Hive.

Penahapan tabel mirip dengan penahapan data, kecuali data yang ditahapkan berbentuk tabel basis data, khususnya.

AWS Data Pipeline mendukung skenario pementasan berikut:

- Penahapan data dengan `ShellCommandActivity`
- Penahapan tabel dengan Hive dan simpul data yang didukung penahapan
- Penahapan tabel dengan Hive dan simpul data yang tidak didukung penahapan

> **Note**
>
> Penahapan hanya berfungsi ketika bidang `stage` diatur ke `true` pada suatu aktivitas, seperti `ShellCommandActivity`. Untuk informasi selengkapnya, lihat ShellCommandActivity.

Selain itu, simpul data dan aktivitas dapat berhubungan dalam empat cara:

Penahapan data secara lokal pada sumber daya

Data input secara otomatis disalin ke sistem file lokal sumber daya. Data output secara otomatis disalin dari sistem file lokal sumber daya ke simpul data output. Misalnya, ketika Anda mengonfigurasi input dan output `ShellCommandActivity` dengan penahapan = true, data input tersedia sebagai INPUTx_STAGING_DIR dan data output tersedia sebagai OUTPUTx_STAGING_DIR, di mana x adalah jumlah input atau output.

Penahapan definisi input dan output untuk suatu aktivitas

Format data input (nama kolom dan nama tabel) secara otomatis disalin ke sumber daya aktivitas. Misalnya, ketika Anda mengkonfigurasi `HiveActivity` dengan penahapan = true. Format data yang ditentukan pada input `S3DataNode` digunakan untuk menentukan definisi tabel dari tabel Hive.

Penahapan tidak diaktifkan

Objek input dan output serta bidangnya tersedia untuk aktivitas, tetapi datanya sendiri tidak tersedia. Misalnya, `EmrActivity` secara default atau saat Anda mengonfigurasi aktivitas lain dengan penahapan = false. Dalam konfigurasi ini, bidang data tersedia untuk aktivitas untuk membuat referensi ke mereka menggunakan sintaks AWS Data Pipeline ekspresi, dan ini hanya terjadi ketika ketergantungan terpenuhi. Ini berfungsi sebagai pemeriksaan dependensi saja. Kode dalam aktivitas bertanggung jawab untuk menyalin data dari input ke sumber daya yang menjalankan aktivitas.

Hubungan dependensi antar objek

Ada hubungan tergantung-pada antara dua objek, yang menghasilkan situasi yang sama ketika penahapan tidak diaktifkan. Hal ini menyebabkan simpul data atau aktivitas bertindak sebagai prasyarat untuk eksekusi aktivitas lain.

Pementasan Data dengan ShellCommandActivity

Pertimbangkan skenario menggunakan `S3DataNode` objek `ShellCommandActivity` with sebagai input dan output data. AWS Data Pipeline secara otomatis tahap node data untuk membuat mereka dapat diakses oleh perintah shell seolah-olah mereka adalah folder file lokal menggunakan variabel lingkungan `${INPUT1_STAGING_DIR}` dan `${OUTPUT1_STAGING_DIR}` seperti yang ditunjukkan dalam contoh berikut. Bagian numerik dari variabel bernama `INPUT1_STAGING_DIR` dan kenaikan `OUTPUT1_STAGING_DIR` tergantung pada jumlah simpul data referensi aktivitas Anda.

> **Note**
>
> Skenario ini hanya berfungsi seperti yang dijelaskan jika input dan output data Anda adalah objek `S3DataNode`. Selain itu, penahapan data output hanya diperbolehkan jika `directoryPath` diatur pada objek `S3DataNode` output.

```
{
```

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

Penahapan Tabel dengan Hive dan Simpul Data yang Didukung Penahapan

Pertimbangkan skenario menggunakan `S3DataNode` objek `HiveActivity` with sebagai input dan output data. AWS Data Pipeline secara otomatis tahap node data untuk membuat mereka dapat diakses oleh skrip Hive seolah-olah mereka adalah tabel Hive menggunakan variabel `${input1}` dan `${output1}` seperti yang ditunjukkan dalam contoh berikut untuk. `HiveActivity` Bagian numerik dari variabel bernama `input` dan kenaikan `output` tergantung pada jumlah simpul data referensi aktivitas Anda.

> **Note**
>
> Skenario ini hanya berfungsi seperti yang dijelaskan jika input dan output data Anda adalah objek `S3DataNode` atau `MySqlDataNode`. Penahapan tabel tidak didukung untuk `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"
  }
},
```

```
...
```

Penahapan Tabel dengan Hive dan Simpul Data yang Tidak Didukung Penahapan

Pertimbangkan skenario menggunakan `HiveActivity` dengan `DynamoDBDataNode` sebagai input data dan objek `S3DataNode` sebagai output. Tidak ada penahapan data yang tersedia untuk `DynamoDBDataNode`, oleh karena itu Anda harus terlebih dahulu secara manual membuat tabel dalam skrip Hive Anda, menggunakan nama variabel `#{input.tableName}` untuk merujuk ke tabel DynamoDB. Nomenklatur serupa berlaku jika tabel DynamoDB adalah outputnya, kecuali Anda menggunakan variabel `#{output.tableName}`. Penahapan tersedia untuk objek `S3DataNode` output dalam contoh ini, oleh karena itu Anda dapat merujuk ke simpul data output sebagai `${output1}`.

> **Note**
>
> Dalam contoh ini, variabel nama tabel memiliki awalan karakter # (hash) karena AWS Data Pipeline menggunakan ekspresi untuk mengakses or. `tableName` `directoryPath` Untuk informasi lebih lanjut tentang cara kerja evaluasi ekspresi AWS Data Pipeline, lihatEvaluasi Ekspresi.

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

Menggunakan Alur dengan Sumber Daya di Beberapa Wilayah

Secara default, `EmrCluster` sumber daya `Ec2Resource` dan berjalan di wilayah yang sama dengan AWS Data Pipeline, namun AWS Data Pipeline mendukung kemampuan untuk mengatur aliran data di beberapa wilayah, seperti menjalankan sumber daya di satu wilayah yang mengkonsolidasikan data input dari wilayah lain. Dengan mengizinkan sumber daya untuk menjalankan wilayah tertentu, Anda juga memiliki fleksibilitas untuk mengkolokasi sumber daya Anda bersama dengan set data dependennya dan memaksimalkan performa dengan mengurangi latensi dan menghindari biaya transfer data lintas wilayah. Anda dapat mengonfigurasi sumber daya untuk dijalankan di wilayah yang berbeda AWS Data Pipeline dengan menggunakan `region` bidang pada `Ec2Resource` dan`EmrCluster`.

Contoh file JSON alur berikut menunjukkan cara menjalankan sumber daya `EmrCluster` di wilayah Europe (Ireland), dengan asumsi bahwa sejumlah besar data untuk klaster yang akan dikerjakan ada di wilayah yang sama. Dalam contoh ini, satu-satunya perbedaan dari alur khas adalah bahwa `EmrCluster` memiliki nilai bidang `region` yang diatur ke `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"
    }
  ]
}
```

Tabel berikut mencantumkan wilayah yang dapat Anda pilih dan kode wilayah terkait untuk digunakan di bidang `region`.

> **Note**
>
> Daftar berikut mencakup wilayah yang AWS Data Pipeline dapat mengatur alur kerja dan meluncurkan sumber daya Amazon EMR atau Amazon. EC2 AWS Data Pipeline mungkin

tidak didukung di wilayah ini. Untuk informasi tentang wilayah yang AWS Data Pipeline didukung, lihat Wilayah dan Titik Akhir AWS.

Nama wilayah	Kode Wilayah
US East (Northern Virginia)	us-east-1
US East (Ohio)	us-east-2
AS Barat (California Utara)	us-west-1
US West (Oregon)	us-west-2
Kanada (Pusat)	ca-central-1
Eropa (Irlandia)	eu-west-1
Europe (London)	eu-west-2
Eropa (Frankfurt)	eu-central-1
Asia Pasifik (Singapura)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pasifik (Mumbai)	ap-south-1
Asia Pasifik (Tokyo)	ap-northeast-1
Asia Pasifik (Seoul)	ap-northeast-2
Amerika Selatan (Sao Paulo)	sa-east-1

Kegagalan dan tayangan ulang yang berulang

AWS Data Pipeline memungkinkan Anda mengonfigurasi cara objek pipeline berperilaku ketika ketergantungan gagal atau dibatalkan oleh pengguna. Anda dapat memastikan bahwa kegagalan kaskade ke objek alur lain (konsumen), untuk mencegah menunggu tanpa batas. Semua aktivitas,

simpul data, dan prasyarat memiliki bidang bernama `failureAndRerunMode` dengan nilai default `none`. Untuk mengaktifkan cascading kegagalan, atur bidang `failureAndRerunMode` ke `cascade`.

Saat bidang ini diaktifkan, kegagalan kaskade terjadi jika objek alur diblokir dalam status `WAITING_ON_DEPENDENCIES` dan dependensi apa pun telah gagal tanpa perintah tertunda. Selama kegagalan kaskade, peristiwa berikut terjadi:

- Ketika sebuah objek gagal, konsumennya diatur ke `CASCADE_FAILED` dan objek asli serta prasyarat konsumennya diatur ke `CANCELED`.
- Setiap objek yang sudah `FINISHED`, `FAILED`, atau `CANCELED` diabaikan.

Kegagalan kaskade tidak beroperasi pada dependensi objek gagal (hulu), kecuali untuk prasyarat yang terkait dengan objek gagal asli. Objek alur yang terpengaruh oleh kegagalan kaskade dapat memicu percobaan ulang atau tindakan pasca apa pun, seperti `onFail`.

Efek detail dari cascading kegagalan bergantung pada jenis objek.

Aktivitas

Aktivitas berubah menjadi `CASCADE_FAILED` jika salah satu dependensinya gagal, dan selanjutnya memicu kegagalan kaskade di konsumen aktivitas. Jika sumber daya gagal yang mana aktivitas tergantung padanya, aktivitas tersebut `CANCELED` dan semua konsumennya berubah menjadi `CASCADE_FAILED`.

Node dan prasyarat data

Jika simpul data dikonfigurasi sebagai output dari aktivitas yang gagal, simpul data berubah ke status `CASCADE_FAILED`. Kegagalan simpul data menyebar ke setiap prasyarat terkait, yang berubah ke status `CANCELED`.

Sumber daya

Jika objek yang bergantung pada sumber daya berada dalam status `FAILED` dan sumber daya itu sendiri dalam status `WAITING_ON_DEPENDENCIES`, maka sumber daya berubah ke status `FINISHED`.

Menjalankan kembali objek yang gagal bertingkat

Secara default, menjalankan ulang aktivitas atau simpul data apa pun hanya akan menjalankan ulang sumber daya terkait. Namun, mengatur bidang `failureAndRerunMode` ke `cascade` pada objek alur memungkinkan perintah jalankan ulang pada objek target untuk disebarkan ke semua konsumen, dalam kondisi berikut:

- Konsumen objek target berada dalam status `CASCADE_FAILED`.
- Dependensi objek target tidak memiliki perintah jalankan ulang yang tertunda.
- Dependensi objek target tidak dalam status `FAILED`, `CASCADE_FAILED`, atau `CANCELED`.

Jika Anda mencoba menjalankan ulang objek `CASCADE_FAILED` dan salah satu dependensinya adalah `FAILED`, `CASCADE_FAILED`, atau `CANCELED`, penjalanan ulang akan gagal dan mengembalikan objek ke status `CASCADE_FAILED`. Agar berhasil menjalankan ulang objek yang gagal, Anda harus melacak kegagalan ke rantai dependensi untuk menemukan sumber asli kegagalan dan menjalankan ulang objek tersebut. Ketika Anda mengeluarkan perintah jalankan ulang pada sumber daya, Anda juga mencoba menjalankan ulang objek apa pun yang bergantung padanya.

Kegagalan bertingkat dan penimbunan ulang

Jika Anda mengaktifkan kegagalan kaskade dan memiliki pipeline yang membuat banyak pengisian ulang, kesalahan runtime pipeline dapat menyebabkan sumber daya dibuat dan dihapus secara berurutan tanpa melakukan pekerjaan yang berguna. AWS Data Pipeline mencoba mengingatkan Anda tentang situasi ini dengan pesan peringatan berikut saat Anda menyimpan 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.` Ini terjadi karena kegagalan kaskade dapat dengan cepat mengatur aktivitas hilir dan mematikan kluster EMR `CASCADE_FAILED` dan EC2 sumber daya yang tidak lagi diperlukan. Kami merekomendasikan agar Anda menguji alur dengan rentang waktu yang singkat untuk membatasi efek dari situasi ini.

Sintaks file definisi pipa

Instruksi di bagian ini adalah untuk bekerja secara manual dengan file definisi pipa menggunakan antarmuka baris AWS Data Pipeline perintah (CLI). Ini adalah alternatif untuk merancang pipa secara interaktif menggunakan AWS Data Pipeline konsol.

Anda dapat membuat file definisi pipeline secara manual menggunakan editor teks apa pun yang mendukung penyimpanan file menggunakan format file UTF-8, dan mengirimkan file menggunakan antarmuka baris AWS Data Pipeline perintah.

AWS Data Pipeline juga mendukung berbagai ekspresi dan fungsi kompleks dalam definisi pipeline. Untuk informasi selengkapnya, lihat Ekspresi dan Fungsi Alur.

Struktur File

Langkah pertama dalam pembuatan alur adalah membuat objek definisi alur dalam file definisi alur. Contoh berikut mengilustrasikan struktur umum file definisi alur. File ini mendefinisikan dua objek, yang dibatasi oleh '{' dan '}', dan dipisahkan dengan koma.

Dalam contoh berikut, objek pertama mendefinisikan dua pasangan nama-nilai, yang dikenal sebagai bidang. Objek kedua mendefinisikan tiga bidang.

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

Saat membuat file definisi alur, Anda harus memilih tipe objek alur yang Anda butuhkan, menambahkannya ke file definisi alur, lalu menambahkan bidang yang sesuai. Untuk informasi selengkapnya tentang objek alur, lihat Referensi Objek Alur.

Misalnya, Anda bisa membuat objek definisi alur untuk simpul data input dan yang lain untuk simpul data output. Kemudian buat objek definisi alur lain untuk suatu aktivitas, seperti memproses data input menggunakan Amazon EMR.

Bidang Alur

Setelah Anda mengetahui tipe objek mana yang akan disertakan dalam file definisi alur, Anda menambahkan bidang ke definisi setiap objek alur. Nama bidang diapit dalam tanda kutip, dan dipisahkan dari nilai bidang dengan spasi, titik dua, dan spasi, seperti yang diperlihatkan dalam contoh berikut.

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

Nilai bidang dapat berupa string teks, referensi ke objek lain, pemanggilan fungsi, ekspresi, atau daftar berurutan dari tipe sebelumnya. Untuk informasi selengkapnya tentang tipe data yang bisa digunakan untuk nilai bidang, lihat Tipe Data Sederhana. Untuk informasi selengkapnya tentang fungsi yang dapat Anda gunakan untuk mengevaluasi nilai bidang, lihat Evaluasi Ekspresi.

Bidang dibatasi hingga 2048 karakter. Objek dapat berukuran 20 KB, yang berarti Anda tidak dapat menambahkan banyak bidang besar ke objek.

Setiap objek alur harus berisi bidang berikut: `id` dan `type`, seperti yang ditunjukkan dalam contoh berikut. Bidang lain mungkin juga diperlukan berdasarkan tipe objek. Pilih nilai untuk `id` yang berarti bagi Anda, dan unik dalam definisi alur. Nilai untuk `type` menentukan tipe objek. Tentukan salah satu tipe objek definisi alur yang didukung, yang tercantum dalam topik Referensi Objek Alur.

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

Untuk informasi selengkapnya tentang bidang wajib dan opsional untuk setiap objek, lihat dokumentasi untuk objek tersebut.

Untuk menyertakan bidang dari satu objek ke objek lain, gunakan bidang `parent` dengan referensi ke objek. Misalnya, objek "B" mencakup bidangnya, "B1" dan "B2", ditambah bidang dari objek "A", "A1" dan "A2".

```
{
  "id" : "A",
  "A1" : "value",
  "A2" : "value"
},
```

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

Anda dapat menentukan bidang umum dalam objek dengan ID "Default". Bidang ini secara otomatis disertakan dalam setiap objek dalam file definisi alur yang tidak secara eksplisit mengatur bidang `parent`-nya untuk mereferensikan objek yang berbeda.

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

Bidang yang ditentukan pengguna

Anda dapat membuat bidang yang ditentukan pengguna atau bidang khusus pada komponen alur Anda dan merujuknya dengan ekspresi. Contoh berikut menunjukkan bidang kustom bernama `myCustomField` dan `my_customFieldReference` ditambahkan ke DataNode objek S3:

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

Bidang yang ditentukan pengguna harus memiliki nama yang diawali dengan kata "saya" dalam huruf kecil semua, diikuti dengan huruf kapital atau karakter garis bawah. Selain itu, bidang yang ditentukan pengguna dapat berupa nilai string seperti contoh `myCustomField` sebelumnya, atau mereferensikan ke komponen alur lain seperti contoh `my_customFieldReference` sebelumnya.

> **Note**
>
> Pada bidang yang ditentukan pengguna, AWS Data Pipeline hanya memeriksa referensi yang valid ke komponen pipeline lainnya, bukan nilai string bidang kustom apa pun yang Anda tambahkan.

Bekerja dengan API

> **Note**
>
> Jika Anda tidak menulis program yang berinteraksi AWS Data Pipeline, Anda tidak perlu menginstal AWS apa pun SDKs. Anda dapat membuat dan menjalankan alur menggunakan konsol atau antarmuka baris perintah. Untuk informasi selengkapnya, lihat Menyiapkan untuk AWS Data Pipeline

Cara termudah untuk menulis aplikasi yang berinteraksi dengan AWS Data Pipeline atau mengimplementasikan Task Runner khusus adalah dengan menggunakan salah satu AWS SDKs. AWS SDKs menyediakan fungsionalitas yang menyederhanakan panggilan layanan web APIs dari lingkungan pemrograman pilihan Anda. Untuk informasi selengkapnya, lihat Pasang AWS SDK.

Pasang AWS SDK

AWS SDKs Menyediakan fungsi yang membungkus API dan menangani banyak detail koneksi, seperti menghitung tanda tangan, menangani percobaan ulang permintaan, dan penanganan kesalahan. Ini SDKs juga berisi kode contoh, tutorial, dan sumber daya lainnya untuk membantu Anda mulai menulis aplikasi yang memanggil AWS. Memanggil fungsi pembungkus dalam SDK dapat sangat menyederhanakan proses penulisan aplikasi. AWS Untuk informasi lebih lanjut tentang cara mengunduh dan menggunakan AWS SDKs, buka Kode Sampel & Perpustakaan.

AWS Data Pipeline dukungan tersedia SDKs untuk platform berikut:

- AWS SDK for Java
- AWS SDK untuk Node.js
- AWS SDK for PHP
- AWS SDK untuk Python (Boto)

- AWS SDK for Ruby
- AWS SDK for .NET

Membuat Permintaan HTTP ke AWS Data Pipeline

Untuk deskripsi lengkap tentang objek terprogram AWS Data Pipeline, lihat Referensi AWS Data Pipeline API.

Jika Anda tidak menggunakan salah satu AWS SDKs, Anda dapat melakukan AWS Data Pipeline operasi melalui HTTP menggunakan metode permintaan POST. Metode POST mengharuskan Anda untuk menentukan operasi di header permintaan dan memberikan data untuk operasi dalam format JSON dalam isi permintaan.

Konten Header HTTP

AWS Data Pipeline memerlukan informasi berikut di header permintaan HTTP:

- `host` AWS Data Pipeline Titik akhir.

  Untuk informasi lebih lanjut tentang titik akhir, lihat Wilayah dan Titik Akhir.
- `x-amz-date`Anda harus memberikan stempel waktu di header Tanggal HTTP atau x-amz-date header AWS. (Beberapa perpustakaan klien HTTP tidak mengizinkan Anda untuk mengatur header Tanggal.) Saat x-amz-date header hadir, sistem mengabaikan header Tanggal apa pun selama otentikasi permintaan.

  Tanggal harus ditentukan dalam salah satu dari tiga format berikut, seperti yang ditentukan dalam HTTP/1.1 RFC:
  - Min, 06 Nov 1994 08:49:37 GMT (RFC 822, diperbarui oleh RFC 1123)
  - Min, 06-Nov-94 08:49:37 GMT (RFC 850, diusangkan oleh RFC 1036)
  - Min 06 Nov 1994 08:49:37 (format asctime() ANSI C)
- `Authorization` Set parameter otorisasi yang AWS gunakan untuk memastikan validitas dan keaslian permintaan. Untuk informasi selengkapnya tentang cara membuat header ini, kunjungi Proses Tanda Tangan 4 4.
- `x-amz-target` Layanan tujuan dari permintaan dan operasi untuk data, dalam format: `<<serviceName>>_<<API version>>.<<operationName>>`

  Sebagai contoh, `DataPipeline_20121129.ActivatePipeline`.

- `content-type` Menentukan JSON dan versi. Sebagai contoh, `Content-Type: application/x-amz-json-1.0`.

Berikut ini adalah contoh header untuk permintaan HTTP untuk mengaktifkan alur.

```
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
```

Konten Isi HTTP

Isi permintaan HTTP berisi data untuk operasi yang ditentukan dalam header permintaan HTTP. Data harus diformat sesuai dengan skema data JSON untuk setiap API. AWS Data Pipeline Skema data AWS Data Pipeline JSON mendefinisikan jenis data dan parameter (seperti operator perbandingan dan konstanta enumerasi) yang tersedia untuk setiap operasi.

Format Isi Permintaan HTTP

Gunakan format data JSON untuk menyampaikan nilai-nilai data dan struktur data, secara bersamaan. Elemen dapat bersarang dalam elemen lain dengan menggunakan notasi braket. Contoh berikut menunjukkan permintaan untuk menempatkan definisi alur yang terdiri dari tiga objek dan slot yang sesuai mereka.

```
{
 "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"}

      ]
    }
  ]
}
```

Menangani Tanggapan HTTP

Berikut adalah beberapa header penting dalam respons HTTP, dan bagaimana Anda harus menangani respons tersebut dalam aplikasi Anda:

- HTTP/1.1—Header ini diikuti dengan kode status. Sebuah nilai kode 200 menunjukkan operasi yang berhasil. Nilai lain menunjukkan kesalahan.

- x-amzn- RequestId —Header ini berisi ID permintaan yang dapat Anda gunakan jika Anda perlu memecahkan masalah permintaan. AWS Data Pipeline Contoh ID permintaan adalah K2 QH8 DNOU9 FNA2 GDLL8 OBVV4 KQNSO5 AEMVJF66 07N97 Q9ASUAAJG.
- x-amz-crc32 -AWS Data Pipeline menghitung CRC32 checksum dari payload HTTP dan mengembalikan checksum ini di header 32. x-amz-crc Kami menyarankan Anda menghitung CRC32 checksum Anda sendiri di sisi klien dan membandingkannya dengan header x-amz-crc 32; jika checksum tidak cocok, ini mungkin menunjukkan bahwa data rusak saat transit. Jika hal ini terjadi, Anda harus mencoba kembali permintaan Anda.

Pengguna AWS SDK tidak perlu melakukan verifikasi ini secara manual, karena SDKs menghitung checksum setiap balasan dari Amazon DynamoDB dan secara otomatis mencoba lagi jika ketidakcocokan terdeteksi.

Contoh Permintaan dan Respons AWS Data Pipeline JSON

Contoh berikut menunjukkan permintaan untuk membuat alur baru. Kemudian menunjukkan AWS Data Pipeline respons, termasuk pengidentifikasi pipa dari pipa yang baru dibuat.

Permintaan POST HTTP

```
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 Respon

```
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"}
```

Keamanan di AWS Data Pipeline

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku AWS Data Pipeline, lihat AWS Services in Scope by Compliance Program .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Data Pipeline. Topik berikut menunjukkan cara mengonfigurasi AWS Data Pipeline untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan layanan AWS lain yang membantu Anda memantau dan mengamankan AWS Data Pipeline sumber daya Anda.

Topik

- Perlindungan Data di AWS Data Pipeline
- Identity and Access Management untuk AWS Data Pipeline
- Logging dan Monitoring di AWS Data Pipeline
- Respon Insiden di AWS Data Pipeline
- Validasi Kepatuhan untuk AWS Data Pipeline
- Ketahanan di AWS Data Pipeline
- Keamanan Infrastruktur di AWS Data Pipeline
- Konfigurasi dan Analisis Kerentanan di AWS Data Pipeline

Perlindungan Data di AWS Data Pipeline

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di AWS Data Pipeline. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Konten ini meliputi konfigurasi keamanan dan tugas-tugas pengelolaan untuk berbagai layanan Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensi dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami merekomendasikan TLS 1.2 atau versi yang lebih baru.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.
- AWS Data Pipeline mendukung IMDSv2 sumber daya Amazon EMR dan Amazon EC2 . Untuk menggunakan IMDSv2 dengan Amazon EMR, gunakan versi 5.23.1, 5.27.1, atau 5.32 atau yang lebih baru atau versi 6.2 atau yang lebih baru. Untuk informasi selengkapnya, lihat Mengonfigurasi permintaan layanan metadata ke EC2 instans dan Penggunaan Amazon. IMDSv2

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS Data Pipeline atau lainnya Layanan AWS

menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Identity and Access Management untuk AWS Data Pipeline

Kredensial keamanan Anda mengidentifikasi Anda pada layanan dalam AWS dan mengizinkan Anda untuk menggunakan sumber daya AWS, seperti alur Anda. Anda dapat menggunakan fitur AWS Data Pipeline dan AWS Identity and Access Management (IAM) untuk memungkinkan AWS Data Pipeline dan pengguna lain mengakses AWS Data Pipeline sumber daya Anda tanpa membagikan kredensi keamanan Anda.

Organizations dapat berbagi akses ke alur sehingga individu dalam organisasi yang dapat mengembangkan dan memelihara mereka secara kolaboratif. Namun, misalnya, mungkin perlu melakukan hal berikut:

- Kontrol pengguna mana yang dapat mengakses saluran pipa tertentu
- Melindungi alur produksi agar tidak disunting karena kesalahan
- Mengizinkan auditor memiliki akses hanya-baca ke alur, namun mencegahnya untuk melakukan perubahan

AWS Data Pipeline terintegrasi dengan AWS Identity and Access Management (IAM), yang menawarkan berbagai fitur:

- Buat pengguna dan grup di Anda Akun AWS.
- Bagikan AWS sumber daya Anda dengan mudah di antara pengguna di Anda Akun AWS.
- Menetapkan kredensial keamanan unik untuk setiap pengguna.
- Mengontrol setiap akses pengguna untuk layanan dan sumber daya.
- Dapatkan tagihan tunggal untuk semua pengguna di Akun AWS Anda.

Dengan menggunakan IAM dengan AWS Data Pipeline, Anda dapat mengontrol apakah pengguna di organisasi Anda dapat melakukan tugas menggunakan tindakan API tertentu dan apakah mereka dapat menggunakan sumber daya AWS tertentu. Anda dapat menggunakan kebijakan IAM

berdasarkan tanda alur dan kelompok pekerja untuk berbagi alur Anda dengan pengguna lain dan mengontrol tingkat akses yang mereka miliki.

Daftar Isi

- Kebijakan IAM untuk AWS Data Pipeline
- Contoh Kebijakan untuk AWS Data Pipeline
- Peran IAM untuk AWS Data Pipeline

Kebijakan IAM untuk AWS Data Pipeline

Secara default, entitas IAM tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS. Untuk mengizinkan entitas IAM membuat atau memodifikasi sumber daya dan melakukan tugas, Anda harus membuat kebijakan IAM yang memberikan izin kepada entitas IAM untuk menggunakan sumber daya spesifik dan tindakan API yang mereka perlukan, lalu melampirkan kebijakan tersebut ke entitas IAM yang memerlukan izin tersebut.

Saat Anda melampirkan kebijakan ke pengguna atau grup pengguna, kebijakan itu mengizinkan atau menolak izin pengguna untuk melakukan tugas yang ditentukan pada sumber daya yang ditentukan. Untuk informasi umum tentang kebijakan IAM, lihat Izin dan Kebijakan dalam panduan Panduan Pengguna IAM. Untuk informasi selengkapnya tentang cara mengelola dan membuat kebijakan IAM, lihat Mengelola Kebijakan IAM.

Daftar Isi

- Sintaks Kebijakan
- Mengontrol Akses ke Alur Menggunakan Tanda
- Mengontrol Akses ke Alur Menggunakan Grup Pekerja

Sintaks Kebijakan

kebijakan IAM adalah dokumen JSON yang terdiri dari satu atau beberapa pernyataan. Setiap pernyataan memiliki struktur sebagai berikut:

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

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

Unsur-unsur berikut membuat pernyataan kebijakan:

- Efek: Efek bisa jadi `Allow` atau `Deny`. Secara default, entitas IAM tidak memiliki izin untuk menggunakan sumber daya dan tindakan API, sehingga semua permintaan ditolak. izin eksplisit akan menggantikan izin default. penolakan eksplisit akan menggantikan izin apa pun.
- Tindakan: Tindakan adalah tindakan API tertentu yang Anda izinkan atau tolak. Untuk daftar tindakan AWS Data Pipeline, lihat Tindakan di Referensi AWS Data Pipeline API.
- Sumber daya: Sumber daya yang dipengaruhi oleh tindakan. Satu-satunya nilai yang valid adalah `"*"`.
- Syarat: Syarat-syarat bersifat opsional. Ketentuan ini dapat digunakan untuk mengontrol kapan kebijakan Anda akan berlaku.

  AWS Data Pipeline mengimplementasikan kunci konteks seluruh AWS (lihat Kunci yang Tersedia untuk Kondisi), ditambah kunci khusus layanan berikut.

  - `datapipeline:PipelineCreator` — Untuk memberikan akses ke pengguna yang membuat alur. Sebagai contoh, lihat Berikan akses penuh kepada pemilik alur.
  - `datapipeline:Tag` — Untuk memberikan akses berdasarkan penandaan alur. Untuk informasi selengkapnya, lihat Mengontrol Akses ke Alur Menggunakan Tanda.
  - `datapipeline:workerGroup` — Untuk memberikan akses berdasarkan nama kelompok pekerja. Untuk informasi selengkapnya, lihat Mengontrol Akses ke Alur Menggunakan Grup Pekerja.

Mengontrol Akses ke Alur Menggunakan Tanda

Anda dapat membuat kebijakan IAM yang mereferensi tanda untuk alur Anda. Hal ini memungkinkan Anda untuk menggunakan alur penandaan untuk melakukan hal berikut:

- Memberikan akses hanya baca ke alur

- Berikan read/write akses ke pipa
- Memblokir akses ke alur

Sebagai contoh, misalkan seorang manajer memiliki dua lingkungan alur, produksi dan pengembangan, dan grup IAM untuk setiap lingkungan. Untuk jaringan pipa di lingkungan produksi, manajer memberikan read/write akses ke pengguna di grup IAM produksi, tetapi memberikan akses hanya-baca ke pengguna di grup IAM pengembang. Untuk jaringan pipa di lingkungan pengembangan, manajer memberikan read/write akses ke grup IAM produksi dan pengembang.

Untuk mencapai skenario ini, manajer menandai alur produksi dengan tanda “"environment=production" dan menempel kebijakan berikut untuk grup IAM developer. Pernyataan pertama memberikan akses hanya baca ke semua alur. Pernyataan kedua memberikan read/write akses ke pipeline yang tidak memiliki 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"}
      }
    }
  ]
}
```

Selain itu, manajer melampirkan kebijakan berikut ke grup IAM produksi. Pernyataan ini memberikan akses penuh ke semua alur.

JSON

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

Untuk contoh lainnya, lihat Berikan akses hanya-baca kepada pengguna berdasarkan tanda dan Berikan akses penuh kepada pengguna berdasarkan tanda.

Mengontrol Akses ke Alur Menggunakan Grup Pekerja

Anda dapat membuat kebijakan IAM yang membuat nama grup referensi pekerja.

Sebagai contoh, misalkan seorang manajer memiliki dua lingkungan alur, produksi dan pengembangan, dan grup IAM untuk setiap lingkungan. Manajer memiliki tiga server basis data dengan tugas pelari dikonfigurasi masing-masing untuk produksi, pra-produksi, dan lingkungan developer. Manajer ingin memastikan bahwa pengguna dalam grup IAM produksi dapat membuat alur yang mendorong tugas untuk sumber daya produksi, dan bahwa pengguna dalam grup IAM pengembangan dapat membuat alur yang mendorong tugas untuk kedua sumber daya pra-produksi dan developer.

Untuk mencapai skenario ini, manajer menginstal tugas pelari pada sumber daya produksi dengan kredensial produksi, dan mengatur `workerGroup` ke “prodresource”. Di samping itu, pengurus menginstal tugas pelari pada sumber daya pengembangan dengan kredensial pengembangan, dan mengatur `workerGroup` ke “pra-produksi” dan “pengembangan”. Manajer melampirkan kebijakan berikut ke grup IAM developer untuk memblokir akses ke sumber daya “prodresource”. Pernyataan pertama memberikan akses hanya baca ke semua alur. Pernyataan kedua memberikan read/write akses ke saluran pipa ketika nama kelompok pekerja memiliki awalan “dev” atau “pre-prod”.

Sebagai tambahan, manajer melampirkan kebijakan berikut, ke grup IAM produksi untuk memberikan akses ke sumber daya “prodresource”. Pernyataan pertama memberikan akses hanya baca ke semua alur. Pernyataan kedua memberikan read/write akses ketika nama grup pekerja memiliki awalan “prod”.

Contoh Kebijakan untuk AWS Data Pipeline

Contoh berikut ini menunjukkan cara memberikan akses penuh atau terbatas pada pengguna ke alur.

Daftar Isi

- Contoh 1: Memberikan pengguna akses hanya-baca berdasarkan tanda
- Contoh 2: Memberikan pengguna akses penuh berdasarkan tanda
- Contoh 3: Memberikan akses penuh untuk pemilik alur
- Contoh 4: Berikan pengguna akses ke AWS Data Pipeline konsol

Contoh 1: Memberikan pengguna akses hanya-baca berdasarkan tanda

Kebijakan berikut memungkinkan pengguna untuk menggunakan tindakan AWS Data Pipeline API hanya-baca, tetapi hanya dengan pipeline yang memiliki tag “environment=production”.

Tindakan ListPipelines API tidak mendukung otorisasi berbasis 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"
        }
      }
    }
  ]
}
```

Contoh 2: Memberikan pengguna akses penuh berdasarkan tanda

Kebijakan berikut memungkinkan pengguna untuk menggunakan semua tindakan AWS Data Pipeline API, kecuali ListPipelines, tetapi hanya dengan pipeline yang memiliki tag “environment=test”.

JSON

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

Contoh 3: Memberikan akses penuh untuk pemilik alur

Kebijakan berikut memungkinkan pengguna untuk menggunakan semua tindakan AWS Data Pipeline API, tetapi hanya dengan pipeline mereka sendiri.

Contoh 4: Berikan pengguna akses ke AWS Data Pipeline konsol

Kebijakan berikut memungkinkan pengguna untuk membuat dan mengelola alur dengan menggunakan konsol AWS Data Pipeline .

Kebijakan ini mencakup tindakan untuk `PassRole` izin untuk sumber daya tertentu yang terkait dengan AWS Data Pipeline kebutuhan `role`ARN tersebut. Untuk informasi selengkapnya tentang `PassRole` izin berbasis identitas (IAM), lihat posting blog Memberikan Izin untuk Meluncurkan EC2 Instans dengan Peran IAM (Izin). 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"
    ]
   }
  ]
}
```

Peran IAM untuk AWS Data Pipeline

AWS Data Pipeline menggunakan AWS Identity and Access Management peran. Kebijakan izin yang dilampirkan pada peran IAM menentukan tindakan AWS Data Pipeline dan aplikasi apa yang dapat dilakukan, dan AWS sumber daya apa yang dapat mereka akses. Untuk informasi selengkapnya, lihat peran IAM dalam Panduan Pengguna IAM.

AWS Data Pipeline membutuhkan dua peran IAM:

- Peran pipeline mengontrol AWS Data Pipeline akses ke sumber daya AWS Anda. Dalam definisi objek alur, bidang `role` menentukan peran ini.
- Peran EC2 instans mengontrol akses yang dimiliki aplikasi yang berjalan pada EC2 instance, termasuk EC2 instans di kluster EMR Amazon, ke sumber daya. AWS Dalam definisi objek alur, bidang `resourceRole` menentukan peran ini.

> **Important**
>
> Jika Anda membuat pipeline sebelum 3 Oktober 2022 menggunakan AWS Data Pipeline konsol dengan peran default, AWS Data Pipeline buat `DataPipelineDefaultRole` untuk Anda dan lampirkan kebijakan `AWSDataPipelineRole` terkelola ke peran tersebut. Mulai 3 Oktober 2022, kebijakan `AWSDataPipelineRole` terkelola tidak digunakan lagi dan peran pipeline harus ditentukan untuk pipeline saat menggunakan konsol.
> Kami merekomendasikan bahwa Anda meninjau jaringan alur yang ada dan menentukan apakah `DataPipelineDefaultRole` dikaitkan dengan alur dan apakah `AWSDataPipelineRole` dilampirkan pada peran tersebut. Jika demikian, tinjau akses yang diizinkan kebijakan ini untuk memastikan sesuai dengan persyaratan keamanan Anda. Menambahkan, memperbarui, atau mengganti kebijakan dan pernyataan kebijakan yang

melekat pada peran ini diperlukan. Atau, Anda dapat memperbarui alur untuk menggunakan peran yang Anda buat dengan kebijakan izin yang berbeda.

Contoh Kebijakan Izin untuk AWS Data Pipeline Peran

Setiap peran memiliki satu kebijakan izin atau lebih yang dilampirkan padanya yang menentukan sumber daya AWS yang dapat diakses peran tersebut dan tindakan yang dapat dilakukan peran tersebut. Topik ini menyediakan contoh kebijakan izin untuk peran alur. Ini juga menyediakan kontenAmazonEC2RoleforDataPipelineRole, yang merupakan kebijakan terkelola untuk peran EC2 instance default,DataPipelineDefaultResourceRole.

Contoh Kebijakan Izin Peran Alur

Contoh kebijakan berikut dicakup untuk mengizinkan fungsi-fungsi penting yang AWS Data Pipeline diperlukan untuk menjalankan pipeline dengan sumber daya EMR EC2 Amazon dan Amazon. Ini juga menyediakan izin untuk mengakses AWS sumber daya lain, seperti Amazon Simple Storage Service dan Amazon Simple Notification Service, yang dibutuhkan banyak saluran pipa. Jika objek yang didefinisikan dalam pipeline tidak memerlukan sumber daya AWS layanan, kami sangat menyarankan Anda menghapus izin untuk mengakses layanan tersebut. Misalnya, jika alur Anda tidak mendefinisikan Node Dinamo DBData atau menggunakan tindakan SnsAlarm, kami sarankan Anda menghapus pernyataan izin untuk tindakan tersebut.

- Ganti *111122223333* dengan ID AWS akun Anda.
- Ganti *NameOfDataPipelineRole* dengan nama peran alur (peran yang dilampirkan kebijakan ini).
- Ganti *NameOfDataPipelineResourceRole* dengan nama peran EC2 instance.
- Ganti *us-west-1* dengan Wilayah yang sesuai untuk aplikasi Anda.

Kebijakan Terkelola Default untuk Peran EC2 Instans

Isi dari AmazonEC2RoleforDataPipelineRole ditunjukkan di bawah ini. Ini adalah kebijakan terkelola yang dilampirkan ke peran sumber daya default untuk AWS Data Pipeline,DataPipelineDefaultResourceRole. Saat menentukan peran sumber daya untuk pipeline, sebaiknya mulai dengan kebijakan izin ini, lalu hapus izin untuk tindakan AWS layanan yang tidak diperlukan.

Versi 3 dari kebijakan ditampilkan, yang merupakan versi terbaru pada saat penulisan ini. Lihat versi terbaru kebijakan menggunakan konsol 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": ["*"]
    }]
}
```

Membuat Peran IAM untuk AWS Data Pipeline dan Mengedit Izin Peran

Gunakan prosedur berikut untuk membuat peran untuk AWS Data Pipeline menggunakan konsol IAM. Prosesnya terdiri atas dua langkah. Pertama, Anda membuat kebijakan izin untuk dilampirkan ke peran tersebut. Selanjutnya, Anda membuat peran dan melampirkan kebijakan tersebut. Setelah membuat peran, Anda dapat mengubah izin peran dengan melampirkan dan memisahkan kebijakan izin.

> **Note**
>
> Saat Anda membuat peran untuk AWS Data Pipeline menggunakan konsol seperti yang dijelaskan di bawah ini, IAM membuat dan melampirkan kebijakan kepercayaan yang sesuai yang diperlukan peran tersebut.

Untuk membuat kebijakan izin untuk digunakan dengan peran AWS Data Pipeline

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pada panel navigasi, pilih Kebijakan, lalu pilih Buat kebijakan.
3. Pilih tab JSON.
4. Jika Anda menciptakan peran alur, salin dan tempelkan isi contoh kebijakan dalam Contoh Kebijakan Izin Peran Alur, mengedit sesuai dengan persyaratan keamanan Anda. Atau, jika Anda membuat peran EC2 instance khusus, lakukan hal yang sama untuk contoh diKebijakan Terkelola Default untuk Peran EC2 Instans.
5. Pilih Tinjau kebijakan.
6. Masukkan nama untuk kebijakan—misalnya, `MyDataPipelineRolePolicy`—dan sebuah pilihan Deskripsi, lalu pilih Buat kebijakan.
7. Ingat nama kebijakan. Anda memerlukannya saat Anda membuat peran Anda.

Untuk membuat peran IAM untuk AWS Data Pipeline

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran, lalu pilih Buat peran.
3. Di bawah Pilih kasus penggunaan, pilih Data Pipeline.
4. Di bawah Pilih kasus penggunaan Anda, lakukan salah satu langkah berikut:
   - Pilih `Data Pipeline` untuk membuat peran alur.
   - Pilih `EC2 Role for Data Pipeline` untuk membuat peran sumber daya.
5. Pilih Berikutnya: Izin.
6. Jika kebijakan default untuk AWS Data Pipeline terdaftar, lanjutkan dengan langkah-langkah berikut untuk membuat peran dan kemudian mengeditnya sesuai dengan instruksi di prosedur

berikutnya. Jika tidak, masukkan nama kebijakan yang Anda buat dalam prosedur di atas, dan kemudian pilih dari daftar.

7. Pilih Berikutnya: Tanda, masukkan tanda apa pun untuk ditambahkan ke peran, lalu pilih Berikutnya: Ulasan.
8. Masukkan nama untuk peran—misalnya, `MyDataPipelineRole`—dan sebuah pilihan Deskripsi, lalu pilih Buat peran.

Untuk melampirkan atau melepaskan kebijakan izin untuk peran IAM AWS Data Pipeline

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran
3. Di kotak pencarian, mulailah mengetik nama peran yang ingin Anda edit—misalnya, DataPipelineDefaultRoleMyDataPipelineRoleatau—lalu pilih nama Peran dari daftar.
4. Di tab Izin, lakukan hal berikut:
   - Untuk melepaskan kebijakan izin, di bawah Kebijakan izin, pilih tombol hapus di ujung kanan entri kebijakan. Pilih Lepaskan saat diminta untuk mengonfirmasi.
   - Untuk melampirkan kebijakan yang Anda buat sebelumnya, pilih Lampirkan kebijakan. Di kotak pencarian, mulai ketik nama kebijakan yang ingin Anda edit, pilih dari daftar, lalu pilih Lampirkan kebijakan.

Mengubah Peran untuk Alur yang sudah Ada

Jika Anda ingin menetapkan peran pipeline atau peran sumber daya yang berbeda ke pipeline, Anda dapat menggunakan editor arsitek di AWS Data Pipeline konsol.

Untuk mengedit peran yang ditetapkan ke alur menggunakan konsol

1. Buka AWS Data Pipeline konsol di https://console.aws.amazon.com/datapipeline/.
2. Pilih alur dari daftar, lalu pilih Tindakan, Edit.
3. Di panel kanan editor arsitek, pilih Lainnya.
4. Dari daftar Peran dan Peran Sumber Daya, pilih peran AWS Data Pipeline yang ingin Anda tetapkan, lalu pilih Simpan.

Logging dan Monitoring di AWS Data Pipeline

AWS Data Pipeline terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS Data Pipeline. CloudTrail menangkap semua panggilan API untuk AWS Data Pipeline sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari AWS Data Pipeline konsol dan panggilan kode ke operasi AWS Data Pipeline API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk. AWS Data Pipeline Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat AWS Data Pipeline, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

AWS Data Pipeline Informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di AWS Data Pipeline, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS . Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk AWS Data Pipeline, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, ketika Anda membuat jejak di konsol tersebut, jejak tersebut diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

Semua AWS Data Pipeline tindakan dicatat oleh CloudTrail dan didokumentasikan di bagian Tindakan Referensi API AWS Data Pipeline API. Misalnya, panggilan ke CreatePipelinetindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan dibuat dengan kredenal peran root atau IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

Memahami Entri File AWS Data Pipeline Log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreatePipeline` operasi:

```
    {
"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
  ]
}
```

Respon Insiden di AWS Data Pipeline

Respon insiden untuk AWS Data Pipeline adalah AWS tanggung jawab. AWS memiliki kebijakan dan program formal yang terdokumentasi yang mengatur respons insiden.

Masalah operasional AWS dengan dampak luas di-posting pada AWS Service Health Dashboard. Masalah operasional juga di-posting ke akun individu melalui Personal Health Dashboard.

Validasi Kepatuhan untuk AWS Data Pipeline

AWS Data Pipeline tidak dalam lingkup program kepatuhan AWS apa pun. Untuk daftar layanan AWS dalam cakupan program kepatuhan tertentu, lihat Layanan AWS dalam Cakupan oleh Program Kepatuhan. Untuk informasi umum, lihat Program Kepatuhan AWS.

Ketahanan di AWS Data Pipeline

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang

secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan Infrastruktur di AWS Data Pipeline

Sebagai layanan terkelola, AWS Data Pipeline dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper Amazon Web Services: Tinjauan Proses Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS Data Pipeline melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.0 atau versi yang lebih baru. Kami merekomendasikan TLS 1.2 atau versi yang lebih baru. Klien juga harus mendukung suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan principal IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Konfigurasi dan Analisis Kerentanan di AWS Data Pipeline

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara AWS dan Anda, pelanggan kami. Untuk informasi selengkapnya, lihat model tanggung jawab AWS bersama.

Tutorial

Tutorial berikut memandu Anda step-by-step melalui proses membuat dan menggunakan saluran pipa dengan AWS Data Pipeline.

Tutorial

- Proses Data Menggunakan Amazon EMR dengan Hadoop Streaming
- Salin Data CSV Antara Bucket Amazon S3 Menggunakan AWS Data Pipeline
- Ekspor Data MySQL ke Amazon S3 Menggunakan AWS Data Pipeline
- Salin Data ke Amazon Redshift Menggunakan AWS Data Pipeline

Proses Data Menggunakan Amazon EMR dengan Hadoop Streaming

Anda dapat menggunakan AWS Data Pipeline untuk mengelola cluster EMR Amazon Anda. Dengan AWS Data Pipeline Anda dapat menentukan prasyarat yang harus dipenuhi sebelum cluster diluncurkan (misalnya, memastikan bahwa data hari ini telah diunggah ke Amazon S3), jadwal untuk menjalankan cluster berulang kali, dan konfigurasi cluster yang akan digunakan. Tutorial berikut memandu Anda melalui meluncurkan klaster sederhana.

Dalam tutorial ini, Anda membuat alur untuk klaster Amazon EMR sederhana untuk menjalankan pekerjaan Hadoop Streaming yang sudah ada yang disediakan oleh Amazon EMR dan mengirim notifikasi Amazon SNS setelah tugas selesai dengan sukses. Anda menggunakan sumber daya kluster EMR Amazon yang disediakan oleh AWS Data Pipeline untuk tugas ini. Aplikasi sampel disebut WordCount, dan juga dapat dijalankan secara manual dari konsol EMR Amazon. Perhatikan bahwa klaster yang muncul atas nama Anda ditampilkan AWS Data Pipeline di konsol EMR Amazon dan ditagih ke akun AWS Anda.

Objek Alur

Alur menggunakan objek berikut:

EmrActivity

Mendefinisikan pekerjaan untuk melakukan dalam alur (menjalankan pekerjaan Hadoop Streaming yang sudah ada yang disediakan oleh Amazon EMR).

EmrCluster

Sumber daya AWS Data Pipeline digunakan untuk melakukan kegiatan ini.

Cluster adalah sekumpulan EC2 instance Amazon. AWS Data Pipeline meluncurkan cluster dan kemudian menghentikannya setelah tugas selesai.

Jadwal

Tanggal mulai, waktu, dan durasi untuk aktivitas ini. Anda juga dapat menentukan tanggal dan waktu akhir.

SnsAlarm

Mengirimkan notifikasi Amazon SNS ke topik yang Anda tentukan setelah tugas selesai dengan sukses.

Daftar Isi

- Sebelum Anda Memulai
- Luncurkan Klaster Menggunakan Baris Perintah

Sebelum Anda Memulai

Pastikan Anda telah menyelesaikan langkah-langkah berikut.

- Selesaikan tugas dalam Menyiapkan untuk AWS Data Pipeline.
- (Opsional) Mengatur VPC untuk klaster dan grup keamanan untuk VPC.
- Membuat topik untuk mengirim notifikasi email dan membuat catatan dari topik Amazon Resource Name (ARN). Untuk informasi lebih lanjut, lihat Buat Topik di Panduan Memulai Amazon Simple Notification Service.

Luncurkan Klaster Menggunakan Baris Perintah

Jika Anda secara teratur menjalankan klaster EMR Amazon untuk menganalisis log web atau melakukan analisis data ilmiah, Anda dapat menggunakannya AWS Data Pipeline untuk mengelola kluster EMR Amazon Anda. Dengan AWS Data Pipeline, Anda dapat menentukan prasyarat yang harus dipenuhi sebelum cluster diluncurkan (misalnya, memastikan bahwa data hari ini telah diunggah ke Amazon S3.) Tutorial ini memandu Anda melalui meluncurkan klaster yang dapat

menjadi model untuk alur berbasis Amazon EMR sederhana, atau sebagai bagian dari alur yang lebih terlibat.

Prasyarat

Sebelum Anda dapat menggunakan CLI, Anda harus menyelesaikan langkah-langkah berikut:

1. Pasang dan konfigurasi antarmuka baris perintah (CLI). Untuk informasi selengkapnya, lihat Mengakses AWS Data Pipeline.
2. Pastikan bahwa peran IAM diberi nama DataPipelineDefaultRoledan DataPipelineDefaultResourceRoleada. AWS Data Pipeline Konsol membuat peran ini untuk Anda secara otomatis. Jika Anda belum pernah menggunakan AWS Data Pipeline konsol setidaknya sekali, maka Anda harus membuat peran ini secara manual. Untuk informasi selengkapnya, lihat Peran IAM untuk AWS Data Pipeline.

Tugas

- Membuat File Definisi Alur
- Mengunggah dan Mengaktifkan Definisi Alur
- Pantau Alur Berjalan

Membuat File Definisi Alur

Kode berikut adalah file definisi alur untuk klaster Amazon EMR sederhana yang menjalankan tugas streaming Hadoop yang sudah ada yang disediakan oleh Amazon EMR. Contoh aplikasi ini disebut WordCount, dan Anda juga dapat menjalankannya menggunakan konsol EMR Amazon.

Salin kode ini ke dalam file teks dan simpan sebagai `MyEmrPipelineDefinition.json`. Anda harus mengganti lokasi bucket Amazon S3 dengan nama bucket Amazon S3 yang Anda miliki. Anda juga harus mengganti tanggal mulai dan akhir. Untuk segera meluncurkan cluster, atur `startDateTime` ke tanggal satu hari di masa lalu dan `endDateTime` ke satu hari di masa depan. AWS Data Pipeline kemudian mulai meluncurkan cluster “past due” segera dalam upaya untuk mengatasi apa yang dianggapnya sebagai tumpukan pekerjaan. Penimbunan ulang ini berarti Anda tidak perlu menunggu satu jam untuk melihat AWS Data Pipeline peluncuran cluster pertamanya.

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

Alur ini memiliki tiga objek:

- `Hourly`, yang mewakili jadwal pekerjaan. Anda dapat mengatur jadwal sebagai salah satu bidang pada suatu aktivitas. Ketika Anda melakukannya, aktivitas berjalan sesuai dengan jadwal itu, atau dalam hal ini, per jam.
- `MyCluster`, yang mewakili kumpulan EC2 instance Amazon yang digunakan untuk menjalankan cluster. Anda dapat menentukan ukuran dan jumlah EC2 instance yang akan dijalankan sebagai cluster. Jika Anda tidak menentukan jumlah instans, klaster meluncurkan dengan dua, simpul utama dan simpul tugas. Anda dapat menentukan subnet untuk meluncurkan klasternya. Anda dapat menambahkan konfigurasi tambahan untuk klaster, seperti tindakan bootstrap untuk memuat perangkat lunak tambahan ke Amazon EMR disediakan AMI.

- `MyEmrActivity`, yang merupakan perhitungan untuk memproses dengan klaster. Amazon EMR mendukung beberapa jenis klaster, termasuk streaming, Cascading, dan Scripted Hive. `runsOn`Bidang mengacu kembali ke MyCluster, menggunakannya sebagai spesifikasi untuk dasar-dasar cluster.

Mengunggah dan Mengaktifkan Definisi Alur

Anda harus mengunggah definisi alur Anda dan mengaktifkan alur Anda. Dalam contoh perintah berikut, ganti *`pipeline_name`* dengan label untuk pipeline Anda dan *`pipeline_file`* dengan jalur yang sepenuhnya memenuhi syarat untuk file definisi `.json` pipeline.

AWS CLI

Untuk membuat definisi alur Anda dan mengaktifkan alur Anda, gunakan perintah create-pipeline. Perhatikan ID alur Anda, karena Anda akan menggunakan nilai ini dengan sebagian besar perintah CLI.

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

Untuk mengunggah definisi pipeline Anda, gunakan put-pipeline-definitionperintah berikut.

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

Jika validasi alur Anda berhasil, bidang `validationErrors` akan kosong. Anda harus meninjau peringatan apa pun.

Untuk mengaktifkan alur Anda, gunakan perintah activate-pipeline.

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

Anda dapat memverifikasi bahwa alur Anda muncul dalam daftar alur menggunakan perintah list-pipelines berikut.

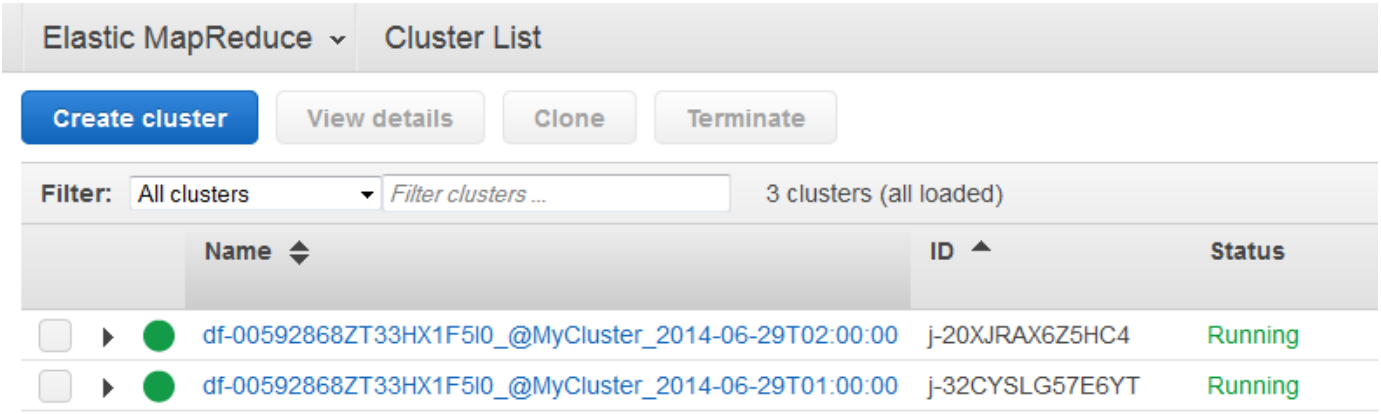
```
aws datapipeline list-pipelines
```

Pantau Alur Berjalan

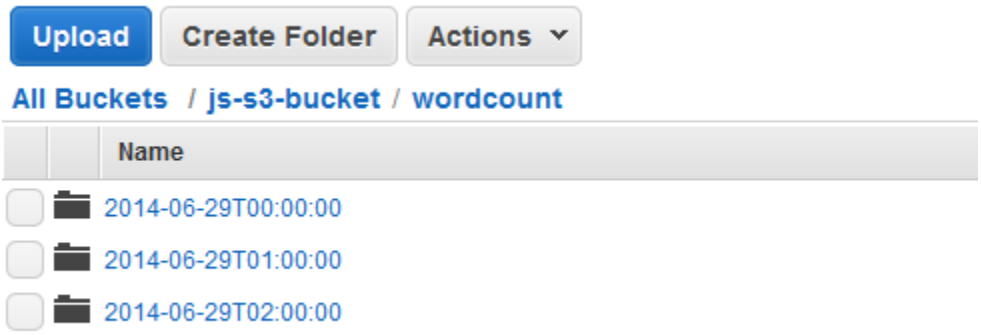
Anda dapat melihat cluster yang diluncurkan dengan AWS Data Pipeline menggunakan konsol Amazon EMR dan Anda dapat melihat folder output menggunakan konsol Amazon S3.

Untuk memeriksa kemajuan cluster yang diluncurkan oleh AWS Data Pipeline

1. Buka konsol Amazon EMR.
2. Cluster yang muncul AWS Data Pipeline memiliki nama yang diformat sebagai berikut: _@ _. *<pipeline-identifier> <emr-cluster-name> <launch-time>*



3. Setelah salah satu berjalan selesai, buka konsol Amazon S3 dan periksa bahwa folder output tertera waktu ada dan berisi hasil yang diharapkan dari klaster.



Salin Data CSV Antara Bucket Amazon S3 Menggunakan AWS Data Pipeline

Setelah Anda membaca Apa itu AWS Data Pipeline? dan memutuskan Anda ingin menggunakan AWS Data Pipeline untuk mengotomatiskan pergerakan dan transformasi data Anda, sekarang

saatnya untuk memulai dengan membuat pipeline data. Untuk membantu Anda memahami bagaimana AWS Data Pipeline bekerja, mari kita perhatikan melalui tugas sederhana.

Tutorial ini memandu Anda melalui proses membuat alur data untuk menyalin data dari satu bucket Amazon S3 ke yang lain dan kemudian mengirim notifikasi Amazon SNS setelah aktivitas salin selesai dengan sukses. Anda menggunakan EC2 instance yang dikelola oleh AWS Data Pipeline untuk aktivitas penyalinan ini.

Objek Alur

Alur menggunakan objek berikut:

CopyActivity

Aktivitas yang AWS Data Pipeline dilakukan untuk pipeline ini (salin data CSV dari satu bucket Amazon S3 ke bucket Amazon S3 lainnya).

> **Important**
>
> Ada keterbatasan saat menggunakan format file CSV dengan `CopyActivity` dan `S3DataNode`. Untuk informasi selengkapnya, lihat CopyActivity.

Jadwal

Tanggal mulai, waktu, dan pengulangan untuk kegiatan ini. Anda juga dapat menentukan tanggal dan waktu akhir.

Ec2Resource

Sumber daya (EC2 instance) yang AWS Data Pipeline digunakan untuk melakukan aktivitas ini.

S3 DataNode

Simpul input dan output (bucket Amazon S3) untuk alur ini.

SnsAlarm

Tindakan AWS Data Pipeline harus diambil ketika kondisi yang ditentukan terpenuhi (kirim notifikasi Amazon SNS ke topik setelah tugas selesai dengan sukses).

Daftar Isi

- Sebelum Anda Memulai

- Salin Data CSV Menggunakan Baris Perintah

Sebelum Anda Memulai

Pastikan Anda telah menyelesaikan langkah-langkah berikut.

- Selesaikan tugas dalam Menyiapkan untuk AWS Data Pipeline.
- (Opsional) Mengatur VPC untuk instans dan grup keamanan untuk VPC.
- Buat bucket Amazon S3 sebagai sumber data.

  Untuk informasi selengkapnya, lihat Membuat Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Unggah data Anda ke bucket Amazon S3.

  Untuk informasi selengkapnya, lihat Menambahkan Objek ke Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Buat bucket Amazon S3 lain sebagai target data
- Membuat topik untuk mengirim notifikasi email dan membuat catatan dari topik Amazon Resource Name (ARN). Untuk informasi lebih lanjut, lihat Buat Topik di Panduan Memulai Amazon Simple Notification Service.
- (Opsional) Tutorial ini menggunakan kebijakan IAM role default yang dibuat oleh AWS Data Pipeline. Jika Anda lebih suka membuat dan mengonfigurasi kebijakan IAM role dan hubungan kepercayaan Anda sendiri, ikuti petunjuk yang dijelaskan di Peran IAM untuk AWS Data Pipeline.

Salin Data CSV Menggunakan Baris Perintah

Anda dapat membuat dan menggunakan alur untuk menyalin data dari satu bucket Amazon S3 ke lainnya.

Prasyarat

Sebelum memulai tutorial ini, Anda harus menyelesaikan langkah berikut:

1. Pasang dan konfigurasi antarmuka baris perintah (CLI). Untuk informasi selengkapnya, lihat Mengakses AWS Data Pipeline.
2. Pastikan bahwa peran IAM diberi nama DataPipelineDefaultRoledan DataPipelineDefaultResourceRoleada. AWS Data Pipeline Konsol membuat peran ini untuk

Anda secara otomatis. Jika Anda belum pernah menggunakan AWS Data Pipeline konsol setidaknya sekali, maka Anda harus membuat peran ini secara manual. Untuk informasi selengkapnya, lihat Peran IAM untuk AWS Data Pipeline.

Tugas

- Mendefinisikan Alur dalam Format JSON
- Unggah dan Aktifkan Definisi Alur

Mendefinisikan Alur dalam Format JSON

Contoh skenario ini menunjukkan bagaimana menggunakan definisi alur JSON dan AWS Data Pipeline CLI untuk menjadwalkan menyalin data antara dua bucket Amazon S3 pada interval waktu tertentu. Ini adalah file JSON definisi alur lengkap diikuti dengan penjelasan untuk setiap bagiannya.

> **Note**
>
> Kami merekomendasikan bahwa Anda menggunakan editor teks yang dapat membantu Anda memverifikasi sintaks file yang diformat JSON, dan nama file menggunakan ekstensi file .json.

Dalam contoh ini, untuk kejelasan, kita melewatkan bidang opsional dan hanya menampilkan bidang yang diperlukan. File JSON alur lengkap untuk contoh ini adalah:

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

Jadwal

Alur mendefinisikan jadwal dengan tanggal mulai dan akhir, bersama dengan periode untuk menentukan seberapa sering aktivitas dalam alur ini berjalan.

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

Simpul Data Amazon S3

Selanjutnya, komponen DataNode pipeline input S3 menentukan lokasi untuk file input; dalam hal ini, lokasi bucket Amazon S3. DataNodeKomponen input S3 didefinisikan oleh bidang-bidang berikut:

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

Id

Nama yang ditetapkan pengguna untuk lokasi input (label untuk referensi Anda saja).

Tipe

Jenis komponen pipeline, yaitu “S3DataNode” agar sesuai dengan lokasi di mana data berada, di bucket Amazon S3.

Jadwal

Referensi ke komponen jadwal yang kita buat di baris sebelumnya dari file JSON berlabel “”. MySchedule

Jalan

Jalan ke data yang terkait dengan simpul data. Sintaks untuk simpul data ditentukan oleh tipenya. Sebagai contoh, sintaks untuk jalur Amazon S3 mengikuti sintaks yang berbeda yang sesuai untuk tabel basis data.

Selanjutnya, DataNode komponen output S3 mendefinisikan lokasi tujuan output untuk data. Ini mengikuti format yang sama dengan DataNode komponen input S3, kecuali nama komponen dan jalur yang berbeda untuk menunjukkan file target.

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

Sumber daya

Ini adalah definisi sumber daya komputasi yang melakukan operasi penyalinan. Dalam contoh ini, AWS Data Pipeline harus secara otomatis membuat sebuah EC2 instance untuk melakukan tugas copy dan mengakhiri sumber daya setelah tugas selesai. Bidang yang didefinisikan di sini mengontrol pembuatan dan fungsi EC2 instance yang melakukan pekerjaan. Sumber EC2 daya didefinisikan oleh bidang-bidang berikut:

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

Id

Nama yang ditetapkan pengguna untuk jadwal alur, yang merupakan label untuk referensi Anda saja.

Tipe

Jenis sumber daya komputasi untuk melakukan pekerjaan; dalam hal ini, sebuah EC2 instance. Ada jenis sumber daya lain yang tersedia, seperti EmrCluster tipe.

Jadwal

Jadwal untuk membuat sumber daya komputasi ini.

instanceType

Ukuran EC2 instance yang akan dibuat. Pastikan Anda menetapkan ukuran EC2 instance yang sesuai yang paling sesuai dengan beban pekerjaan yang ingin Anda lakukan AWS Data Pipeline. Dalam hal ini, kami menetapkan instance m1.medium EC2. Untuk informasi selengkapnya tentang jenis instans yang berbeda dan kapan menggunakannya, lihat topik Jenis EC2 Instance Amazon di http://aws.amazon. com/ec2/instance-jenis/.

Peran

IAM role akun yang mengakses sumber daya, seperti mengakses bucket Amazon S3 untuk mengambil data.

resourceRole

Peran IAM dari akun yang menciptakan sumber daya, seperti membuat dan mengonfigurasi EC2 instance atas nama Anda. Peran dan ResourceRole dapat menjadi peran yang sama, tetapi secara terpisah memberikan perincian yang lebih besar dalam konfigurasi keamanan Anda.

Aktifitas

Bagian terakhir dalam file JSON yang merupakan definisi dari aktivitas yang mewakili pekerjaan yang akan dilakukan. Contoh ini digunakan `CopyActivity` untuk menyalin data dari file CSV di http://aws.amazon. com/ec2/instance-jenis/ember ke yang lain. Komponen `CopyActivity` didefinisikan oleh bidang berikut:

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

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

Id

Nama yang ditetapkan pengguna untuk aktivitas, yang merupakan label untuk referensi Anda saja.

Tipe

Jenis kegiatan yang harus dilakukan, seperti MyCopyActivity.

runsOn

Sumber daya komputasi yang melakukan pekerjaan yang ditentukan oleh aktivitas ini. Dalam contoh ini, kami memberikan referensi ke EC2 instance yang didefinisikan sebelumnya. Menggunakan `runs0n` bidang menyebabkan AWS Data Pipeline untuk membuat EC2 instance untuk Anda. Bidang `runs0n` menunjukkan bahwa sumber daya yang ada di infrastruktur AWS, sedangkan nilai `workerGroup` tersebut menunjukkan bahwa Anda ingin menggunakan sumber daya lokal Anda sendiri untuk melakukan pekerjaan.

Input

Lokasi data yang akan disalin.

Output

Data lokasi target.

Jadwal

Jadwal untuk menjalankan kegiatan ini.

Unggah dan Aktifkan Definisi Alur

Anda harus mengunggah definisi alur Anda dan mengaktifkan alur Anda. Dalam contoh perintah berikut, ganti *`pipeline_name`* dengan label untuk pipeline Anda dan *`pipeline_file`* dengan jalur yang sepenuhnya memenuhi syarat untuk file definisi `.json` pipeline.

AWS CLI

Untuk membuat definisi alur Anda dan mengaktifkan alur Anda, gunakan perintah create-pipeline. Perhatikan ID alur Anda, karena Anda akan menggunakan nilai ini dengan sebagian besar perintah CLI.

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

Untuk mengunggah definisi pipeline Anda, gunakan put-pipeline-definitionperintah berikut.

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

Jika validasi alur Anda berhasil, bidang `validationErrors` akan kosong. Anda harus meninjau peringatan apa pun.

Untuk mengaktifkan alur Anda, gunakan perintah activate-pipeline.

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

Anda dapat memverifikasi bahwa alur Anda muncul dalam daftar alur menggunakan perintah list-pipelines berikut.

```
aws datapipeline list-pipelines
```

Ekspor Data MySQL ke Amazon S3 Menggunakan AWS Data Pipeline

Tutorial ini memandu Anda melalui proses pembuatan alur data untuk menyalin data (baris) dari tabel di basis data MySQL ke file CSV (nilai yang dipisahkan koma) di bucket Amazon S3 dan kemudian mengirimkan notifikasi Amazon SNS setelah aktivitas penyalinan selesai dengan sukses. Anda akan menggunakan EC2 instance yang disediakan oleh AWS Data Pipeline untuk aktivitas penyalinan ini.

Objek Alur

Alur menggunakan objek berikut:

- CopyActivity
- Ec2Resource
- MySqlDataNode
- S3 DataNode
- SnsAlarm

Daftar Isi

- Sebelum Anda Memulai
- Salin Data MySQL Menggunakan Baris Perintah

Sebelum Anda Memulai

Pastikan Anda telah menyelesaikan langkah-langkah berikut.

- Selesaikan tugas dalam Menyiapkan untuk AWS Data Pipeline.
- (Opsional) Mengatur VPC untuk instans dan grup keamanan untuk VPC.
- Buat sebuah bucket Amazon S3 sebagai output data.

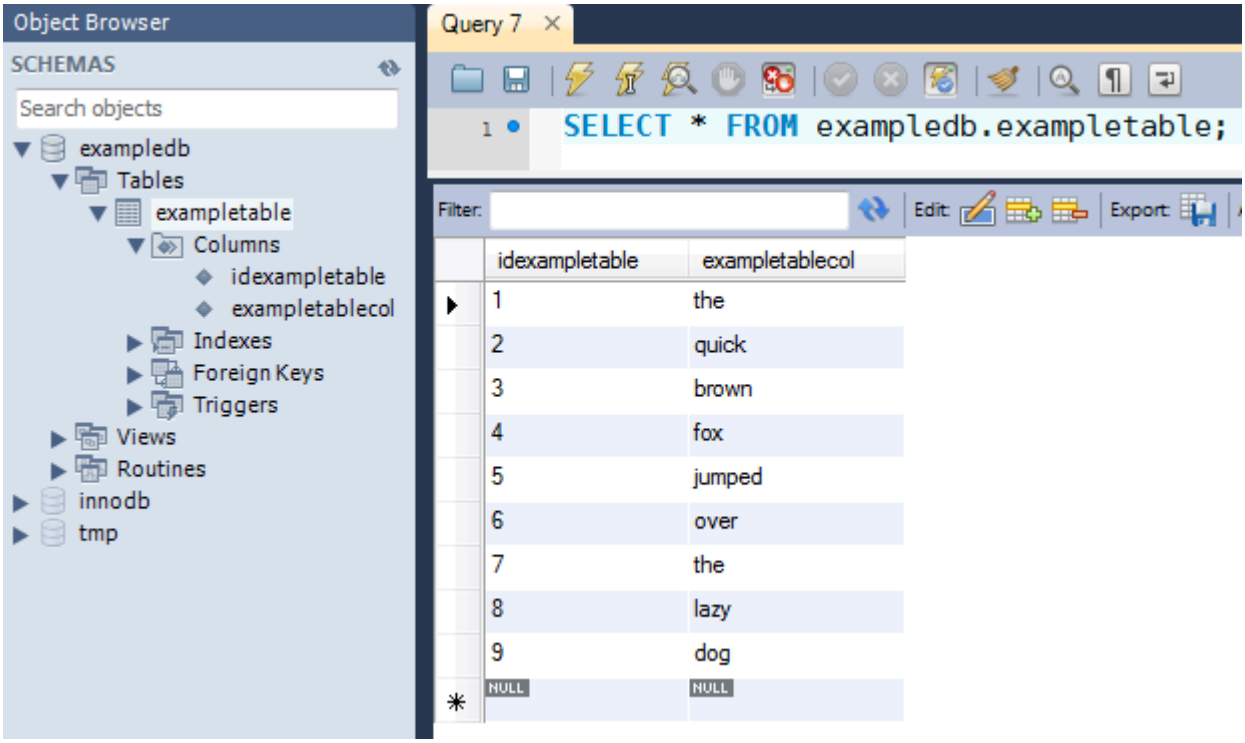
  Untuk informasi selengkapnya, lihat Membuat Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Membuat dan meluncurkan instans basis data MySQL sebagai sumber data Anda.

  Untuk informasi selengkapnya, lihat Meluncurkan Instans DB di Panduan Memulai Amazon RDS. Setelah Anda memiliki instans Amazon RDS, lihat Buat Tabel dalam dokumentasi MySQL.

  > **Note**
  >
  > Buat catatan dari nama pengguna dan kata sandi yang Anda gunakan untuk membuat instans MySQL. Setelah Anda meluncurkan instans basis data MySQL Anda, buat catatan dari titik akhir instans ini. Anda akan memerlukan informasi ini nanti.

- Hubungkan ke instans basis data MySQL Anda, membuat tabel, dan kemudian menambahkan nilai data uji ke tabel yang baru dibuat.

  Untuk tujuan ilustrasi, kami membuat tutorial ini menggunakan tabel MySQL dengan konfigurasi dan sampel data berikut. Screenshot berikut adalah dari MySQL Workbench 5.2 CE:



Untuk informasi selengkapnya, lihat Buat Tabel di dokumentasi MySQL dan Halaman produk MySQL Workbench.

- Membuat topik untuk mengirim notifikasi email dan membuat catatan dari topik Amazon Resource Name (ARN). Untuk informasi lebih lanjut, lihat Buat Topik di Panduan Memulai Amazon Simple Notification Service.
- (Opsional) Tutorial ini menggunakan kebijakan peran IAM default yang dibuat oleh AWS Data Pipeline. Jika Anda lebih suka membuat dan mengonfigurasi kebijakan IAM role dan hubungan kepercayaan, ikuti petunjuk yang dijelaskan di Peran IAM untuk AWS Data Pipeline.

Salin Data MySQL Menggunakan Baris Perintah

Anda dapat membuat alur untuk menyalin data dari tabel MySQL ke file di bucket Amazon S3.

Prasyarat

Sebelum memulai tutorial ini, Anda harus menyelesaikan langkah berikut:

1. Pasang dan konfigurasi antarmuka baris perintah (CLI). Untuk informasi selengkapnya, lihat Mengakses AWS Data Pipeline.

2. Pastikan bahwa peran IAM diberi nama DataPipelineDefaultRoledan DataPipelineDefaultResourceRoleada. AWS Data Pipeline Konsol membuat peran ini untuk Anda secara otomatis. Jika Anda belum pernah menggunakan AWS Data Pipeline konsol setidaknya sekali, maka Anda harus membuat peran ini secara manual. Untuk informasi selengkapnya, lihat Peran IAM untuk AWS Data Pipeline.
3. Mengatur bucket Amazon S3 dan instans Amazon RDS. Untuk informasi selengkapnya, lihat Sebelum Anda Memulai.

Tugas

- Mendefinisikan Alur dalam Format JSON
- Unggah dan Aktifkan Definisi Alur

Mendefinisikan Alur dalam Format JSON

Contoh skenario ini menunjukkan bagaimana menggunakan definisi alur JSON dan CLI AWS Data Pipeline untuk menyalin data (baris) dari tabel dalam basis data MySQL ke file CSV (nilai yang dipisahkan koma) dalam bucket Amazon S3 pada interval waktu yang ditentukan.

Ini adalah file JSON definisi alur lengkap diikuti dengan penjelasan untuk setiap bagiannya.

> **Note**
>
> Kami merekomendasikan bahwa Anda menggunakan editor teks yang dapat membantu Anda memverifikasi sintaks file yang diformat JSON, dan nama file menggunakan ekstensi 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"
    }
  ]
}
```

Simpul Data MySQL

Komponen MySqlDataNode pipa input mendefinisikan lokasi untuk data input; dalam hal ini, instans Amazon RDS. MySqlDataNode Komponen input didefinisikan oleh bidang-bidang berikut:

```
{
  "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

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

nama pengguna

Nama pengguna aku basis data yang memiliki izin yang cukup untuk mengambil data dari tabel basis data. Ganti *my-username* dengan nama pengguna Anda.

Jadwal

Sebuah referensi ke komponen jadwal yang kita buat di baris sebelumnya dari file JSON.

Nama

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

*Kata Sandi

Kata sandi untuk akun database dengan awalan tanda bintang untuk menunjukkan bahwa AWS Data Pipeline harus mengenkripsi nilai kata sandi. Ganti *my-password* dengan kata sandi yang benar untuk pengguna Anda. Bidang kata sandi didahului oleh karakter khusus tanda bintang. Untuk informasi selengkapnya, lihat Karakter khusus.

Tabel

Nama tabel basis data yang mengandung data untuk disalin. Ganti *table-name* dengan nama tabel database Anda.

connectionString

String koneksi JDBC untuk CopyActivity objek untuk terhubung ke database.

selectQuery

Sebuah query SQL SELECT valid yang menentukan data untuk menyalin dari tabel basis data. Perhatikan bahwa `#{table}` adalah ekspresi yang kembali menggunakan nama tabel yang disediakan oleh variabel “tabel” di baris sebelumnya dari file JSON.

Tipe

SqlDataNode Tipe, yang merupakan instance Amazon RDS menggunakan MySQL dalam contoh ini.

> **Note**
>
> MySqlDataNode Tipe ini tidak digunakan lagi. Meskipun Anda masih dapat menggunakan MySqlDataNode, kami sarankan untuk menggunakan SqlDataNode.

Simpul Data Amazon S3

Selanjutnya, komponen pipa input S3Output mendefinisikan lokasi untuk file output; dalam hal ini, file CSV di lokasi bucket Amazon S3. DataNode Komponen output S3 didefinisikan oleh bidang-bidang berikut:

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

Id

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

Jadwal

Sebuah referensi ke komponen jadwal yang kita buat di baris sebelumnya dari file JSON.

filePath

Jalur ke data yang terkait dengan simpul data, yang merupakan file output CSV dalam contoh ini.

Nama

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

Tipe

Jenis objek pipeline, yaitu S3 DataNode agar sesuai dengan lokasi tempat data berada, di bucket Amazon S3.

Sumber daya

Ini adalah definisi sumber daya komputasi yang melakukan operasi penyalinan. Dalam contoh ini, AWS Data Pipeline harus secara otomatis membuat sebuah EC2 instance untuk melakukan tugas copy dan mengakhiri sumber daya setelah tugas selesai. Bidang yang didefinisikan di sini mengontrol pembuatan dan fungsi EC2 instance yang melakukan pekerjaan. Sumber EC2 daya didefinisikan oleh bidang-bidang berikut:

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

Id

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

Jadwal

Jadwal untuk membuat sumber daya komputasi ini.

Nama

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

Peran

IAM role akun yang mengakses sumber daya, seperti mengakses bucket Amazon S3 untuk mengambil data.

Tipe

Jenis sumber daya komputasi untuk melakukan pekerjaan; dalam hal ini, sebuah EC2 instance. Ada jenis sumber daya lain yang tersedia, seperti EmrCluster tipe.

resourceRole

Peran IAM dari akun yang menciptakan sumber daya, seperti membuat dan mengonfigurasi EC2 instance atas nama Anda. Peran dan ResourceRole dapat menjadi peran yang sama, tetapi secara terpisah memberikan perincian yang lebih besar dalam konfigurasi keamanan Anda.

Aktifitas

Bagian terakhir dalam file JSON yang merupakan definisi dari aktivitas yang mewakili pekerjaan yang akan dilakukan. Dalam hal ini kami menggunakan CopyActivity komponen untuk menyalin data dari file di bucket Amazon S3 ke file lain. CopyActivity Komponen didefinisikan oleh bidang-bidang berikut:

```
{
  "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

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja

Input

Lokasi data MySQL yang akan disalin

Jadwal

Jadwal untuk menjalankan kegiatan ini

Nama

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja

runsOn

Sumber daya komputasi yang melakukan pekerjaan yang ditentukan oleh aktivitas ini. Dalam contoh ini, kami memberikan referensi ke EC2 instance yang didefinisikan sebelumnya. Menggunakan `runs0n` bidang menyebabkan AWS Data Pipeline untuk membuat EC2 instance untuk Anda. Bidang `runs0n` menunjukkan bahwa sumber daya yang ada di infrastruktur AWS, sedangkan nilai workerGroup tersebut menunjukkan bahwa Anda ingin menggunakan sumber daya lokal Anda sendiri untuk melakukan pekerjaan.

onSuccess

SnsAlarm untuk mengirim jika aktivitas selesai dengan sukses

onFail

SnsAlarm untuk mengirim jika aktivitas gagal

Output

Lokasi Amazon S3 dari file output CSV

Tipe

Jenis aktivitas yang harus dilakukan.

Unggah dan Aktifkan Definisi Alur

Anda harus mengunggah definisi alur Anda dan mengaktifkan alur Anda. Dalam contoh perintah berikut, ganti *pipeline_name* dengan label untuk pipeline Anda dan *pipeline_file* dengan jalur yang sepenuhnya memenuhi syarat untuk file definisi `.json` pipeline.

AWS CLI

Untuk membuat definisi alur Anda dan mengaktifkan alur Anda, gunakan perintah create-pipeline. Perhatikan ID alur Anda, karena Anda akan menggunakan nilai ini dengan sebagian besar perintah CLI.

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

Untuk mengunggah definisi pipeline Anda, gunakan put-pipeline-definitionperintah berikut.

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

Jika validasi alur Anda berhasil, bidang `validationErrors` akan kosong. Anda harus meninjau peringatan apa pun.

Untuk mengaktifkan alur Anda, gunakan perintah activate-pipeline.

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

Anda dapat memverifikasi bahwa alur Anda muncul dalam daftar alur menggunakan perintah list-pipelines berikut.

```
aws datapipeline list-pipelines
```

Salin Data ke Amazon Redshift Menggunakan AWS Data Pipeline

Tutorial ini memandu Anda melalui proses pembuatan pipeline yang secara berkala memindahkan data dari Amazon S3 ke Amazon Redshift menggunakan template Copy to Redshift di AWS Data Pipeline konsol, atau file definisi pipeline dengan CLI. AWS Data Pipeline

Amazon S3 adalah layanan web yang memungkinkan Anda untuk menyimpan data di cloud. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon Simple Storage Service.

Amazon Redshift adalah layanan gudang data di cloud. Untuk informasi selengkapnya, lihat Panduan Manajemen Amazon Redshift.

Tutorial ini memiliki beberapa prasyarat. Setelah menyelesaikan langkah-langkah berikut, Anda dapat melanjutkan tutorial menggunakan baik konsol atau CLI.

Daftar Isi

- Sebelum Anda Mulai: Mengonfigurasi Opsi COPY dan Beban Data
- Mengatur Alur, membuat Grup Keamanan, dan membuat Klaster Amazon Redshift
- Salin Data ke Amazon Redshift menggunakan Baris Perintah

Sebelum Anda Mulai: Mengonfigurasi Opsi COPY dan Beban Data

Sebelum menyalin data ke Amazon Redshift di AWS Data Pipeline dalamnya, pastikan Anda:

- Memuat data dari Amazon S3.
- Menyiapkan aktivitas `COPY` di Amazon Redshift.

Setelah Anda memastikan opsi ini bekerja dan berhasil menyelesaikan beban data, transfer opsi ini ke AWS Data Pipeline, untuk melakukan penyalinan di dalamnya.

Untuk pilihan `COPY`, lihat COPY di Panduan Developer Basis Data Amazon Redshift.

Untuk langkah memuat data dari Amazon S3, lihat Memuat data dari Amazon S3 di Panduan Developer Basis Data Amazon Redshift.

Misalnya, perintah SQL berikut di Amazon Redshift membuat tabel baru bernama `LISTING` dan menyalin data sampel dari bucket yang tersedia untuk umum di Amazon S3.

Ganti <iam-role-arn> dan wilayah dengan milik Anda sendiri.

Untuk detail tentang contoh ini, lihat Memuat Data Contoh dari Amazon S3 di Panduan Memulai Amazon Redshift.

```
create table listing(
 listid integer not null distkey,
 sellerid integer not null,
 eventid integer not null,
 dateid smallint not null  sortkey,
 numtickets smallint not null,
 priceperticket decimal(8,2),
 totalprice decimal(8,2),
 listtime timestamp);

copy listing from 's3://awssampledbuswest2/tickit/listings_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region 'us-west-2';
```

Mengatur Alur, membuat Grup Keamanan, dan membuat Klaster Amazon Redshift

Untuk mengatur tutorial

1. Selesaikan tugas dalam Menyiapkan untuk AWS Data Pipeline.
2. Buat grup keamanan.
    a. Buka EC2 konsol Amazon.
    b. Di panel navigasi, klik Grup Keamanan.
    c. Klik Buat Grup Keamanan.
    d. Tentukan nama dan deskripsi untuk grup keamanan.
    e. [EC2-Classic] Pilih No VPC untuk VPC.
    f. [EC2-VPC] Pilih ID VPC Anda untuk VPC.
    g. Klik Buat.
3. [EC2-Classic] Buat grup keamanan klaster Amazon Redshift dan tentukan grup keamanan EC2 Amazon.
    a. Buka konsol Amazon Redshift.

b. Di panel navigasi, klik Grup Keamanan.

c. Klik Buat Grup Keamanan Klaster.

d. Di kotak dialog Buat Grup Keamanan Klaster, tentukan nama untuk grup keamanan dan deskripsi untuk grup keamanan klaster.

e. Klik nama grup keamanan klaster baru.

f. Klik Tambah Jenis Koneksi.

g. Di kotak dialog Tambahkan Jenis Koneksi, pilih Grup EC2 Keamanan dari Jenis Koneksi, pilih grup keamanan yang Anda buat dari Nama Grup EC2 Keamanan, lalu klik Otorisasi.

4. [EC2-VPC] Buat grup keamanan klaster Amazon Redshift dan tentukan grup keamanan VPC.

a. Buka EC2 konsol Amazon.

b. Di panel navigasi, klik Grup Keamanan.

c. Klik Buat Grup Keamanan.

d. Di kotak dialog Buat Grup Keamanan, tentukan nama dan deskripsi untuk grup keamanan dan pilih ID dari VPC untuk VPC.

e. Klik Tambahkan Aturan. Tentukan jenis, protokol, dan rentang port, dan mulailah mengetikkan ID dari grup keamanan di Sumber. Pilih grup keamanan yang Anda buat di langkah kedua.

f. Klik Buat.

5. Berikut ini ringkasan langkah-langkah.

Jika Anda memiliki klaster Amazon Redshift yang sudah ada, buat catatan ID klaster.

Untuk membuat klaster baru dan memuat sampel data, ikuti langkah-langkah dalam Memulai dengan Amazon Redshift. Untuk informasi selengkapnya tentang membuat cluster, lihat Membuat Cluster di Panduan Manajemen Pergeseran Merah Amazon.

a. Buka konsol Amazon Redshift.

b. Klik Luncurkan Klaster .

c. Memberikan detail yang diperlukan untuk klaster Anda, dan kemudian klik Lanjutkan.

d. Sediakan konfigurasi simpul, dan kemudian klik Lanjutkan.

e. Pada halaman untuk informasi konfigurasi tambahan, pilih grup keamanan klaster yang Anda buat, dan kemudian klik Lanjutkan.

f. Meninjau spesifikasi klaster Anda, dan kemudian klik Luncurkan klaster.

Salin Data ke Amazon Redshift menggunakan Baris Perintah

Tutorial ini menunjukkan cara menyalin data dari Amazon S3 ke Amazon Redshift. Anda akan membuat tabel baru di Amazon Redshift, lalu gunakan AWS Data Pipeline untuk mentransfer data ke tabel ini dari bucket Amazon S3 publik, yang berisi data input sampel dalam format CSV. Log disimpan ke bucket Amazon S3 yang Anda miliki.

Amazon S3 adalah layanan web yang memungkinkan Anda untuk menyimpan data di cloud. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon Simple Storage Service. Amazon Redshift adalah layanan gudang data di cloud. Untuk informasi selengkapnya, lihat Panduan Manajemen Amazon Redshift.

Prasyarat

Sebelum memulai tutorial ini, Anda harus menyelesaikan langkah berikut:

1. Pasang dan konfigurasi antarmuka baris perintah (CLI). Untuk informasi selengkapnya, lihat Mengakses AWS Data Pipeline.
2. Pastikan bahwa peran IAM bernama DataPipelineDefaultRoledan DataPipelineDefaultResourceRoleada. AWS Data Pipeline Konsol membuat peran ini untuk Anda secara otomatis. Jika Anda belum pernah menggunakan AWS Data Pipeline konsol setidaknya sekali, maka Anda harus membuat peran ini secara manual. Untuk informasi selengkapnya, lihat Peran IAM untuk AWS Data Pipeline.
3. Siapkan perintah `COPY` di Amazon Redshift, karena Anda akan perlu memiliki opsi yang sama bekerja ketika Anda melakukan penyalinan dalam AWS Data Pipeline. Untuk informasi, lihat Sebelum Anda Mulai: Mengonfigurasi Opsi COPY dan Beban Data.
4. Mengatur basis data Amazon Redshift. Untuk informasi selengkapnya, lihat Mengatur Alur, membuat Grup Keamanan, dan membuat Klaster Amazon Redshift.

Tugas

- Definisikan Alur di Format JSON
- Unggah dan Aktifkan Definisi Alur

Definisikan Alur di Format JSON

Contoh skenario ini menunjukkan cara menyalin data dari bucket Amazon S3 ke Amazon Redshift.

Ini adalah file JSON definisi alur lengkap diikuti dengan penjelasan untuk setiap bagiannya. Kami merekomendasikan bahwa Anda menggunakan editor teks yang dapat membantu Anda memverifikasi sintaks file yang diformat JSON, dan nama file menggunakan ekstensi 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"
      }
    }
  ]
}
```

Untuk informasi lebih lanjut tentang objek ini, lihat dokumentasi berikut.

Objek

- Simpul Data
- Sumber daya
- Aktifitas

Simpul Data

Contoh ini menggunakan simpul data input, simpul data output, dan basis data.

Simpul Data Input

Komponen alur `S3DataNode` input mendefinisikan lokasi input data di Amazon S3 dan format data dari input data. Untuk informasi selengkapnya, lihat S3 DataNode.

Komponen input ini didefinisikan oleh bidang berikut:

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

`id`

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`schedule`

Sebuah referensi untuk komponen jadwal.

`filePath`

Jalur ke data yang terkait dengan simpul data, yang merupakan file input CSV dalam contoh ini.

`name`

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`dataFormat`

Sebuah referensi ke format data untuk aktivitas untuk memproses.

Simpul Data Output

Komponen alur `RedshiftDataNode` output mendefinisikan lokasi untuk data output; dalam hal ini, tabel dalam basis data Amazon Redshift. Untuk informasi selengkapnya, lihat RedshiftDataNode. Komponen output ini didefinisikan oleh bidang-bidang berikut:

```
{
  "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`

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`schedule`

Sebuah referensi untuk komponen jadwal.

`tableName`

Nama tabel Amazon Redshift.

`name`

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`createTableSql`

Ekspresi SQL untuk membuat tabel di basis data.

`database`

Sebuah referensi ke basis data Amazon Redshift.

Basis Data

Komponen `RedshiftDatabase` didefinisikan oleh bidang berikut. Untuk informasi selengkapnya, lihat RedshiftDatabase.

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

`id`

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`databaseName`

Nama basis data logis.

`username`

Nama pengguna untuk terhubung ke basis data.

`name`

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`password`

Kata sandi untuk terhubung ke basis data.

`clusterId`

ID dari klaster Redshift.

Sumber daya

Ini adalah definisi sumber daya komputasi yang melakukan operasi penyalinan. Dalam contoh ini, AWS Data Pipeline harus secara otomatis membuat sebuah EC2 instance untuk melakukan tugas copy dan mengakhiri instance setelah tugas selesai. Bidang didefinisikan di sini mengontrol pembuatan dan fungsi dari instans yang melakukan pekerjaan. Untuk informasi selengkapnya, lihat Ec2Resource.

`Ec2Resource` didefinisikan oleh bidang berikut:

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

`id`

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`schedule`

Jadwal untuk membuat sumber daya komputasi ini.

`securityGroups`

Grup keamanan untuk digunakan untuk instans di kolam sumber daya.

`name`

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`role`

IAM role akun yang mengakses sumber daya, seperti mengakses bucket Amazon S3 untuk mengambil data.

`logUri`

Jalur tujuan Amazon S3 untuk membuat cadangan log Task Runner dari `Ec2Resource`.

`resourceRole`

Peran IAM dari akun yang menciptakan sumber daya, seperti membuat dan mengonfigurasi EC2 instance atas nama Anda. Peran dan ResourceRole dapat menjadi peran yang sama, tetapi secara terpisah memberikan perincian yang lebih besar dalam konfigurasi keamanan Anda.

Aktifitas

Bagian terakhir dalam file JSON yang merupakan definisi dari aktivitas yang mewakili pekerjaan yang akan dilakukan. Dalam kasus ini, kami menggunakan komponen `RedshiftCopyActivity` untuk menyalin data dari Amazon S3 ke Amazon Redshift. Untuk informasi selengkapnya, lihat RedshiftCopyActivity.

Komponen `RedshiftCopyActivity` didefinisikan oleh bidang berikut:

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

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

`id`

ID yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`input`

Sebuah referensi ke file sumber Amazon S3.

`schedule`

Jadwal untuk menjalankan aktivitas ini.

`insertMode`

Jenis sisipan (`KEEP_EXISTING`, `OVERWRITE_EXISTING`, atau `TRUNCATE`).

`name`

Nama yang ditetapkan pengguna, yang merupakan label untuk referensi Anda saja.

`runsOn`

Sumber daya komputasi yang melakukan pekerjaan yang mendefinisikan aktivitas ini.

`output`

Sebuah referensi ke tabel tujuan Amazon Redshift.

Unggah dan Aktifkan Definisi Alur

Anda harus mengunggah definisi alur Anda dan mengaktifkan alur Anda. Dalam contoh perintah berikut, ganti *`pipeline_name`* dengan label untuk pipeline Anda dan *`pipeline_file`* dengan jalur yang sepenuhnya memenuhi syarat untuk file definisi `.json` pipeline.

AWS CLI

Untuk membuat definisi alur Anda dan mengaktifkan alur Anda, gunakan perintah create-pipeline. Perhatikan ID alur Anda, karena Anda akan menggunakan nilai ini dengan sebagian besar perintah CLI.

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

Untuk mengunggah definisi pipeline Anda, gunakan put-pipeline-definitionperintah berikut.

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

Jika validasi alur Anda berhasil, bidang `validationErrors` akan kosong. Anda harus meninjau peringatan apa pun.

Untuk mengaktifkan alur Anda, gunakan perintah activate-pipeline.

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

Anda dapat memverifikasi bahwa alur Anda muncul dalam daftar alur menggunakan perintah list-pipelines berikut.

```
aws datapipeline list-pipelines
```

Ekspresi dan Fungsi Alur

Bagian ini menjelaskan sintaks untuk menggunakan ekspresi dan fungsi dalam alur, termasuk tipe data terkait.

Tipe Data Sederhana

Tipe data berikut dapat ditetapkan sebagai nilai bidang.

Tipe

- DateTime
- Numerik
- Referensi Objek
- Periode
- String

DateTime

AWS Data Pipeline mendukung tanggal dan waktu yang dinyatakan dalam format "YYYY-MM-DDTHH: MM: SS" hanya dalam format. UTC/GMT Contoh berikut menetapkan `startDateTime` bidang `Schedule` objek ke`1/15/2012, 11:59 p.m.`, di UTC/GMT zona waktu.

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

Numerik

AWS Data Pipeline mendukung nilai bilangan bulat dan floating-point.

Referensi Objek

Sebuah objek dalam definisi alur. Ini bisa berupa objek saat ini, nama objek yang didefinisikan di tempat lain dalam alur, atau objek yang mencantumkan objek saat ini di bidang, direferensikan oleh kata kunci `node`. Untuk informasi selengkapnya tentang `node`, lihat Mereferensikan Bidang dan Objek. Untuk informasi selengkapnya tentang tipe objek alur, lihat Referensi Objek Alur.

Periode

Menunjukkan seberapa sering acara terjadwal harus dijalankan. Ini dinyatakan dalam format "N [`years`|`months`|`weeks`|`days`|`hours`|`minutes`]", di mana N adalah nilai bilangan bulat positif.

Jangka waktu minimum adalah 15 menit dan jangka waktu maksimum adalah 3 tahun.

Contoh berikut mengatur bidang `period` objek `Schedule` menjadi 3 jam. Ini menciptakan jadwal yang berjalan setiap tiga jam.

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

String

Nilai string standar. String harus diapit oleh kutipan ganda ("). Anda dapat menggunakan karakter garis miring terbalik (\) untuk keluar dari karakter dalam sebuah string. String multiline tidak didukung.

Contoh berikut menunjukkan contoh nilai string yang valid untuk bidang `id`.

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

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

String juga dapat berisi ekspresi yang mengevaluasi nilai string. Ini dimasukkan ke dalam string, dan dibatasi dengan: "#{" dan "}". Contoh berikut menggunakan ekspresi untuk menyisipkan nama objek saat ini ke dalam jalur.

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

Untuk informasi selengkapnya tentang menggunakan ekspresi, lihat Mereferensikan Bidang dan Objek dan Evaluasi Ekspresi.

Ekspresi

Ekspresi memungkinkan Anda untuk berbagi nilai di seluruh objek terkait. Ekspresi diproses oleh layanan AWS Data Pipeline web saat runtime, memastikan bahwa semua ekspresi diganti dengan nilai ekspresi.

Ekspresi dibatasi oleh: "#{" dan "}". Anda dapat menggunakan ekspresi di objek definisi alur apa pun di mana string itu legal. Jika slot adalah referensi atau salah satu dari jenis ID, NAME, TYPE, SPHERE, nilainya tidak dievaluasi dan digunakan kata demi kata.

Ekspresi berikut memanggil salah satu AWS Data Pipeline fungsi. Untuk informasi selengkapnya, lihat Evaluasi Ekspresi.

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

Mereferensikan Bidang dan Objek

Ekspresi bisa menggunakan bidang objek saat ini di mana ekspresi ada, atau bidang objek lain yang ditautkan oleh referensi.

Format slot terdiri dari waktu pembuatan diikuti oleh waktu pembuatan objek, seperti `@S3BackupLocation_2018-01-31T11:05:33`.

Anda juga dapat mereferensikan ID slot persis yang ditentukan dalam definisi alur, seperti ID slot lokasi backup Amazon S3. Untuk mereferensikan ID slot, gunakan `#{parent.@id}`.

Dalam contoh berikut, bidang `filePath` mereferensikan bidang `id` di objek yang sama untuk membentuk nama file. Nilai `filePath` dievaluasi menjadi "`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"}
}
```

Untuk menggunakan bidang yang ada pada objek lain yang ditautkan oleh referensi, gunakan kata kunci `node`. Kata kunci ini hanya tersedia dengan objek alarm dan prasyarat.

Melanjutkan contoh sebelumnya, ekspresi dalam `SnsAlarm` dapat mereferensi ke rentang tanggal dan waktu dalam `Schedule`, karena `S3DataNode` mereferensikan keduanya.

Secara khusus, bidang `FailureNotify` `message` dapat menggunakan bidang waktu aktif `@scheduledStartTime` dan `@scheduledEndTime` dari `ExampleSchedule`, karena referensi

bidang `ExampleDataNode` `onFail` `FailureNotify` dan bidang referensi `schedule` `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**
>
> Anda dapat membuat alur yang memiliki dependensi, seperti tugas dalam alur Anda yang bergantung pada pekerjaan sistem atau tugas lain. Jika alur Anda memerlukan sumber daya tertentu, tambahkan dependensi tersebut ke alur menggunakan prasyarat yang Anda kaitkan dengan simpul data dan tugas. Ini membuat alur Anda lebih mudah untuk di-debug dan lebih tangguh. Selain itu, pertahankan dependensi Anda dalam satu alur jika memungkinkan, karena pemecahan masalah lintas alur sulit dilakukan.

Ekspresi Terinduk

AWS Data Pipeline memungkinkan Anda untuk membuat nilai sarang untuk membuat ekspresi yang lebih kompleks. Misalnya, untuk melakukan penghitungan waktu (kurangi 30 menit dari `scheduledStartTime`) dan memformat hasilnya untuk digunakan dalam definisi alur, Anda bisa menggunakan ekspresi berikut dalam aktivitas:

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

dan menggunakan node awalan jika ekspresi adalah bagian dari SnsAlarm atau Prasyarat:

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

Daftar

Ekspresi dapat dievaluasi pada daftar dan fungsi pada daftar. Misalnya, asumsikan bahwa daftar didefinisikan seperti berikut: `"myList":["one","two"]`. Jika daftar ini digunakan dalam ekspresi

`#{'this is ' + myList}`, itu akan dievaluasi ke `["this is one", "this is two"]`. Jika Anda memiliki dua daftar, Data Pipeline pada akhirnya akan meratakannya dalam evaluasi mereka. Misalnya, jika `myList1` didefinisikan sebagai `[1,2]` dan `myList2` didefinisikan sebagai `[3,4]` maka ekspresi `[#{myList1}, #{myList2}]` akan dievaluasi menjadi `[1,2,3,4]`.

Ekspresi simpul

AWS Data Pipeline menggunakan `#{node.*}` ekspresi di salah satu `SnsAlarm` atau `PreCondition` untuk referensi balik ke objek induk komponen pipeline. Karena `SnsAlarm` dan `PreCondition` direferensikan dari aktivitas atau sumber daya tanpa referensi balik dari mereka, `node` menyediakan cara untuk mereferensi ke pereferensi. Misalnya, definisi alur berikut menunjukkan bagaimana notifikasi kegagalan dapat menggunakan `node` untuk membuat referensi ke induknya, dalam hal ini `ShellCommandActivity`, dan menyertakan waktu mulai dan berakhir terjadwal induk dalam pesan `SnsAlarm`. scheduledStartTime Referensi pada ShellCommandActivity tidak memerlukan `node` awalan karena scheduledStartTime mengacu pada dirinya sendiri.

> **Note**
>
> Kolom yang diawali dengan tanda AT (@) menunjukkan bahwa kolom tersebut adalah kolom waktu aktif.

```
{
  "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 mendukung referensi transitif untuk bidang yang ditentukan pengguna, tetapi bukan bidang runtime. Referensi transitif adalah referensi antara dua komponen alur yang bergantung pada komponen alur lain sebagai perantara. Contoh berikut menunjukkan referensi ke bidang yang ditentukan pengguna transitif dan referensi ke bidang waktu proses non-transitif, keduanya valid. Untuk informasi selengkapnya, lihat Bidang yang ditentukan pengguna.

```
{
  "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"
}
```

Evaluasi Ekspresi

AWS Data Pipeline menyediakan seperangkat fungsi yang dapat Anda gunakan untuk menghitung nilai bidang. Contoh berikut menggunakan fungsi `makeDate` untuk mengatur bidang `startDateTime` dari objek `Schedule` ke `"2011-05-24T0:00:00"` GMT/UTC.

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

Fungsi Matematika

Fungsi berikut tersedia untuk dikerjakan dengan nilai numerik.

Fungsi	Deskripsi
+	Penambahan. Contoh: `#{1 + 2}` Hasil: `3`
-	Pengurangan. Contoh: `#{1 - 2}` Hasil: `-1`
*	Perkalian. Contoh: `#{1 * 2}` Hasil: `2`
/	Pembagian. Jika Anda membagi dua bilangan bulat, hasilnya terpotong. Contoh: `#{1 / 2}`, Hasil: `0` Contoh: `#{1.0 / 2}`, Hasil: `.5`
^	Eksponen. Contoh: `#{2 ^ 2}` Hasil: `4.0`

Fungsi String

Fungsi berikut tersedia untuk bekerja dengan nilai string.

Fungsi	Deskripsi
+	Rangkaian. Nilai non-string pertama kali dikonversi ke string. Contoh: `#{"hel" + "lo"}` Hasil: `"hello"`

Fungsi Tanggal dan Waktu

Fungsi-fungsi berikut tersedia untuk bekerja dengan DateTime nilai-nilai. Sebagai contoh, nilai dari `myDateTime` adalah `May 24, 2011 @ 5:10 pm GMT`.

> **Note**
>
> date/time AWS Data Pipeline Formatnya adalah Joda Time, yang merupakan pengganti kelas tanggal dan waktu Java. Untuk informasi lebih lanjut, lihat Joda Time - Class DateTimeFormat.

Fungsi	Deskripsi
`int day(DateTime myDateTime)`	Mendapat hari DateTime nilai sebagai bilangan bulat. Contoh: `#{day(myDateTime)}` Hasil: `24`
`int dayOfYear(DateTime myDateTime)`	Mendapat hari tahun DateTime nilai sebagai bilangan bulat. Contoh: `#{dayOfYear(myDateTime)}`

Fungsi	Deskripsi
	Hasil: `144`
`DateTime firstOfMonth(DateTime myDateTime)`	Membuat DateTime objek untuk awal bulan dalam yang ditentukan DateTime. Contoh: `#{firstOf Month(myDateTime)}` Hasil: `"2011-05-01T17:10:00z"`
`String format(DateTime myDateTime,String format)`	Menciptakan objek String yang merupakan hasil dari konversi yang ditentukan DateTime menggunakan string format yang ditentukan. Contoh: `#{format( myDateTime,'YYYY-M M-dd HH:mm:ss z')}` Hasil: `"2011-05-24T17:10:00 UTC"`
`int hour(DateTime myDateTime)`	Mendapat jam dari DateTime nilai sebagai bilangan bulat. Contoh: `#{hour(my DateTime)}` Hasil: `17`

Fungsi	Deskripsi
`DateTime makeDate(int year,int month,int day)`	Membuat DateTime objek, di UTC, dengan tahun, bulan, dan hari yang ditentukan, pada tengah malam. Contoh: `#{makeDate(2011,5,24)}` Hasil: `"2011-05-24T0:00:00z"`
`DateTime makeDateTime(int year,int month,int day,int hour,int minute)`	Membuat DateTime objek, di UTC, dengan tahun, bulan, hari, jam, dan menit yang ditentukan. Contoh: `#{makeDateTime(2011,5,24,14,21)}` Hasil: `"2011-05-24T14:21:00z"`
`DateTime midnight(DateTime myDateTime)`	Menciptakan DateTime objek untuk tengah malam saat ini, relatif terhadap yang ditentukan DateTime. Misalnya, di mana `MyDateTime` adalah `2011-05-25T17:10:00z` , hasilnya adalah sebagai berikut. Contoh: `#{midnight(myDateTime)}` Hasil: `"2011-05-25T0:00:00z"`

Fungsi	Deskripsi
`DateTime minusDays(DateTime myDateTime,int daysToSub)`	Menciptakan DateTime objek yang merupakan hasil dari pengurangan jumlah hari yang ditentukan dari yang ditentuka n. DateTime Contoh: `#{minusDa ys(myDateTime,1)}` Hasil: `"2011-05- 23T17:10:00z"`
`DateTime minusHours(DateTime myDateTime,int hoursToSub)`	Menciptakan DateTime objek yang merupakan hasil dari pengurangan jumlah jam yang ditentukan dari yang ditentuka n. DateTime Contoh: `#{minusHo urs(myDateTime,1)}` Hasil: `"2011-05- 24T16:10:00z"`
`DateTime minusMinutes(DateTime myDateTim e,int minutesToSub)`	Menciptakan DateTime objek yang merupakan hasil dari pengurangan jumlah menit yang ditentukan dari yang ditentukan. DateTime Contoh: `#{minusMi nutes(myDateTime,1 )}` Hasil: `"2011-05- 24T17:09:00z"`

Fungsi	Deskripsi
`DateTime minusMonths(DateTime myDateTime,int monthsToSub)`	Menciptakan DateTime objek yang merupakan hasil dari pengurangan jumlah bulan yang ditentukan dari yang ditentukan. DateTime Contoh: `#{minusMonths(myDateTime,1)}` Hasil: `"2011-04-24T17:10:00z"`
`DateTime minusWeeks(DateTime myDateTime,int weeksToSub)`	Menciptakan DateTime objek yang merupakan hasil pengurangan jumlah minggu yang ditentukan dari yang ditentukan. DateTime Contoh: `#{minusWeeks(myDateTime,1)}` Hasil: `"2011-05-17T17:10:00z"`
`DateTime minusYears(DateTime myDateTime,int yearsToSub)`	Menciptakan DateTime objek yang merupakan hasil pengurangan jumlah tahun yang ditentukan dari yang ditentukan. DateTime Contoh: `#{minusYears(myDateTime,1)}` Hasil: `"2010-05-24T17:10:00z"`

Fungsi	Deskripsi
`int minute(DateTime myDateTime)`	Mendapat menit dari DateTime nilai sebagai bilangan bulat. Contoh: `#{minute(myDateTime)}` Hasil: `10`
`int month(DateTime myDateTime)`	Mendapat bulan dari DateTime nilai sebagai bilangan bulat. Contoh: `#{month(myDateTime)}` Hasil: `5`
`DateTime plusDays(DateTime myDateTime,int daysToAdd)`	Menciptakan DateTime objek yang merupakan hasil dari menambahkan jumlah hari yang ditentukan ke yang ditentukan DateTime. Contoh: `#{plusDays(myDateTime,1)}` Hasil: `"2011-05-25T17:10:00z"`

Fungsi	Deskripsi
`DateTime plusHours(DateTime myDateTime,int hoursToAdd)`	Menciptakan DateTime objek yang merupakan hasil dari menambahkan jumlah jam yang ditentukan ke yang ditentukan DateTime. Contoh: `#{plusHours(myDateTime,1)}` Hasil: `"2011-05-24T18:10:00z"`
`DateTime plusMinutes(DateTime myDateTime,int minutesToAdd)`	Menciptakan DateTime objek yang merupakan hasil dari menambahkan jumlah menit yang ditentukan ke yang ditentukan DateTime. Contoh: `#{plusMinutes(myDateTime,1)}` Hasil: `"2011-05-24 17:11:00z"`
`DateTime plusMonths(DateTime myDateTime,int monthsToAdd)`	Menciptakan DateTime objek yang merupakan hasil dari menambahkan jumlah bulan yang ditentukan ke yang ditentukan DateTime. Contoh: `#{plusMonths(myDateTime,1)}` Hasil: `"2011-06-24T17:10:00z"`

Fungsi	Deskripsi
`DateTime plusWeeks(DateTime myDateTime,int weeksToAdd)`	Menciptakan DateTime objek yang merupakan hasil dari menambahkan jumlah minggu yang ditentukan ke yang ditentukan DateTime. Contoh: `#{plusWeeks(myDateTime,1)}` Hasil: `"2011-05-31T17:10:00z"`
`DateTime plusYears(DateTime myDateTime,int yearsToAdd)`	Menciptakan DateTime objek yang merupakan hasil dari menambahkan jumlah tahun yang ditentukan ke yang ditentukan DateTime. Contoh: `#{plusYears(myDateTime,1)}` Hasil: `"2012-05-24T17:10:00z"`

Fungsi	Deskripsi
`DateTime sunday(DateTime myDateTime)`	Membuat DateTime objek untuk hari Minggu sebelumnya, relatif terhadap yang ditentukan DateTime. Jika yang ditentukan DateTime adalah hari Minggu, hasilnya adalah yang ditentukan DateTime. Contoh: `#{sunday(myDateTime)}` Hasil: `"2011-05-22 17:10:00 UTC"`
`int year(DateTime myDateTime)`	Mendapat tahun DateTime nilai sebagai bilangan bulat. Contoh: `#{year(myDateTime)}` Hasil: `2011`
`DateTime yesterday(DateTime myDateTime)`	Menciptakan DateTime objek untuk hari sebelumnya, relatif terhadap yang ditentukan DateTime. Hasilnya sama dengan minusDays(1). Contoh: `#{yesterday(myDateTime)}` Hasil: `"2011-05-23T17:10:00z"`

Karakter khusus

AWS Data Pipeline menggunakan karakter tertentu yang memiliki arti khusus dalam definisi pipeline, seperti yang ditunjukkan pada tabel berikut.

Karakter khusus	Deskripsi	Contoh
@	Bidang waktu aktif. Karakter ini adalah prefiks nama bidang untuk bidang yang hanya tersedia saat alur berjalan.	@actualStartTime @failureReason @resourceStatus
#	Ekspresi Ekspresi dibatasi oleh: “# {” dan “}” dan isi kawat gigi dievaluasi oleh. AWS Data Pipeline Untuk informasi selengkapnya, lihat Ekspresi.	# {format (myDateTime, 'YYYY-MM-dd hh:mm: ss')} s3://amzn-s3-demo-bucket/#{ id}.csv
*	Bidang terenkripsi. Karakter ini adalah awalan nama bidang untuk menunjukkan bahwa AWS Data Pipeline harus mengenkripsi isi bidang ini dalam perjalanan antara konsol atau CLI dan layanan. AWS Data Pipeline	*kata sandi

Referensi Objek Alur

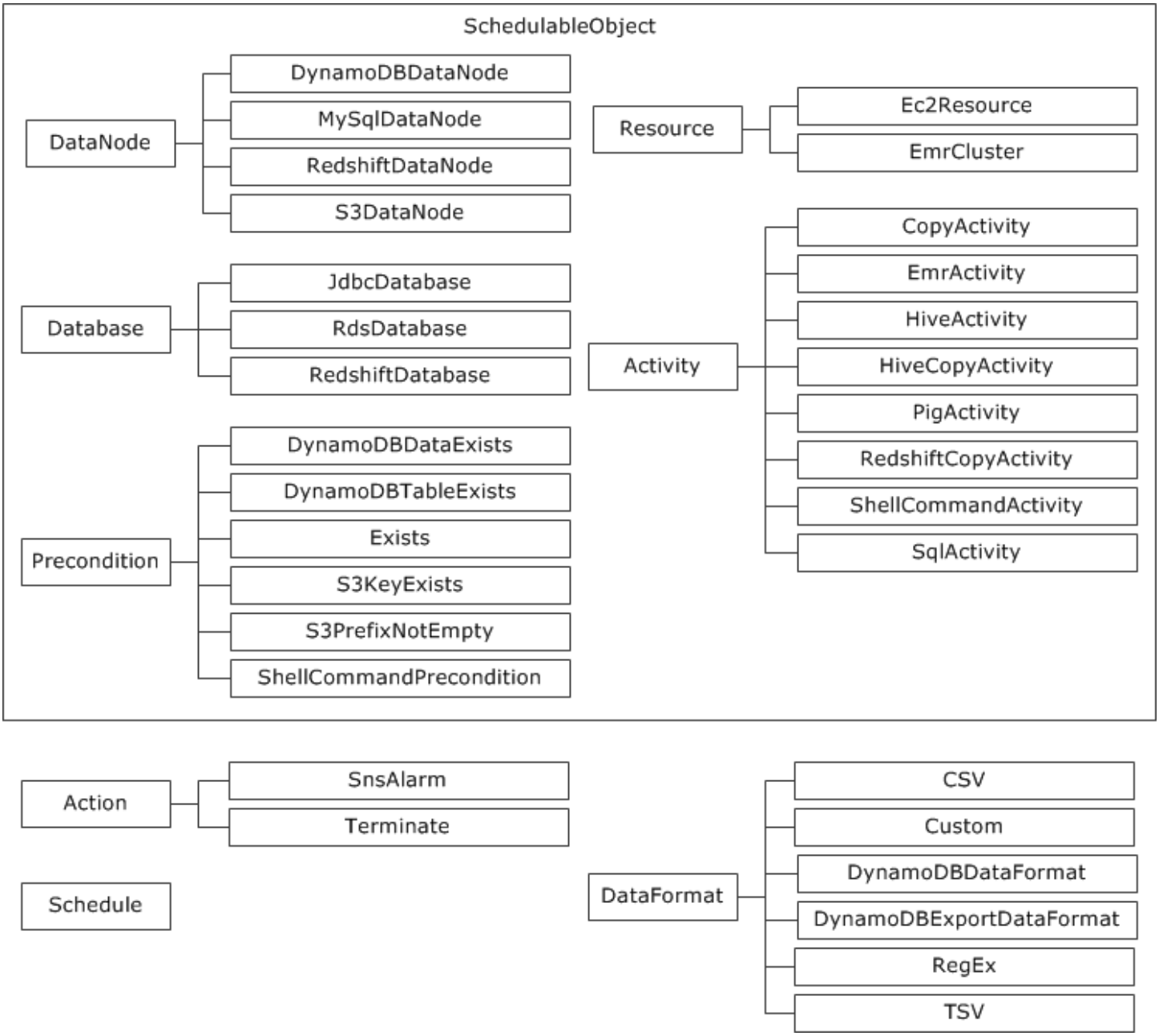
Menjelaskan objek alur dan komponen berikut yang dapat Anda gunakan dalam file definisi alur Anda.

Daftar Isi

- Simpul Data
- Aktivitas
- Sumber daya
- Prasyarat
- Basis Data
- Format Data
- Tindakan
- Jadwal
- Utilitas

> **Note**
>
> Untuk contoh aplikasi yang menggunakan AWS Data Pipeline Java SDK, lihat Data Pipeline DynamoDB Export Java Sample on. GitHub

Berikut ini adalah hirarki objek untuk AWS Data Pipeline.



Simpul Data

Berikut ini adalah objek node AWS Data Pipeline data:

Objek

- Node Dinamo DBData
- MySqlDataNode
- RedshiftDataNode

- S3 DataNode
- SqlDataNode

Node Dinamo DBData

Mendefinisikan simpul data menggunakan DynamoDB, yang ditetapkan sebagai masukan ke objek `HiveActivity` atau `EMRActivity`.

> **Note**
>
> Objek `DynamoDBDataNode` tidak support prasyarat `Exists`.

Contoh

Berikut adalah contoh dari jenis objek ini. Objek ini mereferensikan dua objek lain yang Anda akan definisikan dalam file definisi alur yang sama. `CopyPeriod` adalah objek `Schedule` dan `Ready` adalah objek prasyarat.

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

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
tableName	Tabel DynamoDB.	String

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref": ""“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang contoh konfigurasi jadwal opsional, lihat Jadwal.	Objek Referensi, misalnya, “schedule”: {"ref”:” mScheduleId “}

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika bidang ini disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
dataFormat	DataFormat untuk data yang dijelaskan oleh node data ini. Saat ini didukung untuk HiveActiv ity dan HiveCopyActivity.	Objek Referensi , “DataFormat”: {"ref” :"MyDynamo “} DBData FormatId
dependsOn	Tentukan ketergantungan pada objek lain yang bisa dijalankan	Objek Referensi, misalnya “DependsO

Bidang Opsional	Deskripsi	Jenis Slot
		n”: {"ref”:” “} myActivityId
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}

Bidang Opsional	Deskripsi	Jenis Slot
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyara t”: {"ref”:” “myPrecon ditionId}
readThroughputPerc ent	Mengatur tingkat operasi baca untuk menjaga tingkat throughput yang disediakan DynamoDB Anda berada dalam kisaran dialokasikan untuk tabel Anda. Nilainya adalah dua kali lipat antara 0,1 dan 1,0, secara inklusif.	Ganda
region	Kode untuk wilayah di mana tabel DynamoDB ada. Misalnya, us-east-1. Ini digunakan oleh HiveActivity ketika melakukan pementasan untuk tabel DynamoDB di Hive.	Pencacahan
reportProgressTime out	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporka n kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourc eId

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.	String
writeThroughputPercent	Mengatur tingkat operasi tulis untuk menjaga tingkat throughput yang disediakan DynamoDB Anda berada dalam kisaran yang dialokasikan untuk tabel Anda. Nilainya adalah dua kali lipat antara 0,1 dan 1,0, secara inklusif.	Ganda

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai ketergantungan tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingOn”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

MySqlDataNode

Mendefinisikan simpul data menggunakan MySQL.

> **Note**
>
> Jenis `MySqlDataNode` tidak lagi digunakan. Sebagai gantinya, kami rekomendasikan Anda menggunakan SqlDataNode.

Contoh

Berikut adalah contoh dari jenis objek ini. Objek ini mereferensikan dua objek lain yang Anda akan definisikan dalam file definisi alur yang sama. `CopyPeriod` adalah objek `Schedule` dan `Ready` adalah objek prasyarat.

```
{
  "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" }
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
tabel	Nama tabel dalam basis data MySQL.	String

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref”: "“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapka n dapat dicoba lagi.	Periode
createTableSql	SQL membuat ekspresi tabel yang menciptak an tabel.	String
basis data	Nama basis data.	Objek Referensi, misalnya “database”: {"ref”:” myDatabaseId “}
dependsOn	Menentukan dependensi pada objek lain yang bisa dijalankan.	Objek Referensi, misalnya “DependsO n”: {"ref”:” “} myActivit yId
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
insertQuery	Sebuah pernyataan SQL untuk memasukkan data ke dalam tabel.	String
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat

Bidang Opsional	Deskripsi	Jenis Slot
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyarat”: {"ref”:” “myPreconditionId}
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourc eId
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizink an Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalank annya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan
schemaName	Nama skema yang memegang tabel	String
selectQuery	Sebuah pernyataan SQL untuk mengambil data dari tabel.	String
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingOn”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Lihat Juga

- S3 DataNode

RedshiftDataNode

Mendefinisikan simpul data menggunakan Amazon Redshift. `RedshiftDataNode`emewakili properti data di dalam basis data, seperti tabel data, yang digunakan oleh alur Anda.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "MyRedshiftDataNode",
  "type" : "RedshiftDataNode",
  "database": { "ref": "MyRedshiftDatabase" },
  "tableName": "adEvents",
  "schedule": { "ref": "Hour" }
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
basis data	Basis data tempat tabel berada.	Objek Referensi, misalnya “database ”: {"ref”:” myRedshif tDatabase Id "}
tableName	Nama tabel Amazon Redshift. Tabel dibuat jika belum ada dan Anda telah menyediakannya createTableSql.	String

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref”: "“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapka n dapat dicoba lagi.	Periode
createTableSql	Ekspresi SQL untuk membuat tabel dalam basis data. Kami menyarankan Anda menentukan skema di mana tabel harus dibuat, misalnya: CREATE TABLE MySchema. myTable (bestColumn varchar (25) primary key distkey, integer sortKey). numberOfWins AWS Data Pipeline menjalankan skrip di createTab leSql bidang jika tabel, ditentukan oleh TableName, tidak ada dalam skema, ditentukan oleh bidang schemaName. Misalnya, jika Anda menentukan schemaName sebagai MySchema tetapi tidak menyertakan mySchema di createTableSql bidang, tabel dibuat dalam skema yang salah (secara default, itu akan dibuat di PUBLIC). Hal ini terjadi karena AWS Data Pipeline tidak mengurai pernyataan CREATE TABLE Anda.	String
dependsOn	Tentukan ketergantungan pada objek lain yang bisa dijalankan	Objek Referensi, misalnya “DependsO n”: {"ref”:” “} myActivit yId
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan

Bidang Opsional	Deskripsi	Jenis Slot
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Upaya jumlah maksimum mencoba lagi pada kegagalan.	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyarat”: {"ref”:” “myPreconditionId}

Bidang Opsional	Deskripsi	Jenis Slot
primaryKeys	Jika Anda tidak menentukan primaryKeys untuk tabel tujuan di `RedShiftCopyActivity` , Anda dapat menentukan daftar kolom menggunakan primaryKeys yang akan bertindak sebagai mergeKey. Namun, jika Anda memiliki primaryKey yang ada yang didefinisikan dalam tabel Amazon Redshift, pengaturan ini menimpa kunci yang ada.	String
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourceId

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan
schemaName	Bidang opsional ini menentukan nama skema untuk tabel Amazon Redshift. Jika tidak ditentukan, nama skemanya adalah PUBLIK, yang merupakan skema default di Amazon Redshift. Untuk informasi selengkapnya, lihat Panduan Developer Basis Data Amazon Redshift.	String
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingOn”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

S3 DataNode

Mendefinisikan simpul data menggunakan Amazon S3. Secara default, S3 DataNode menggunakan enkripsi sisi server. Jika Anda ingin menonaktifkan ini, setel s3 EncryptionType ke NONE.

> **Note**
>
> Saat Anda menggunakan `S3DataNode` sebagai masukan ke `CopyActivity`, hanya format data CSV dan TSV yang disupport.

Contoh

Berikut adalah contoh dari jenis objek ini. Objek ini mereferensikan objek lain yang Anda akan definisikan dalam file definisi alur yang sama. `CopyPeriod` adalah objek `Schedule`.

```
{
  "id" : "OutputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://amzn-s3-demo-bucket/#{@scheduledStartTime}.csv"
}
```

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref”: "“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
Kompresi	Jenis kompresi untuk data yang dijelaskan oleh S3DataNode. “none” tidak ada kompresi dan “gzip” dikompresi dengan algoritma	Pencacahan

Bidang Opsional	Deskripsi	Jenis Slot
	gzip. Bidang ini hanya didukung untuk digunakan dengan Amazon Redshift dan saat Anda menggunakan DataNode S3 dengan. CopyActivity	
dataFormat	DataFormat untuk data yang dijelaskan oleh S3 DataNode ini.	Objek Referensi, misalnya “DataForm at”: {"ref”:” myDataFor mat Id "}
dependsOn	Tentukan ketergantungan pada objek lain yang bisa dijalankan	Objek Referensi, misalnya “DependsO n”: {"ref”:” “} myActivit yId
directoryPath	Jalur direktori Amazon S3 sebagai URI: s3:// my-bucket/. my-key-for-directory Anda harus memberikan nilai filePath atau directoryPath.	String
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
filePath	Jalur ke objek di Amazon S3 sebagai URI, misalnya: s3://my-bucket/. my-key-for-file Anda harus memberikan nilai filePath atau directoryPath. Ini mewakili folder dan nama file. Gunakan nilai directoryPath untuk mengakomo dasi beberapa file dalam direktori.	String
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
manifestFilePath	Jalur Amazon S3 ke file manifes dalam format yang didukung oleh Amazon Redshift. AWS Data Pipeline menggunakan file manifes untuk menyalin file Amazon S3 yang ditentukan ke dalam tabel. Bidang ini hanya valid ketika RedShiftCopyActivity referensi S3DataNode.	String
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String

Bidang Opsional	Deskripsi	Jenis Slot
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyara t”: {"ref”:” “myPrecon ditionId}
reportProgressTime out	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporka n kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourc eId
s3 EncryptionType	Mengganti jenis enkripsi Amazon S3. Nilai adalah SERVER_SIDE_ENCRYPTION atau NONE. Enkripsi sisi server diaktifkan secara default.	Pencacahan

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveInstances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromI nstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String

Bidang Runtime	Deskripsi	Jenis Slot
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingOn”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component	String

Bidang Sistem	Deskripsi	Jenis Slot
	Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	

Lihat Juga

- MySqlDataNode

SqlDataNode

Mendefinisikan simpul data menggunakan SQL.

Contoh

Berikut adalah contoh dari jenis objek ini. Objek ini mereferensikan dua objek lain yang Anda akan definisikan dalam file definisi alur yang sama. `CopyPeriod` adalah objek `Schedule` dan `Ready` adalah objek prasyarat.

```
{
  "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" }
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
tabel	Nama tabel dalam basis data SQL.	String

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref": ""}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi, misalnya “schedule”: {"ref”:” mScheduleId “}

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
createTableSql	SQL membuat ekspresi tabel yang menciptakan tabel.	String
basis data	Nama basis data.	Objek Referensi, misalnya “database”:

Bidang Opsional	Deskripsi	Jenis Slot
		{"ref”:” myDatabaseId “}
dependsOn	Menentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, misalnya “DependsO n”: {"ref”:” “} myActivit yId
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
insertQuery	Sebuah pernyataan SQL untuk memasukkan data ke dalam tabel.	String
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAc tion“: {" ref”:” myActionId “}

Bidang Opsional	Deskripsi	Jenis Slot
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSucces s”: {"ref”:” “} myActionI d
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyara t”: {"ref”:” “myPrecon ditionId}
reportProgressTime out	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporka n kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourc eId

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizink an Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalank annya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan
schemaName	Nama skema yang memegang tabel	String
selectQuery	Sebuah pernyataan SQL untuk mengambil data dari tabel.	String
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn

Bidang Runtime	Deskripsi	Jenis Slot
		stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromI nstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTi me	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String

Bidang Sistem	Deskripsi	Jenis Slot
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Lihat Juga

- S3 DataNode

Aktivitas

Berikut ini adalah objek AWS Data Pipeline aktivitas:

Objek

- CopyActivity
- EmrActivity
- HadoopActivity
- HiveActivity
- HiveCopyActivity
- PigActivity
- RedshiftCopyActivity
- ShellCommandActivity
- SqlActivity

CopyActivity

Menyalin data dari satu lokasi ke lokasi lain. `CopyActivity`mendukung S3 DataNode dan SqlDataNode sebagai input dan output dan operasi penyalinan biasanya dilakukan record-by-record. Namun, `CopyActivity` menyediakan performa tinggi Amazon S3 untuk salinan Amazon S3 ketika semua syarat berikut terpenuhi:

- Input dan outputnya adalah S3 DataNodes
- Bidang `dataFormat` adalah sama untuk input dan output

Jika Anda menyediakan file data terkompresi sebagai input dan tidak menunjukkan ini menggunakan bidang `compression` pada simpul data S3, `CopyActivity` mungkin gagal. Dalam kasus ini, `CopyActivity` tidak mendeteksi dengan benar akhir karakter catatan dan operasi gagal. Selanjutnya, `CopyActivity` mendukung penyalinan dari direktori ke direktori lain dan menyalin file ke direktori, tetapi record-by-record salinan terjadi ketika menyalin direktori ke file. Akhirnya, `CopyActivity` tidak men-support penyalinan file Amazon S3 multibagian.

`CopyActivity` memiliki keterbatasan khusus untuk support CSV nya. Saat Anda menggunakan S3 DataNode sebagai masukan`CopyActivity`, Anda hanya dapat menggunakan Unix/Linux varian format file data CSV untuk bidang input dan output Amazon S3. Unix/Linux Varian membutuhkan yang berikut:

- Pemisah harus karakter "," (koma).
- Catatan tidak dikutip.
- Karakter escape default adalah nilai ASCII 92 (backslash).
- Akhir dari pengidentifikasi catatan adalah nilai ASCII 10 (atau "\n").

Sistem berbasis Windows biasanya menggunakan urutan end-of-record karakter yang berbeda: carriage return dan line feed bersama-sama (nilai ASCII 13 dan nilai ASCII 10). Anda harus mengakomodasi perbedaan ini menggunakan mekanisme tambahan, seperti skrip pra-copy untuk memodifikasi input data, untuk memastikan bahwa `CopyActivity` dapat mendeteksi akhir dari sebuah catatan dengan benar; jika tidak, `CopyActivity` akan gagal berulang kali.

Saat menggunakan `CopyActivity` untuk mengekspor dari objek PostgreSQL RDS ke format data TSV, karakter NULL default adalah \n.

Contoh

Berikut adalah contoh dari jenis objek ini. Objek ini mereferensikan tiga objek lain yang akan Anda tetapkan dalam file definisi alur yang sama. `CopyPeriod` adalah objek `Schedule` dan `InputData` dan `OutputData` adalah objek simpul data.

```
{
  "id" : "S3ToS3Copy",
  "type" : "CopyActivity",
```

```
  "schedule" : { "ref" : "CopyPeriod" },
  "input" : { "ref" : "InputData" },
  "output" : { "ref" : "OutputData" },
  "runsOn" : { "ref" : "MyEc2Resource" }
}
```

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref”: "“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”:

Grup yang diperluka n (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
		{"ref":" "} myResourc eId
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapka n dapat dicoba lagi.	Periode
dependsOn	Tentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, misalnya "DependsO n": {"ref":" "} myActivit yId
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
input	Sumber data input.	Objek Referensi , misalnya "input": {"ref":" myDataNode Id "}
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
output	Sumber data output.	Objek Referensi, misalnya “output”: {"ref”:” myDataNodeId "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String

Bidang Opsional	Deskripsi	Jenis Slot
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyara t”: {"ref”:” “myPrecon ditionId}
reportProgressTime out	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporka n kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizink an Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalank annya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai ketergantungan tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingOn”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Lihat Juga

- ShellCommandActivity
- EmrActivity
- Ekspor Data MySQL ke Amazon S3 Menggunakan AWS Data Pipeline

EmrActivity

Menjalankan klaster EMR.

AWS Data Pipeline menggunakan format yang berbeda untuk langkah-langkah dari Amazon EMR; misalnya, AWS Data Pipeline menggunakan argumen yang dipisahkan koma setelah nama JAR di bidang langkah. `EmrActivity` Contoh berikut menunjukkan langkah yang diformat untuk Amazon EMR, diikuti oleh AWS Data Pipeline nya yang setara:

```
s3://amzn-s3-demo-bucket/MyWork.jar arg1 arg2 arg3
```

```
"s3://amzn-s3-demo-bucket/MyWork.jar,arg1,arg2,arg3"
```

Contoh

Berikut adalah contoh dari jenis objek ini. Contoh ini menggunakan versi lama Amazon EMR. Verifikasi contoh ini untuk kebenaran dengan versi klaster Amazon EMR yang Anda gunakan.

Objek ini mereferensikan tiga objek lain yang akan Anda tetapkan dalam file definisi alur yang sama. `MyEmrCluster` adalah objek `EmrCluster` dan `MyS3Input` dan `MyS3Output` adalah objek `S3DataNode`.

> **Note**
>
> Dalam contoh ini, Anda dapat mengganti bidang `step` dengan rangkaian klaster yang Anda inginkan, yang bisa saja script Pig, klaster Hadoop streaming, JAR kustom Anda sendiri termasuk parameternya, atau sebagainya.

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**
>
> Untuk melewati argumen untuk aplikasi dalam langkah, Anda perlu menentukan Wilayah di jalur script, seperti dalam contoh berikut. Selain itu, Anda mungkin perlu melarikan diri dari argumen yang Anda lewati. Misalnya, jika Anda menggunakan `script-runner.jar` untuk menjalankan script dan ingin melewatkan argumen ke script, Anda harus melarikan diri koma yang memisahkan mereka. Slot langkah berikut menggambarkan cara melakukannya:
>
> ```
> "step" : "s3://eu-west-1.elasticmapreduce/libs/script-runner/script-
> runner.jar,s3://datapipeline/echo.sh,a\\\\,b\\\\,c"
> ```
>
> Langkah ini menggunakan `script-runner.jar` untuk menjalankan shell script `echo.sh` dan melewati `a`, `b`, dan `c` sebagai argumen tunggal untuk script. Karakter escape pertama

dihapus dari argumen yang dihasilkan sehingga Anda mungkin perlu untuk melarikan diri lagi. Misalnya, jika Anda memiliki `File\.gz` sebagai sebuah argumen di JSON, anda bisa meloloskan diri menggunakan `File\\\\.gz`. Namun, karena escape pertama dibuang, Anda harus menggunakan `File\\\\\\\\.gz` .

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Tentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi dependens i untuk objek ini. Anda dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan `"schedule": {"ref": "DefaultSchedule"}` . Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), Anda dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi, misalnya, “schedule”: {"ref”:” myScheduleId “}

Grup yang diperluka n (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	Klaster Amazon EMR di mana tugas ini akan berjalan.	Objek Referensi, misalnya, “RunsOn”:

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
		{"ref":" myEmrClusterId "}
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai `runsOn` dan `workerGroup` ada, `workerGroup` akan diabaikan.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
dependsOn	Tentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, misalnya, "DependsOn": {"ref":" "} myActivityId
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
input	Lokasi data input.	Objek Referensi, misalnya, "input": {"ref":" myDataNodeId "}

Bidang Opsional	Deskripsi	Jenis Slot
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum upaya mencoba ulang pada kegagalan.	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya, “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya, "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya, “onSuccess”: {"ref”:” “} myActionId
output	Lokasi data output.	Objek Referensi, misalnya, “output”: {"ref”:” myDataNodeId "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya, “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	URI Amazon S3, seperti 's3://BucketName/Prefix/ 'untuk mengunggah log untuk pipeline.	String

Bidang Opsional	Deskripsi	Jenis Slot
postStepCommand	Shell script untuk dijalankan setelah semua langkah selesai. Untuk menentukan beberapa script, hingga 255, menambahkan beberapa bidang `postStepCommand` .	String
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya, “prasyara t”: {"ref”:” “myPrecon ditionId}
preStepCommand	Shell script untuk dijalankan sebelum langkah-langkah dijalankan. Untuk menentukan beberapa script, hingga 255, menambahkan beberapa bidang `preStepCommand` .	String
reportProgressTime out	Timeout untuk panggilan kerja jarak jauh berturut-turut ke `reportProgress` . Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
resizeClusterBeforeBerlari	Mengubah ukuran klaster sebelum melakukan aktivitas ini untuk mengakomodasi tabel DynamoDB ditentukan sebagai input atau output. **Note** Jika Anda `EmrActivity` menggunakan `DynamoDBDataNode` sebagai node data input atau output, dan jika Anda mengatur `resizeClusterBeforeRunning` keTRUE, AWS Data Pipeline mulai menggunakan tipe `m3.xlarge` instance. Ini akan menimpa pilihan tipe instans Anda dengan `m3.xlarge` , yang dapat menambah biaya bulanan Anda.	Boolean
resizeClusterMaxContoh	Batas pada jumlah maksimum instans yang dapat diminta oleh algoritme resize.	Bilangan Bulat
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval, atau akhir interval. Nilai adalah: `cron`, `ondemand`, dan `timeseries` . Penjadwalan `timeseries` berarti bahwa instans dijadwalkan pada akhir setiap interval. Penjadwalan `cron` berarti bahwa instans dijadwalkan pada awal setiap interval. Jadwal `ondemand` mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal `ondemand` itu harus ditentukan dalam objek default dan harus menjadi satu-satunya `scheduleType` yang ditentukan untuk objek dalam alur. Untuk menggunakan alur `ondemand`, panggil operasi `ActivatePipeline` untuk setiap putaran berikutnya.	Pencacahan
langkah	Satu atau lebih langkah untuk klaster untuk menjalankan. Untuk menentukan beberapa langkah, hingga 255, menambahkan beberapa bidang langkah. Gunakan argumen yang dipisahkan dengan koma setelah nama JAR; misalnya, "`s3://amzn-s3-demo-bucket/MyWork.jar,arg1,arg2,arg3`".	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn

Bidang Runtime	Deskripsi	Jenis Slot
		stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya, "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah Amazon EMR tersedia hanya pada upaya aktivitas EMR	String
errorId	`errorId` jika objek ini gagal.	String
errorMessage	`errorMessage` jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromI nstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTi me	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur tempat objek dibuat.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya, “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String

Bidang Sistem	Deskripsi	Jenis Slot
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Lihat Juga

- ShellCommandActivity
- CopyActivity
- EmrCluster

HadoopActivity

Menjalankan MapReduce pekerjaan di cluster. Cluster dapat berupa cluster EMR yang dikelola oleh AWS Data Pipeline atau sumber daya lain jika Anda menggunakannya. TaskRunner Gunakan HadoopActivity saat Anda ingin menjalankan pekerjaan secara paralel. Ini memungkinkan Anda untuk menggunakan sumber penjadwalan kerangka YARN atau negosiator MapReduce sumber daya di Hadoop 1. Jika Anda ingin menjalankan pekerjaan secara berurutan menggunakan tindakan Langkah Amazon EMR, Anda masih dapat menggunakan EmrActivity.

Contoh

HadoopActivity menggunakan kluster EMR yang dikelola oleh AWS Data Pipeline

HadoopActivity Objek berikut menggunakan EmrCluster sumber daya untuk menjalankan program:

```
{
  "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”
 }
```

Berikut adalah yang sesuai*MyEmrCluster*, yang mengonfigurasi FairScheduler dan antrian di YARN untuk berbasis 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”]
}
```

Ini adalah yang EmrCluster Anda gunakan untuk mengkonfigurasi FairScheduler di Hadoop 1:

```
{
      "id": "MyEmrCluster",
      "type": "EmrCluster",
      "hadoopSchedulerType": "PARALLEL_FAIR_SCHEDULING",
      "amiVersion": "2.4.8",
      "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-m,mapred.queue.names=low\\\\,high\\\\,default,-
m,mapred.fairscheduler.poolnameproperty=mapred.job.queue.name"
```

```
          }
```

Berikut ini EmrCluster mengkonfigurasi CapacityScheduler untuk berbasis 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 menggunakan cluster EMR yang ada

Dalam contoh ini, Anda menggunakan workergroups dan a TaskRunner untuk menjalankan program pada cluster EMR yang ada. Definisi pipeline berikut digunakan HadoopActivity untuk:

- Jalankan MapReduce program hanya pada *myWorkerGroup* sumber daya. Untuk informasi selengkapnya tentang grup pekerja, lihat Menjalankan Pekerjaan pada Sumber Daya yang Ada Menggunakan Runner Tugas.
- Jalankan preActivityTask Config dan 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"
      }
    }
  ]
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
jarUri	Lokasi JAR di Amazon S3 atau sistem file lokal cluster untuk dijalankan. HadoopActivity	String

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref”: "“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Bidang Invokasi Objek	Deskripsi	Jenis Slot
	amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	Klaster EMR di mana tugas ini akan berjalan.	Objek Referensi, misalnya “RunSon”: {"ref”:” myEmrClusterId "}
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.	String

Bidang Opsional	Deskripsi	Jenis Slot
argumen	Argumen untuk diteruskan ke JAR.	String
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
dependsOn	Tentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, misalnya “DependsOn”: {"ref”:” “} myActivityId

Bidang Opsional	Deskripsi	Jenis Slot
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
hadoopQueue	Nama antrean penjadwal Hadoop tempat aktivitas akan dikirimkan.	String
input	Lokasi data input.	Objek Referensi , misalnya “input”: {"ref”:” myDataNode Id "}
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
mainClass	Kelas utama JAR Anda mengeksekusi dengan HadoopActivity.	String
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAc tion“: {" ref”:” myActionId “}

Bidang Opsional	Deskripsi	Jenis Slot
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
output	Lokasi data output.	Objek Referensi, misalnya “output”: {"ref”:” myDataNodeId "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
postActivityTaskConfig	Script konfigurasi post-activity yang akan dijalankan. Ini terdiri dari URI shell script di Amazon S3 dan daftar argumen.	Objek Referensi, misalnya "postActivityTaskConfig”: {"ref”:” “} myShellScript ConfigId
preActivityTaskConfig	Script konfigurasi post-activity yang akan dijalankan. Ini terdiri dari URI shell script di Amazon S3 dan daftar argumen.	Objek Referensi, misalnya "preActivityTaskConfig”: {"ref”:” “} myShellScript ConfigId
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyarat”: {"ref”:” “myPreconditionId}

Bidang Opsional	Deskripsi	Jenis Slot
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn

Bidang Runtime	Deskripsi	Jenis Slot
		stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromI nstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTi me	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String

Bidang Sistem	Deskripsi	Jenis Slot
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Lihat Juga

- ShellCommandActivity
- CopyActivity
- EmrCluster

HiveActivity

Menjalankan kueri Hive pada klaster EMR. `HiveActivity` membuatnya lebih mudah untuk mengatur aktivitas Amazon EMR dan secara otomatis membuat tabel Hive berdasarkan input data yang datang baik dari Amazon S3 atau Amazon RDS. Yang perlu Anda tentukan adalah HiveQL untuk dijalankan pada data sumber. AWS Data Pipeline secara otomatis membuat tabel Hive dengan `${input1}${input2}`,, dan seterusnya, berdasarkan bidang input dalam `HiveActivity` objek.

Untuk input Amazon S3, bidang `dataFormat` digunakan untuk membuat nama kolom Hive.

Untuk input MySQL (Amazon RDS), nama kolom untuk kueri SQL digunakan untuk membuat nama kolom Hive.

> **Note**
>
> Aktivitas ini menggunakan Hive CSV Serde.

Contoh

Berikut adalah contoh dari jenis objek ini. Objek ini mereferensikan tiga objek lain yang akan Anda tetapkan dalam file definisi alur yang sama. `MySchedule` adalah objek `Schedule` dan `MyS3Input` dan `MyS3Output` adalah objek simpul data.

```
{
  "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" }
}
```

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Tentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi dependens i untuk objek ini. Anda dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref”: "“}. DefaultSc hedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), Anda dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional, lihat https://docs.aws.amazon.com/	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Bidang Invokasi Objek	Deskripsi	Jenis Slot
	datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
hiveScript	Script Hive untuk dijalankan.	String
scriptUri	Lokasi script Hive untuk menjalankan (misalnya, s3://scriptLocation).	String

Grup yang Diperlukan	Deskripsi	Jenis Slot
runsOn	Klaster EMR tempat `HiveActivity` ini berjalan.	Objek Referensi, misalnya “RunSon”: {"ref”:” myEmrCluster Id "}
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai `runs0n` dan `workerGroup` ada, `workerGroup` akan diabaikan.	String
input	Sumber data input.	Objek Referensi, seperti “input”: {"ref”:” myDataNode Id "}
output	Sumber data output.	Objek Referensi, seperti “output”: {"ref”:” myDataNode Id "}

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
dependsOn	Tentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, seperti “DependsOn”: {"ref”:” “} myActivityId
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
hadoopQueue	Nama antrean penjadwal Hadoop tempat tugas akan dikirimkan.	String
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum upaya mencoba ulang pada kegagalan.	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi , seperti “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, seperti "onLateAc

Bidang Opsional	Deskripsi	Jenis Slot
		tion“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, seperti “onSuccess”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi , seperti “parent”: {"ref”:” myBaseObject Id "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
postActivityTaskCo nfig	Script konfigurasi post-activity yang akan dijalankan. Ini terdiri dari URI shell script di Amazon S3 dan daftar argumen.	Objek Referensi , seperti "postActi vityTaskConfig”: {"ref”:” “} myShellSc ript ConfigId
preActivityTaskConfig	Script konfigurasi post-activity yang akan dijalankan. Ini terdiri dari URI shell script di Amazon S3 dan daftar argumen.	Objek Referensi , seperti "preActiv ityTaskConfig”: {"ref”:” “} myShellScript ConfigId
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, seperti “prasyarat”: {"ref”:” “myPrecon ditionId}
reportProgressTime out	Timeout untuk panggilan berurutan kerja jarak jauh ke `reportProgress` . Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
resizeClusterBeforeBerlari	Mengubah ukuran klaster sebelum melakukan aktivitas ini untuk mengakomodasi simpul data DynamoDB ditentukan sebagai input atau output. **Note** Jika aktivitas Anda menggunakan node data input atau output, dan jika Anda menyetelnya `resizeClusterBeforeRunning` `TRUE`, AWS Data Pipeline mulailah menggunakan tipe `m3.xlarge` instance. `DynamoDBDataNode` Ini akan menimpa pilihan tipe instans Anda dengan `m3.xlarge` , yang dapat menambah biaya bulanan Anda.	Boolean
resizeClusterMaxContoh	Batas pada jumlah maksimum instans yang dapat diminta oleh algoritme resize.	Bilangan Bulat
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan

Bidang Opsional	Deskripsi	Jenis Slot
scriptVariable	Menentukan variabel script untuk Amazon EMR untuk diteruskan ke Hive saat menjalank an script. Sebagai contoh, variabel script contoh berikut akan melewati SAMPLE dan variabel FILTER_DATE ke Hive :`SAMPLE=s3 ://elasticmapreduce/samples/ hive-ads` dan `FILTER_DATE=#{form at(@scheduledStartTime,'YYYY-MM-dd')}%` . Bidang ini menerima beberapa nilai dan bekerja dengan bidang `script` dan `scriptUri` . Selain itu, fungsi `scriptVar iable` terlepas dari apakah stage diatur ke `true` atau `false`. Bidang ini sangat berguna untuk mengirim nilai-nilai dinamis untuk Hive menggunakan ekspresi dan fungsi AWS Data Pipeline .	String
stage	Menentukan apakah staging diaktifkan sebelum atau setelah menjalankan script. Tidak diizinkan dengan Hive 11, jadi gunakan Amazon EMR AMI versi 3.2.0 atau lebih besar.	Boolean

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi , seperti “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, seperti "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Langkah log Amazon EMR tersedia hanya pada upaya aktivitas EMR.	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromI nstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTi me	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk sebuah objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk sebuah objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, seperti “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Lihat Juga

- ShellCommandActivity
- EmrActivity

HiveCopyActivity

Menjalankan kueri Hive pada klaster EMR. `HiveCopyActivity` membuatnya lebih mudah untuk menyalin data antara tabel DynamoDB. `HiveCopyActivity` menerima pernyataan HiveQL untuk mem-filter data input dari DynamoDB di kolom dan baris tingkat.

Contoh

Contoh berikut menunjukkan cara menggunakan `HiveCopyActivity` dan `DynamoDBExportDataFormat` untuk menyalin data dari satu `DynamoDBDataNode` ke yang lain, sementara mem-filter data, berdasarkan stempel waktu.

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

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi	Objek Referensi, misalnya “schedule”:

Bidang Invokasi Objek	Deskripsi	Jenis Slot
	jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref": "“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	{"ref”:” mySchedule Id “}

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	Tentukan klaster untuk dijalankan.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourc eId
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai `runs0n` dan `workerGroup` ada, `workerGroup` akan diabaikan.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Status yang paling baru dilaporkan dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
dependsOn	Menentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, misalnya “DependsO n”: {"ref”:” “} myActivit yId
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
filterSQL	Sebuah fragmen pernyataan Hive SQL yang mem-filter subset dari data DynamoDB atau Amazon S3 untuk menyalin. Filter seharusnya hanya berisi predikat dan tidak dimulai dengan `WHERE` klausa, karena AWS Data Pipeline menambahkannya secara otomatis.	String
input	Sumber data input. Ini harus menjadi `S3DataNode` atau `DynamoDBDataNode` . Jika Anda menggunakan `DynamoDBNode` , tentukan `DynamoDBExportDataFormat` .	Objek Referensi , misalnya “input”: {"ref”:” myDataNode Id "}
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke `ondemand`.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat

Bidang Opsional	Deskripsi	Jenis Slot
maximumRetries	Upaya jumlah maksimum mencoba lagi pada kegagalan.	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
output	Sumber data output. Jika input adalah `S3DataNode` , ini harus `DynamoDBDataNode` . Jika tidak, ini bisa `S3DataNode` atau `DynamoDBDataNode` . Jika Anda menggunakan `DynamoDBNode` , tentukan `DynamoDBExportDataFormat` .	Objek Referensi, misalnya “output”: {"ref”:” myDataNodeId "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	Amazon S3 URI, seperti `'s3://BucketName/Key/'` , untuk unggah log untuk alur.	String

Bidang Opsional	Deskripsi	Jenis Slot
postActivityTaskConfig	Script konfigurasi post-activity yang akan dijalankan. Ini terdiri dari URI shell script di Amazon S3 dan daftar argumen.	Objek Referensi, misalnya "postActivityTaskConfig”: {"ref”:” “} myShellScript ConfigId
preActivityTaskConfig	Script konfigurasi pre-activity yang akan dijalankan. Ini terdiri dari URI shell script di Amazon S3 dan daftar argumen.	Objek Referensi, misalnya "preActivityTaskConfig”: {"ref”:” “} myShellScript ConfigId
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyarat”: {"ref”:” “myPreconditionId}
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke `reportProgress` . Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
resizeClusterBeforeBerlari	Mengubah ukuran klaster sebelum melakukan aktivitas ini untuk mengakomodasi simpul data DynamoDB ditentukan sebagai input atau output. **Note** Jika aktivitas Anda menggunakan node data input atau output, dan jika Anda menyetelnya `resizeClusterBeforeRunning` `TRUE`, AWS Data Pipeline mulailah menggunakan tipe `m3.xlarge` instance. `DynamoDBDataNode` Ini akan menimpa pilihan tipe instans Anda dengan `m3.xlarge` , yang dapat menambah biaya bulanan Anda.	Boolean
resizeClusterMaxContoh	Batas pada jumlah maksimum instans yang dapat diminta oleh algoritme resize	Bilangan Bulat
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveInstances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Langkah log Amazon EMR tersedia hanya pada upaya aktivitas EMR.	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromI nstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTi me	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Object.	String

Lihat Juga

- ShellCommandActivity
- EmrActivity

PigActivity

PigActivity memberikan dukungan asli untuk skrip Babi AWS Data Pipeline tanpa persyaratan untuk menggunakan `ShellCommandActivity` atau`EmrActivity`. Selain itu, PigActivity mendukung pementasan data. Ketika bidang stage diatur ke BETUL, AWS Data Pipeline men-stage data input sebagai skema di Pig tanpa kode tambahan dari pengguna.

Contoh

Contoh alur berikut menunjukkan cara menggunakan `PigActivity`. Contoh alur melakukan langkah-langkah berikut:

- MyPigActivity1 memuat data dari Amazon S3 dan menjalankan skrip Babi yang memilih beberapa kolom data dan mengunggahnya ke Amazon S3.
- MyPigActivity2 memuat output pertama, memilih beberapa kolom dan tiga baris data, dan mengunggahnya ke Amazon S3 sebagai output kedua.
- MyPigActivity3 memuat data output kedua, menyisipkan dua baris data dan hanya kolom bernama “kelima” ke Amazon RDS.
- MyPigActivity4 memuat data Amazon RDS, memilih baris pertama data, dan mengunggahnya ke 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"
    }
  ]
}
```

Isi dari `pigTestScript.q` adalah sebagai berikut.

```
B = LIMIT ${input1} $three; ${output1} = FOREACH B GENERATE $column1, $column2, Third,
 Fourth, Fifth, Sixth, Seventh, Eighth;
```

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Pengguna harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi ketergantungan untuk objek ini. Pengguna dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan “jadwal”: {"ref”: "“}. DefaultSchedule Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), pengguna dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.	Objek Referensi, misalnya, “schedule”: {"ref”:” myScheduleId “}

Bidang Invokasi Objek	Deskripsi	Jenis Slot
	amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
script	Script Pig yang akan dijalankan.	String
scriptUri	Lokasi script Pig untuk menjalankan (misalnya, s3://scriptLocation).	String

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	EMR Cluster tempat ini PigActivity berjalan.	Objek Referensi, misalnya, “RunsOn”: {"ref”:” myEmrCluster Id "}
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai `runsOn` dan `workerGroup` ada, `workerGroup` akan diabaikan.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Status yang paling baru dilaporkan dari aktivitas jarak jauh.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
dependsOn	Menentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, misalnya, “DependsOn”: {"ref”:” “} myActivityId
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
input	Sumber data input.	Objek Referensi, misalnya, “input”: {"ref”:” myDataNode Id "}
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Upaya jumlah maksimum mencoba lagi pada kegagalan.	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya, “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya, "onLateAction“: {" ref”:” myActionId “}

Bidang Opsional	Deskripsi	Jenis Slot
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya, “onSucces s”: {"ref”:” “} myActionI d
output	Sumber data output.	Objek Referensi, misalnya, “output”: {"ref”:” myDataNode Id "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya, “induk”: {"ref”:” myBaseObject Id "}
pipelineLogUri	URI Amazon S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
postActivityTaskCo nfig	Script konfigurasi post-activity yang akan dijalankan. Ini terdiri dari URI script shell di Amazon S33 dan daftar argumen.	Objek Referensi, misalnya, "postActi vityTaskConfig”: {"ref”:” “} myShellSc ript ConfigId
preActivityTaskConfig	Script konfigurasi post-activity yang akan dijalankan. Ini terdiri dari URI shell script di Amazon S3 dan daftar argumen.	Objek Referensi, misalnya, "preActiv ityTaskConfig”: {"ref”:” “} myShellScript ConfigId
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya, “prasyara t”: {"ref”:” “myPrecon ditionId}

Bidang Opsional	Deskripsi	Jenis Slot
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke `reportProgress` . Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
resizeClusterBeforeBerlari	Mengubah ukuran klaster sebelum melakukan aktivitas ini untuk mengakomodasi simpul data DynamoDB ditentukan sebagai input atau output. **Note** Jika aktivitas Anda menggunakan node data input atau output, dan jika Anda menyetelnya `resizeClusterBeforeRunning` `TRUE`, AWS Data Pipeline mulailah menggunakan tipe `m3.xlarge` instance. `DynamoDBDataNode` Ini akan menimpa pilihan tipe instans Anda dengan `m3.xlarge` , yang dapat menambah biaya bulanan Anda.	Boolean
resizeClusterMaxContoh	Batas pada jumlah maksimum instans yang dapat diminta oleh algoritme resize.	Bilangan Bulat
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Penjadwalan Gaya Deret Waktu berarti instans dijadwalkan pada akhir setiap interval dan Penjadwalan Gaya Cron berarti intans dijadwalkan pada awal setiap interval. Jadwal sesuai permintaan mengizink an Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalank annya lagi. Jika Anda menggunakan jadwal sesuai permintaan itu harus ditentukan dalam objek default dan harus menjadi satu-satunya scheduleType yang ditentukan untuk objek dalam alur. Untuk menggunakan saluran pipa sesuai permintaan, Anda cukup memanggil ActivatePipeline operasi untuk setiap proses berikutnya. Nilai adalah: cron, ondemand, dan timeseries.	Pencacahan
scriptVariable	Argumen untuk diteruskan ke script Pig. Anda dapat menggunakan scriptVariable dengan script atau scriptUri.	String
stage	Menentukan apakah staging diaktifkan dan memungkinkan skrip Pig Anda memiliki akses ke tabel data bertahap, seperti $ {INPUT1} dan $ {}. OUTPUT1	Boolean

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya, “ActiveIn

Bidang Runtime	Deskripsi	Jenis Slot
		stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya, "cascadeFailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Langkah log Amazon EMR tersedia hanya pada upaya aktivitas EMR.	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusinya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTi me	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur tempat objek dibuat.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya, “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String

Bidang Sistem	Deskripsi	Jenis Slot
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Lihat Juga

- ShellCommandActivity
- EmrActivity

RedshiftCopyActivity

Menyalin data dari DynamoDB atau Amazon S3 ke Amazon Redshift. Anda dapat memuat data ke dalam tabel baru, atau dengan mudah menggabungkan data ke dalam tabel yang ada.

Berikut ini adalah gambaran umum kasus penggunaan di mana untuk menggunakan `RedshiftCopyActivity`:

1. Mulailah dengan menggunakan AWS Data Pipeline untuk mementaskan data Anda di Amazon S3.
2. Gunakan `RedshiftCopyActivity` untuk memindahkan data dari Amazon RDS dan Amazon EMR ke Amazon Redshift.

   Hal ini memungkinkan Anda memuat data Anda ke Amazon Redshift di mana Anda dapat menganalisisnya.
3. Gunakan SqlActivity untuk melakukan kueri SQL pada data yang Anda telah muat ke Amazon Redshift.

Selain itu, `RedshiftCopyActivity` memungkinkan Anda bekerja dengan `S3DataNode`, karena men-support file manifes. Untuk informasi selengkapnya, lihat S3 DataNode.

Contoh

Berikut adalah contoh dari jenis objek ini.

Untuk memastikan konversi format, contoh ini menggunakan parameter konversi khusus EMPTYASNULL dan IGNOREBLANKLINES di `commandOptions`. Untuk informasi, lihat Parameter Konversi Data di Panduan Developer Basis Data Amazon Redshift.

```
{
  "id" : "S3ToRedshiftCopyActivity",
  "type" : "RedshiftCopyActivity",
  "input" : { "ref": "MyS3DataNode" },
  "output" : { "ref": "MyRedshiftDataNode" },
  "insertMode" : "KEEP_EXISTING",
  "schedule" : { "ref": "Hour" },
  "runsOn" : { "ref": "MyEc2Resource" },
  "commandOptions": ["EMPTYASNULL", "IGNOREBLANKLINES"]
}
```

Definisi contoh alur berikut menunjukkan aktivitas yang menggunakan mode sisipan 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"
      }
    }
  ]
}
```

Operasi APPEND menambahkan item ke tabel terlepas dari primer atau semacam kunci. Misalnya, jika Anda memiliki tabel berikut, Anda dapat menambahkan catatan dengan ID dan nilai pengguna yang sama.

```
ID(PK)     USER
1          aaa
2          bbb
```

Anda dapat menambahkan catatan dengan ID dan nilai pengguna yang sama:

```
ID(PK)     USER
1          aaa
2          bbb
1          aaa
```

> **Note**
>
> Jika operasi APPEND terganggu dan dicoba lagi, alur jalankan kembali yang dihasilkan berpotensi ditambahkan dari awal. Hal ini dapat menyebabkan duplikasi lebih lanjut, sehingga

Anda harus menyadari perilaku ini, terutama jika Anda memiliki logika yang menghitung jumlah baris.

Untuk tutorial, lihat Salin Data ke Amazon Redshift Menggunakan AWS Data Pipeline.

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
insertMode	Menentukan AWS Data Pipeline apa yang dilakukan dengan data yang sudah ada sebelumnya dalam tabel target yang tumpang tindih dengan baris dalam data yang akan dimuat. Nilai yang valid adalah: `KEEP_EXISTING` , `OVERWRITE_EXISTING` , `TRUNCATE`, dan `APPEND`. `KEEP_EXISTING` menambahkan baris baru ke meja, sementara meninggalkan setiap baris yang ada dimodifikasi. `KEEP_EXISTING` dan `OVERWRITE_EXISTING` menggunakan kunci primer, urutkan, dan kunci distribusi untuk mengident ifikasi baris yang masuk untuk mencocokkan dengan baris yang ada. Lihat Memperbarui dan Memasukkan Data Baru di Amazon Redshift Panduan Developer Basis Data. `TRUNCATE` menghapus semua data dalam tabel tujuan sebelum menulis data baru. `APPEND` menambahkan semua catatan ke akhir tabel Redshift. `APPEND` tidak memerluka n primer, kunci distribusi, atau menyortir kunci	Pencacahan

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
	sehingga item yang mungkin merupakan duplikat potensial dapat ditambahkan.	

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Tentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi dependensi untuk objek ini. Dalam kebanyakan kasus, kami rekomenda sikan untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Misalnya, Anda dapat dengan secara eksplisit mengatur jadwal pada objek dengan menentukan `"schedule": {"ref": "DefaultSchedule"}` . Jika jadwal utama dalam alur Anda berisi jadwal nested, buat objek induk yang memiliki jadwal referensi. Untuk informasi selengkapnya tentang contoh konfigurasi jadwal opsional, lihat Jadwal.	Objek Referensi, seperti: `"schedule":{"ref": "mySchedu leId"}`

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourceId
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai `runsOn` dan `workerGroup` ada, workerGroup akan diabaikan.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
commandOptions	Membawa parameter untuk diteruskan ke simpul data Amazon Redshift selama operasi `COPY`. Untuk informasi tentang parameter, lihat COPY di Panduan Developer Basis Data Amazon Redshift. Saat memuat tabel, `COPY` mencoba untuk secara implisit mengkonversi rangkaian ke tipe data dari kolom target. Selain konversi data default yang terjadi secara otomatis, jika Anda menerima kesalahan atau memiliki kebutuhan konversi lainnya, Anda dapat menentukan	String

Bidang Opsional	Deskripsi	Jenis Slot
	parameter konversi tambahan. Untuk informasi , lihat Parameter Konversi Data di Amazon Redshift Panduan Developer Basis Data. Jika format data dikaitkan dengan input atau output simpul data, maka parameter yang disediakan akan diabaikan. Karena operasi penyalinan pertama kali menggunakan `COPY` untuk memasukkan data ke dalam tabel staging, dan kemudian menggunakan perintah `INSERT` untuk menyalin data dari tabel staging ke tabel tujuan, beberapa parameter `COPY` tidak berlaku, seperti kemampuan perintah `COPY` untuk mengaktifkan kompresi otomatis tabel. Jika kompresi diperlukan, menambahkan detail pengkodean kolom ke pernyataan `CREATE TABLE`. Juga, dalam beberapa kasus ketika perlu membongkar data dari klaster Amazon Redshift dan membuat file di Amazon S3, `RedshiftCopyActivity` bergantung pada operasi `UNLOAD` dari Amazon Redshift. Untuk meningkatkan performa selama penyalinan dan pembongkaran, tentukan parameter `PARALLEL OFF` dari perintah `UNLOAD`. Untuk informasi tentang parameter , lihat UNLOAD di Amazon Redshift Panduan Developer Basis Data.	

Bidang Opsional	Deskripsi	Jenis Slot
dependsOn	Tentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi: `"dependsOn": {"ref":"myActivityId"}`
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
input	Simpul data input. Sumber data bisa jadi Amazon S3, DynamoDB, atau Amazon Redshift.	Objek Referensi: `"input":{"ref":"myDataNodeId"}`
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke `ondemand`.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi: `"onFail": {"ref":"myActionId"}`
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi: `"onLateAction": {"ref":"myActionId"}`

Bidang Opsional	Deskripsi	Jenis Slot
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi: `"onSuccess": {"ref":"myActionId"}`
output	Simpul data output. Lokasi output bisa jadi Amazon S3 atau Amazon Redshift.	Objek Referensi : `"output": {"ref":"myDataNodeId"}`
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi : `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi: `"precondition": {"ref":"myPreconditionId"}`
antrean	Sesuai dengan pengaturan `query_group` di Amazon Redshift, yang mengizinkan Anda untuk menetapkan dan memprioritaskan aktivitas bersamaan berdasarkan penempatan mereka dalam antrean. Amazon Redshift membatasi jumlah koneksi simultan hingga 15. Untuk informasi selengkapnya, lihat Menetapkan Kueri untuk Antrean di Amazon RDS Panduan Developer Basis Data.	String

Bidang Opsional	Deskripsi	Jenis Slot
reportProgressTimeout	Timeout untuk panggilan berurutan kerja jarak jauh ke `reportProgress` . Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
scheduleType	Mengizinkan Anda untuk menentukan apakah jadwal untuk objek dalam alur Anda. Nilai adalah: `cron`, `ondemand`, dan `timeseries` . Penjadwalan `timeseries` berarti bahwa instans dijadwalkan pada akhir setiap interval. Penjadwalan `Cron` berarti bahwa instans dijadwalkan pada awal setiap interval. Jadwal `ondemand` mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Untuk menggunakan alur `ondemand`, panggil operasi `ActivatePipeline` untuk setiap putaran berikutnya. Jika Anda menggunakan jadwal `ondemand`, Anda harus menentukan dalam objek default, dan itu harus menjadi satu-satunya `scheduleType` yang ditentukan untuk objek dalam alur.	Pencacahan

Bidang Opsional	Deskripsi	Jenis Slot
transformSql	Ekspresi `SQL SELECT` yang digunakan untuk mengubah input data. Jalankan ekspresi `transformSql` pada tabel bernama `staging`. Saat Anda menyalin data dari DynamoDB atau Amazon S3, AWS Data Pipeline membuat tabel yang disebut "staging" dan awalnya memuat data di sana. Data dari tabel ini digunakan untuk memperbarui tabel target. Output skema `transformSql` harus sesuai skema tabel target akhir ini. Jika Anda menentukan pilihan `transform Sql` , tabel staging kedua dibuat dari pernyataan SQL tertentu. Data dari tabel staging kedua ini kemudian diperbarui dalam tabel target akhir.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi : `"activeInstances": {"ref":"myRunnableObjectId"}`
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi : `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusinya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi: `"waitingOn": {"ref":"myRunnableObjectId"}`

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup sebuah objek. Menunjukkan tempatnya dalam siklus hidup. Misalnya, Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects..	String

ShellCommandActivity

Menjalankan perintah atau script. Anda dapat menggunakan `ShellCommandActivity` untuk menjalankan tugas terjadwal deret waktu atau seperti cron.

Ketika `stage` bidang disetel ke true dan digunakan dengan`S3DataNode`, `ShellCommandActivity` mendukung konsep pementasan data, yang berarti Anda dapat memindahkan data dari Amazon S3 ke lokasi panggung, seperti EC2 Amazon atau lingkungan lokal Anda, melakukan pekerjaan pada data menggunakan skrip dan, dan memindahkannya `ShellCommandActivity` kembali ke Amazon S3.

Dalam hal ini, ketika perintah shell Anda terhubung ke input `S3DataNode`, script shell anda beroperasi secara langsung pada data menggunakan `${INPUT1_STAGING_DIR}`, `${INPUT2_STAGING_DIR}`, dan bidang lainnya, mengacu pada bidang input `ShellCommandActivity`.

Demikian pula, output dari shell-perintah dapat di-staged dalam direktori output untuk secara otomatis didorong ke Amazon S3, diirujuk oleh `${OUTPUT1_STAGING_DIR}`, `${OUTPUT2_STAGING_DIR}`, dan sebagainya.

Ekspresi ini dapat diteruskan sebagai argumen baris perintah untuk shell-perintah bagi Anda untuk menggunakan dalam logika transformasi data.

`ShellCommandActivity` mengembalikan kode kesalahan bergaya Linux dan rangkaian. Jika hasil `ShellCommandActivity` dalam kesalahan, `error` yang dikembalikan adalah nilai bukan nol.

Contoh

Berikut adalah contoh dari jenis objek ini.

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

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam eksekusi dari selang waktu `schedule`. Untuk menyetel perintah eksekusi dependens i untuk objek ini, tentukan referensi `schedule` ke objek lain. Untuk memenuhi persyaratan ini, atur secara eksplisit `schedule` pada objek, misalnya, dengan menentukan `"schedule": {"ref": "DefaultSchedule"}` . Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi `schedule` pada objek alur default sehingga semua objek mewarisi jadwal itu. Jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), buat objek induk yang memiliki referensi jadwal. Untuk menyebarkan beban, AWS Data Pipeline buat objek fisik sedikit lebih cepat dari jadwal, tetapi jalankan sesuai jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
perintah	Perintah yang akan dijalankan. Gunakan $ untuk referensi parameter posisi dan `scriptArgument` untuk menentukan parameter untuk perintah. Nilai ini dan setiap parameter terkait harus berfungsi di lingkungan dari mana Anda menjalankan Task Runner.	String
scriptUri	Sebuah jalur Amazon S3 URI untuk file untuk mengunduh dan menjalankan sebagai perintah shell. Tentukan hanya satu `scriptUri` , atau bidang `command`. `scriptUri` tidak dapat menggunakan parameter, gunakan `command` sebagai gantinya.	String

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah, misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourceId
workerGroup	Digunakan untuk tugas perutean. Jika Anda memberikan nilai `runs0n` dan `workerGroup` ada, `workerGroup` akan diabaikan.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Status yang paling baru dilaporkan dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
dependsOn	Menentukan dependensi pada objek yang dapat dijalankan lainnya.	Objek Referensi, misalnya “DependsO n”: {"ref”:” “} myActivit yId
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
input	Lokasi data input.	Objek Referensi , misalnya “input”: {"ref”:” myDataNode Id "}
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Upaya jumlah maksimum mencoba lagi pada kegagalan.	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId

Bidang Opsional	Deskripsi	Jenis Slot
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau tidak selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
output	Lokasi data output.	Objek Referensi, misalnya “output”: {"ref”:” myDataNodeId "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	Amazon S3 URI, seperti `'s3://BucketName/Key/'` untuk unggah log untuk alur.	String
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyarat”: {"ref”:” “myPreconditionId}
reportProgressTimeout	Timeout untuk panggilan berturut-turut ke `reportProgress` oleh aktivitas jarak jauh. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
scheduleType	Mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau pada akhir interval. Nilainya adalah `cron`, `ondemand`, dan `timeseries` . Jika disetel ke `timeseries` , instans dijadwalkan pada akhir setiap interval. Jika disetel ke `Cron`, instans dijadwalkan pada awal setiap interval. Jika disetel ke `ondemand`, Anda dapat menjalankan alur satu kali, per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal `ondemand`, tentukan itu dalam objek default sebagai satu-satunya `scheduleType` untuk objek dalam alur. Untuk menggunakan alur `ondemand`, panggil operasi `ActivatePipeline` untuk setiap putaran berikutnya.	Pencacahan

Bidang Opsional	Deskripsi	Jenis Slot
scriptArgument	Array string berformat JSON untuk diteruskan ke perintah yang ditentukan oleh perintah. Misalnya, jika perintah `echo $1 $2`, tentukan `scriptArgument` sebagai `"param1", "param2"`. Untuk beberapa argumen dan parameter, teruskan `scriptArgument` sebagai berikut: `"scriptArgument":"arg1","scriptArgument":"param1","scriptArgument":"arg2","scriptArgument":"param2"`. `scriptArgument` hanya dapat digunakan dengan `command`; Menggunakannya dengan `scriptUri` menyebabkan kesalahan.	String
stage	Menentukan apakah staging diaktifkan dan mengizinkan perintah shell Anda untuk memiliki akses ke variabel data ter-staged, seperti `${INPUT1_STAGING_DIR}` dan `${OUTPUT1_STAGING_DIR}`.	Boolean
stderr	Jalur yang menerima pesan kesalahan sistem yang dialihkan dari perintah. Jika Anda menggunakan bidang `runsOn`, ini harus menjadi jalur Amazon S3 karena sifat sementara dari sumber daya yang menjalankan aktivitas Anda. Namun, jika Anda menentukan bidang `workerGroup`, jalur file lokal diizinkan.	String

Bidang Opsional	Deskripsi	Jenis Slot
stdout	Jalur Amazon S3 yang menerima output yang dialihkan dari perintah. Jika Anda menggunak an bidang `runs0n`, ini harus menjadi jalur Amazon S3 karena sifat sementara dari sumber daya yang menjalankan aktivitas Anda. Namun, jika Anda menentukan bidang `workerGroup` , jalur file lokal diizinkan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	`cancellationReason` jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi yang menyebabk an kegagalan objek.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah Amazon EMR tersedia hanya pada upaya aktivitas Amazon EMR.	String
errorId	`errorId` jika objek ini gagal.	String
errorMessage	`errorMessage` jika objek ini gagal.	String

Bidang Runtime	Deskripsi	Jenis Slot
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu di mana objek menyelesaikan eksekusinya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk aktivitas berbasis Amazon EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu menjalankan terbaru tempat eksekusi selesai.	DateTime
@latestRunTime	Waktu menjalankan terbaru tempat eksekusi dijadwalkan.	DateTime
@nextRunTime	Waktu menjalankan yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwal waktu akhir untuk objek.	DateTime
@scheduledStartTime	Jadwal waktu mulai untuk objek.	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@status	Status objek.	String
@version	AWS Data Pipeline Versi yang digunakan untuk membuat objek.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Kesalahan yang menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Tempat objek dalam siklus hidup. Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Lihat Juga

- CopyActivity
- EmrActivity

SqlActivity

Menjalankan kueri SQL (script) pada basis data.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "MySqlActivity",
  "type" : "SqlActivity",
  "database" : { "ref": "MyDatabaseID" },
  "script" : "SQLQuery" | "scriptUri" : s3://scriptBucket/query.sql,
  "schedule" : { "ref": "MyScheduleID" },
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
basis data	Basis data untuk menjalankan script SQL yang disediakan.	Objek Referensi, misalnya “database”: {"ref”:” myDatabaseId “}

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Anda harus menentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi dependensi untuk objek ini. Misalnya, Anda dapat dengan secara eksplisit mengatur jadwal pada objek dengan menentukan `"schedule": {"ref": "DefaultSchedule"}` . Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Jika alur memiliki pohon jadwal yang bersarang di jadwal utama, buat objek induk yang memiliki	Objek Referensi, misalnya “schedule”: {"ref”:” myScheduleId “}

Bidang Invokasi Objek	Deskripsi	Jenis Slot
	referensi jadwal. Untuk informasi selengkap nya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	

Grup yang diperluka n (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
script	Script SQL untuk menjalankan. Anda harus menentukan script atau scriptUri. Ketika script disimpan di Amazon S3, maka script tidak dievaluasi sebagai ekspresi. Menentukan beberapa nilai untuk scriptArgument sangat membantu ketika script disimpan di Amazon S3.	String
scriptUri	Sebuah URI menentukan lokasi script SQL untuk mengeksekusi dalam aktivitas ini.	String

Grup yang diperluka n (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
runsOn	Sumber daya komputasi untuk menjalankan aktivitas atau perintah. Misalnya, EC2 instans Amazon atau kluster EMR Amazon.	Objek Referensi, misalnya “RunsOn”: {"ref”:” “} myResourc eId
workerGroup	Kelompok pekerja. Ini digunakan untuk tugas perutean. Jika Anda memberikan nilai `runs0n`	String

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
	dan `workerGroup` ada, `workerGroup` akan diabaikan.	

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
dependsOn	Tentukan dependensi pada objek yang bisa dijalankan lainnya.	Objek Referensi, misalnya “DependsOn”: {"ref”:” “} myActivityId
failureAndRerunMode	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
input	Lokasi data input.	Objek Referensi, misalnya “input”: {"ref”:” myDataNodeId "}
lateAfterTimeout	Jangka waktu sejak awal dijadwalkan dari alur di mana objek dijalankan harus dimulai.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat

Bidang Opsional	Deskripsi	Jenis Slot
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika suatu objek belum dijadwalkan atau masih belum selesai dalam periode waktu sejak awal pipeline yang dijadwalkan seperti yang ditentukan oleh 'lateAfterTimeout'.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
output	Lokasi data output. Ini hanya berguna untuk referensi dari dalam skrip (misalnya`#{output.tablename}`) dan untuk membuat tabel output dengan mengatur 'createTableSql' di node data output. Output dari kueri SQL tidak ditulis ke simpul data output.	Objek Referensi, misalnya “output”: {"ref”:” myDataNodeId "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
pipelineLogUri	URI S3 (seperti 's3://BucketName/Key/ ') untuk mengunggah log untuk pipeline.	String

Bidang Opsional	Deskripsi	Jenis Slot
prasyarat	Mendefinisikan prasyarat secara opsional. Sebuah simpul data tidak ditandai "READY" sampai semua prasyarat telah terpenuhi.	Objek Referensi, misalnya “prasyara t”: {"ref”:” “myPrecon ditionId}
antrean	[Amazon Redshift saja] Sesuai dengan pengaturan query_group di Amazon Redshift, yang mengizinkan Anda untuk menetapka n dan memprioritaskan aktivitas bersamaan berdasarkan penempatan mereka dalam antrean. Amazon Redshift membatasi jumlah koneksi simultan hingga 15. Untuk informasi selengkapnya, lihat Menetapkan Kueri untuk Antrean dalam Panduan Developer Basis Data Amazon Redshift.	String
reportProgressTime out	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporka n kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval atau akhir interval. Nilai adalah: `cron`, `ondemand`, dan `timeseries` . Penjadwalan `timeseries` berarti instans dijadwalkan pada akhir setiap interval. Penjadwalan `cron` berarti bahwa instans dijadwalkan pada awal setiap interval. Jadwal `ondemand` mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Ini berarti Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalank annya lagi. Jika Anda menggunakan jadwal `ondemand` itu harus ditentukan dalam objek default dan harus menjadi satu-satunya `scheduleType` yang ditentukan untuk objek dalam alur. Untuk menggunakan alur `ondemand`, panggil operasi `ActivateP ipeline` untuk setiap putaran berikutnya.	Pencacahan
scriptArgument	Daftar variabel untuk script. Sebagai alternatif, Anda dapat menempatkan ekspresi langsung ke bidang script. Beberapa nilai untuk scriptArg ument berguna saat script disimpan di Amazon S3. Contoh: # {format (@scheduledStartTi me, "YY-MM-DD HH:MM:SS"}\n# {format (PlusPeriod (@, “1 hari”)scheduledStartTime, "HH:MM:SS"} YY-MM-DD	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@healthStatusFromInstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingOn”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Sumber daya

Berikut ini adalah objek AWS Data Pipeline sumber daya:

Objek

- Ec2Resource
- EmrCluster
- HttpProxy

Ec2Resource

EC2 Instans Amazon yang melakukan pekerjaan yang ditentukan oleh aktivitas pipeline.

AWS Data Pipeline sekarang mendukung IMDSv2 EC2 instans Amazon, yang menggunakan metode berorientasi sesi untuk menangani otentikasi dengan lebih baik saat mengambil informasi metadata dari instance. Sesi dimulai dan mengakhiri serangkaian permintaan yang digunakan perangkat lunak yang berjalan pada EC2 instans Amazon untuk mengakses metadata dan kredensyal instans EC2 Amazon yang disimpan secara lokal. Perangkat lunak memulai sesi dengan permintaan HTTP PUT sederhana untuk IMDSv2. IMDSv2 mengembalikan token rahasia ke perangkat lunak yang berjalan pada EC2 instance Amazon, yang akan menggunakan token sebagai kata sandi untuk membuat permintaan metadata dan kredensyal. IMDSv2

> **Note**
>
> Untuk digunakan IMDSv2 untuk EC2 instans Amazon Anda, Anda perlu mengubah pengaturan, karena AMI default tidak kompatibel dengan IMDSv2. Anda dapat menentukan versi AMI baru yang dapat Anda ambil melalui parameter SSM berikut:. `/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs`

Untuk informasi tentang EC2 instans Amazon default yang AWS Data Pipeline dibuat jika Anda tidak menentukan instance, lihatEC2 Instans Amazon default menurut Wilayah AWS.

Contoh

EC2-Klasik

> **Important**
>
> Hanya AWS akun yang dibuat sebelum 4 Desember 2013 yang mendukung platform EC2 -Classic. Jika Anda memiliki salah satu akun ini, Anda mungkin memiliki opsi untuk membuat objek EC2 Resource untuk pipeline di jaringan EC2 -Classic daripada VPC. Kami sangat menyarankan agar Anda membuat sumber daya untuk semua saluran pipa Anda. VPCs Selain itu, jika Anda memiliki sumber daya yang ada di EC2 -Classic, kami sarankan Anda memigrasikannya ke VPC.

Contoh objek berikut meluncurkan sebuah EC2 instance ke EC2 -Classic, dengan beberapa bidang opsional ditetapkan.

```
{
  "id" : "MyEC2Resource",
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroups" : [
    "test-group",
    "default"
  ],
  "keyPair" : "my-key-pair"
```

```
}
```

EC2-VPC

Contoh objek berikut meluncurkan sebuah EC2 instance ke dalam VPC nondefault, dengan beberapa bidang opsional ditetapkan.

```
{
  "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"
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
resourceRole	Peran IAM yang mengontrol sumber daya yang dapat diakses EC2 instans Amazon.	String
peran	Peran IAM yang AWS Data Pipeline digunakan untuk membuat EC2 instance.	String

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal.	Objek Referensi, misalnya `"schedule`

Bidang Invokasi Objek	Deskripsi	Jenis Slot
	Untuk mengatur urutan eksekusi dependensi untuk objek ini, tentukan referensi jadwal ke objek lain. Anda dapat melakukannya dengan salah satu cara berikut: • Untuk memastikan bahwa semua objek dalam alur mewarisi jadwal, atur jadwal pada objek secara eksplisit: `"schedule": {"ref": "DefaultSchedule"}` . Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. • Jika alur memiliki pohon jadwal yang bersarang di jadwal utama, Anda dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	`":{"ref": "mySchedu leId"}`

Bidang Opsional	Deskripsi	Jenis Slot
actionOnResourceKegagalan	Tindakan yang diambil setelah kegagalan sumber daya untuk sumber daya ini. Nilai yang valid adalah `"retryall"` dan `"retrynone"` .	String
actionOnTaskKegagalan	Tindakan yang diambil setelah kegagalan tugas untuk sumber daya ini. Nilai-nilai yang valid adalah `"continue"` atau `"terminate"` .	String

Bidang Opsional	Deskripsi	Jenis Slot
associatePublicIpAlamat	Menunjukkan apakah akan menetapkan alamat IP publik pada instans. Jika instancenya ada di Amazon EC2 atau Amazon VPC, nilai defaultnya adalah. `true` Jika tidak, nilai defaultnya adalah `false`.	Boolean
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
availabilityZone	Availability Zone untuk meluncurkan EC2 instans Amazon.	String
menonaktifkan IMDSv1	Nilai default adalah false dan memungkinkan keduanya IMDSv1 dan IMDSv2. Jika Anda mengaturnya ke true maka itu akan dinonaktifkan IMDSv1 dan hanya menyediakan IMDSv2s	Boolean
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
httpProxy	Host proxy yang digunakan klien untuk terhubung ke AWS layanan.	Objek Referensi, misalnya, `"httpProxy": {"ref":"myHttpProxyId"}`

Bidang Opsional	Deskripsi	Jenis Slot
imageId	ID AMI yang digunakan untuk digunakan untuk instans. Secara default, AWS Data Pipeline menggunakan tipe virtualisasi HVM AMI. AMI spesifik yang IDs digunakan didasarka n pada Wilayah. Anda dapat menimpa AMI default dengan menentukan AMI HVM pilihan Anda. Untuk informasi selengkapnya tentang tipe AMI, lihat Jenis Virtualisasi AMI Linux dan Menemukan AMI Linux di Panduan EC2 Pengguna Amazon.	String
initTimeout	Jumlah waktu untuk menunggu sumber daya dimulai.	Periode
instanceCount	Telah usang.	Bilangan Bulat
instanceType	Jenis EC2 instans Amazon untuk memulai.	String
keyPair	Nama pasangan kunci. Jika Anda meluncurk an EC2 instans Amazon tanpa menentukan key pair, Anda tidak dapat masuk ke instans tersebut.	String
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke `ondemand`.	Periode
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Jumlah maksimum upaya mencoba ulang pada kegagalan.	Bilangan Bulat
minInstanceCount	Telah usang.	Bilangan Bulat

Bidang Opsional	Deskripsi	Jenis Slot
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya `"onFail": {"ref":"myActionId"}`
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih berjalan.	Objek Referensi, misalnya`"onLateAction":{"ref":"myActionId"}`
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya, `"onSuccess": {"ref":"myActionId"}`
induk	Induk dari objek saat ini dari mana slot diwariskan.	Objek Referensi, misalnya, `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	Amazon S3 URI (seperti `'s3://BucketName/Key/'`) untuk mengunggah log untuk alur.	String
region	Kode untuk Wilayah tempat EC2 instans Amazon harus dijalankan. Secara default, instans berjalan di Wilayah yang sama dengan alur. Anda dapat menjalankan instans di Wilayah yang sama sebagai set data bergantung.	Pencacahan

Bidang Opsional	Deskripsi	Jenis Slot
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke `reportProgress` . Jika disetel, maka aktivitas jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan akan dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
runAsUser	Pengguna untuk menjalankan TaskRunner.	String
runsOn	Bidang ini tidak diizinkan pada objek ini.	Objek Referensi, misalnya, `"runsOn": {"ref":"myResourceId"}`

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval, atau akhir interval, atau sesuai permintaan. Nilainya adalah: • `timeseries` . Instans dijadwalkan pada akhir setiap interval. • `cron`. Instans dijadwalkan pada awal setiap interval. • `ondemand`. Mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal sesuai permintaan, itu harus ditentukan dalam objek default dan harus menjadi satu-satunya `scheduleType` yang ditentukan untuk objek dalam alur. Untuk menggunakan alur sesuai permintaan, panggil operasi `ActivatePipeline` untuk setiap putaran berikutnya.	Pencacahan
securityGroupIds	IDs Salah satu atau beberapa grup EC2 keamanan Amazon yang akan digunakan untuk instans di kumpulan sumber daya.	String
securityGroups	Satu atau beberapa grup EC2 keamanan Amazon untuk digunakan untuk instans di kumpulan sumber daya.	String
spotBidPrice	Jumlah maksimum per jam untuk Instans Spot Anda dalam dolar, yang merupakan nilai desimal antara 0 dan 20,00, eksklusif.	String

Bidang Opsional	Deskripsi	Jenis Slot
subnetId	ID EC2 subnet Amazon untuk memulai instance.	String
terminateAfter	Jumlah jam setelah itu untuk mengakhiri sumber daya.	Periode
useOnDemandOnLastAttempt	Pada upaya terakhir untuk meminta Instans Spot, buat permintaan untuk Instans Sesuai Permintaan daripada Instans Spot. Hal ini memastikan bahwa jika semua upaya sebelumnya telah gagal, upaya terakhir tidak terganggu.	Boolean
workerGroup	Bidang ini tidak diizinkan pada objek ini.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya, `"activeInstances": {"ref":"myRunnableObjectId"}`
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	`cancellationReason` jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya, `"cascadeFailedOn": {"ref":"m`

Bidang Runtime	Deskripsi	Jenis Slot
		`yRunnable ObjectId"}`
emrStepLog	Log langkah tersedia hanya pada upaya aktivitas Amazon EMR.	String
errorId	ID kesalahan jika objek ini gagal.	String
errorMessage	Pesan galat jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@failureReason	Alasan kegagalan sumber daya.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	Log tugas Hadoop yang tersedia pada upaya untuk aktivitas Amazon EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String
@healthStatusFromI nstanceId	Id dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthSta tusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTi me	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCom pletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwal waktu akhir untuk objek.	DateTime
@scheduledStartTime	Jadwal waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur dengan objek yang dibuat.	String
@waitingOn	Deskripsi daftar dependensi yang menunggu objek ini.	Objek Referensi , misalnya, `"waitingOn": {"ref":"myRunnableObjectId"}`

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Tempat objek dalam siklus hidup. objek komponen memunculkan objek instans, yang mengeksekusi objek percobaan.	String

EmrCluster

Merepresentasikan konfigurasi klaster Amazon EMR. Objek ini digunakan oleh EmrActivity dan HadoopActivity untuk meluncurkan sebuah klaster.

Daftar Isi

- Penjadwal
- Amazon EMR Versi Rilis
- Izin Amazon EMR
- Sintaksis
- Contoh
- Lihat Juga

Penjadwal

Penjadwal menyediakan cara untuk menentukan alokasi sumber daya dan prioritas tugas dalam klaster Hadoop. Administrator atau pengguna dapat memilih penjadwal untuk berbagai kelas pengguna dan aplikasi. Penjadwal bisa menggunakan antrean untuk mengalokasikan sumber daya untuk pengguna dan aplikasi. Anda mengatur antrean tersebut ketika Anda membuat klaster. Anda kemudian dapat mengatur prioritas untuk jenis pekerjaan tertentu dan pengguna atas orang lain. Ini menyediakan untuk efisien penggunaan klaster sumber daya, sementara mengizinkan lebih dari satu pengguna untuk mengirimkan pekerjaan ke klaster. Ada tiga jenis penjadwal yang tersedia:

- FairScheduler— Mencoba menjadwalkan sumber daya secara merata selama periode waktu yang signifikan.
- CapacityScheduler— Menggunakan antrian untuk memungkinkan administrator klaster untuk menetapkan pengguna ke antrian dari berbagai prioritas dan alokasi sumber daya.
- Default — Digunakan oleh klaster, yang dapat dikonfigurasi oleh situs Anda.

Amazon EMR Versi Rilis

Rilis Amazon EMR adalah seperangkat aplikasi sumber terbuka dari ekosistem big data. Setiap rilis terdiri dari aplikasi big data, komponen, dan fitur yang berbeda yang bisa Anda pilih untuk diinstal dan dikonfigurasi oleh Amazon EMR ketika Anda membuat sebuah klaster. Anda menentukan versi rilis menggunakan Label rilis. Label rilis ada dalam bentuk`emr-x.x.x`. Misalnya, `emr-5.30.0`.

Klaster Amazon EMR berdasarkan label rilis `emr-4.0.0` dan kemudian menggunakan properti `releaseLabel` untuk menentukan label rilis dari objek `EmrCluster`. Versi sebelumnya menggunakan properti `amiVersion`.

> **Important**
>
> Semua klaster Amazon EMR dibuat menggunakan rilis versi 5.22.0 atau yang lebih baru menggunakan Tanda Tangan Versi 4 untuk mengautentikasi permintaan ke Amazon S3. Beberapa versi rilis sebelumnya menggunakan Tanda Tangan Versi 2. Support Tanda Tangan versi 2 sedang dihentikan. Untuk informasi selengkapnya, lihat Amazon S3 Update – Sigv2 Periode Pengusangan Diperpanjang dan Dimodifikasi. Kami sangat merekomendasikan agar Anda menggunakan versi rilis Amazon EMR yang men-support Tanda Tangan Versi 4. Untuk rilis versi sebelumnya, dimulai dengan EMR 4.7.x, rilis terbaru dalam seri telah diperbarui untuk men-support Tanda Tangan Versi 4. Saat menggunakan versi rilis Amazon EMR sebelumnya, kami merekomendasikan Anda menggunakan rilis terbaru dalam seri. Selain itu, hindari rilis lebih awal dari EMR 4.7.0.

Pertimbangan dan Batasan

Gunakan versi terbaru Task Runner

Jika Anda menggunakan objek `EmrCluster` yang dikelola sendiri dengan label rilis, gunakan Task Runner terbaru. Untuk informasi selengkapnya tentang Task Runner, lihat Bekerja dengan Runner Tugas. Anda dapat mengonfigurasi nilai properti untuk semua klasifikasi konfigurasi Amazon EMR. Untuk informasi selengkapnya, lihat Mengonfigurasi Aplikasi di Panduan Rilis Amazon EMR, the section called “EmrConfiguration”, dan referensi objek the section called “Properti”.

Support untuk IMDSv2

Sebelumnya, hanya AWS Data Pipeline didukung IMDSv1. Sekarang, AWS Data Pipeline mendukung IMDSv2 di Amazon EMR 5.23.1, 5.27.1, dan 5.32 atau lebih baru, dan Amazon EMR 6.2 atau yang lebih baru. IMDSv2 menggunakan metode berorientasi sesi untuk menangani otentikasi dengan lebih baik saat mengambil informasi metadata dari instance. Anda harus mengonfigurasi instance Anda untuk melakukan IMDSv2 panggilan dengan membuat sumber daya yang dikelola pengguna menggunakan -2.0. TaskRunner

Amazon EMR 5.32 atau yang lebih baru dan Amazon EMR 6.x

Amazon EMR 5.32 atau yang lebih baru dan seri rilis 6.x menggunakan Hadoop versi 3.x, yang memperkenalkan perubahan besar dalam cara classpath Hadoop dievaluasi dibandingkan dengan Hadoop versi 2.x. Perpustakaan umum seperti Joda-Time telah dihapus dari classpath.

Jika EmrActivity atau HadoopActivity menjalankan file Jar yang memiliki dependensi pada perpustakaan yang telah dihapus di Hadoop 3.x, langkah gagal dengan kesalahan `java.lang.NoClassDefFoundError` atau `java.lang.ClassNotFoundException`. Hal ini dapat terjadi untuk file Jar yang berjalan tanpa masalah menggunakan versi rilis Amazon EMR 5.x.

Untuk memperbaiki masalah ini, Anda harus menyalin file Jar dependensi ke classpath Hadoop pada objek `EmrCluster` sebelum memulai `EmrActivity` atau `HadoopActivity`. Kami menyediakan script bash untuk melakukan hal ini. Skrip bash tersedia di lokasi berikut, di mana *`MyRegion`* adalah AWS Wilayah tempat `EmrCluster` objek Anda berjalan, misalnya`us-west-2`.

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

Cara menjalankan skrip tergantung pada apakah `EmrActivity` atau `HadoopActivity` berjalan pada sumber daya yang dikelola oleh AWS Data Pipeline atau dijalankan pada sumber daya yang dikelola sendiri.

Jika Anda menggunakan sumber daya yang dikelola oleh AWS Data Pipeline, tambahkan a `bootstrapAction` ke `EmrCluster` objek. `bootstrapAction` menentukan script dan file Jar untuk menyalin sebagai argumen. Anda dapat menambahkan hingga 255 bidang `bootstrapAction` per objek `EmrCluster`, dan Anda dapat menambahkan bidang `bootstrapAction` ke objek `EmrCluster` yang sudah memiliki tindakan bootstrap.

Untuk menentukan skrip ini sebagai tindakan bootstrap, gunakan sintaks berikut, di mana `JarFileRegion` Wilayah tempat file Jar disimpan, dan masing-masing *`MyJarFilen`* adalah jalur absolut di Amazon S3 dari file Jar yang akan disalin ke classpath Hadoop. Jangan menentukan file Jar yang berada di classpath Hadoop secara default.

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh,JarFileRegion,MyJarFile1,MyJarFile2[, ...]
```

Contoh berikut menentukan tindakan bootstrap yang menyalin dua file Jar di Amazon S3: `my-jar-file.jar` dan `emr-dynamodb-tool-4.14.0-jar-with-dependencies.jar`. Wilayah yang digunakan dalam contoh ini adalah 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"]
}
```

Anda harus menyimpan dan mengaktifkan alur untuk perubahan ke `bootstrapAction` untuk mengambil efek.

Jika Anda menggunakan sumber daya yang dikelola sendiri, Anda dapat mengunduh script untuk instans klaster dan menjalankannya dari baris perintah menggunakan SSH. Script membuat direktori bernama `/etc/hadoop/conf/shellprofile.d` dan sebuah file bernama `datapipeline-jars.sh` dalam direktori itu. File jar disediakan sebagai argumen baris perintah disalin ke direktori yang script ciptakan yang bernama `/home/hadoop/datapipeline_jars`. Jika klaster Anda diatur berbeda, modifikasi script dengan tepat setelah mengunduhnya.

Sintaks untuk menjalankan script pada baris perintah sedikit berbeda dari menggunakan `bootstrapAction` yang ditunjukkan pada contoh sebelumnya. Gunakan spasi bukan koma antara argumen, seperti yang ditunjukkan dalam contoh berikut.

```
./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
```

Izin Amazon EMR

Ketika Anda membuat IAM role kustom, hati-hati mempertimbangkan izin minimum yang diperlukan untuk klaster Anda untuk melakukan pekerjaan. Pastikan untuk memberikan akses ke sumber daya yang diperlukan, seperti file di Amazon S3 atau data di Amazon RDS, Amazon Redshift, atau DynamoDB. Jika Anda ingin mengatur `visibleToAllUsers` ke SALAH, peran Anda harus memiliki izin yang tepat untuk melakukannya. Perhatikan bahwa `DataPipelineDefaultRole` tidak memiliki

izin ini. Anda harus memberikan penyatuan peran `DefaultDataPipelineResourceRole` dan `DataPipelineDefaultRole` sebagai peran objek `EmrCluster`, atau membuat peran Anda sendiri untuk tujuan ini.

Sintaksis

Bidang Invokasi Objek	Deskripsi	Jenis Slot
jadwal	Objek ini dipanggil dalam pelaksanaan interval jadwal. Tentukan referensi jadwal ke objek lain untuk mengatur urutan eksekusi dependens i untuk objek ini. Anda dapat memenuhi persyaratan ini dengan secara eksplisit mengatur jadwal pada objek, misalnya, dengan menentukan `"schedule": {"ref": "DefaultSchedule"}` . Dalam kebanyakan kasus, lebih baik untuk menempatkan referensi jadwal pada objek alur default sehingga semua objek mewarisi jadwal itu. Atau, jika alur memiliki pohon jadwal (jadwal dalam jadwal utama), Anda dapat membuat objek induk yang memiliki referensi jadwal. Untuk informasi selengkapnya tentang konfigurasi jadwal opsional contoh, lihat https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html	Objek Referensi , misalnya, `"schedule ":{"ref": "mySchedu leId"}`

Bidang Opsional	Deskripsi	Jenis Slot
actionOnResourceKegagalan	Tindakan yang diambil setelah kegagalan sumber daya untuk sumber daya ini. Nilai yang benar adalah "`retryall`", yang mencoba semua tugas ke klaster untuk durasi tertentu, dan "`retrynone `".	String

Bidang Opsional	Deskripsi	Jenis Slot
actionOnTaskKegagalan	Tindakan yang diambil setelah kegagalan tugas untuk sumber daya ini. Nilai yang valid adalah "melanjutkan", yang berarti tidak mengakhiri klaster, dan "mengakhiri."	String
additionalMasterSecurityGroupIds	Pengidentifikasi kelompok keamanan master tambahan dari cluster EMR, yang mengikuti formulir sg-01. XXXX6a Untuk informasi selengkapnya, lihat Amazon EMR Grup Keamanan Tambahan di Panduan Manajemen Amazon EMR.	String
additionalSlaveSecurityGroupIds	Pengenal kelompok keamanan budak tambahan dari klaster EMR, yang mengikuti bentuk `sg-01XXXX6a` .	String
amiVersion	Versi Amazon Machine Image (AMI) yang menggunakan Amazon EMR untuk menginstal simpul klaster. Untuk informasi lebih lanjut, lihat Panduan Manajemen Amazon EMR.	String
aplikasi	Aplikasi untuk diinstal di klaster dengan argumen yang dipisahkan koma. Secara default, Hive dan Pig diinstal. Parameter ini hanya berlaku untuk Amazon EMR versi 4.0 dan yang lebih baru.	String
attemptStatus	Status yang paling baru dilaporkan dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
availabilityZone	Availability Zone tempat menjalankan klaster.	String

Bidang Opsional	Deskripsi	Jenis Slot
bootstrapAction	Tindakan untuk dijalankan ketika klaster dimulai. Anda dapat menentukan argumen yang dipisahkan koma. Untuk menentukan beberapa tindakan, hingga 255, menambahk an beberapa bidang `bootstrapAction` . Perilaku defaultnya adalah memulai klaster tanpa tindakan bootstrap.	String
konfigurasi	Konfigurasi untuk klaster Amazon EMR. Parameter ini hanya berlaku untuk Amazon EMR versi 4.0 dan yang lebih baru.	Objek Referensi, misalnya, `"configur ation":{" ref":"myE mrConfigu rationId"}`
coreInstanceBidHarga	Harga Spot maksimum yang bersedia Anda bayarkan untuk EC2 instans Amazon. Jika harga tawaran ditentukan, Amazon EMR menggunakan Instans Spot untuk grup instans. Ditentukan dalam USD.	String
coreInstanceCount	Jumlah simpul inti yang digunakan untuk klaster.	Bilangan Bulat
coreInstanceType	Jenis EC2 instans Amazon yang digunakan untuk node inti. Lihat EC2 Instans Amazon yang Didukung untuk Cluster EMR Amazon.	String
coreGroupConfigura tion	Konfigurasi untuk grup instans inti klaster Amazon EMR. Parameter ini hanya berlaku untuk Amazon EMR versi 4.0 dan yang lebih baru.	Objek Referensi, misalnya `“configur ation”: {“ref”: “myEmrCon figurationId”}`

Bidang Opsional	Deskripsi	Jenis Slot
coreEbsConfiguration	Konfigurasi untuk volume Amazon EBS yang akan melekat pada masing-masing simpul inti dalam kelompok inti di klaster Amazon EMR. Untuk informasi selengkapnya, lihat Jenis Instans yang Mendukung Optimasi EBS di Panduan EC2 Pengguna Amazon.	Objek Referensi, misalnya "coreEbsConfiguration": {"ref": "myEbsConfiguration"}
customAmiId	Berlaku hanya untuk Amazon EMR rilis versi 5.7.0 dan yang lebih baru. Menentukan ID AMI dari AMI kustom yang akan digunakan saat Amazon EMR menyediakan instans Amazon EC2 . Ini juga dapat digunakan sebagai pengganti tindakan bootstrap untuk menyesuaikan konfigurasi node cluster. Untuk informasi selengkapnya, lihat topik berikut di Panduan Manajemen EMR Amazon. Menggunakan AMI kustom	String
EbsBlockDeviceConfig	Konfigurasi perangkat blok Amazon EBS yang diminta terkait dengan grup instans. Termasuk sejumlah volume tertentu yang akan dikaitkan dengan setiap instans dalam grup instans. Termasuk `volumesPerInstance` dan `volumeSpecification` , di mana: • `volumesPerInstance` adalah jumlah volume EBS dengan konfigurasi volume tertentu yang akan dikaitkan dengan setiap instans dalam grup instans. • `volumeSpecification` adalah spesifikasi volume Amazon EBS, seperti tipe volume, IOPS, dan ukuran di Gigibytes (GiB) yang akan diminta untuk volume EBS yang dilampirkan ke instance EC2 di cluster EMR Amazon.	Objek Referensi, misalnya "EbsBlockDeviceConfig": {"ref": "myEbsBlockDeviceConfig"}

Bidang Opsional	Deskripsi	Jenis Slot
emrManagedMasterSecurityGroupId	Pengenal grup keamanan utama tambahan dari klaster Amazon EMR, yang mengikuti bentuk `sg-01XXXX6a` . Untuk informasi selengkapnya, lihat Mengonfigurasi Grup Keamanan dalam Panduan Manajemen Amazon EMR.	String
emrManagedSlaveSecurityGroupId	Pengenal grup keamanan budak tambahan dari klaster Amazon EMR, yang mengikuti bentuk `sg-01XXXX6a` .	String
enableDebugging	Mengaktifkan debugging pada klaster Amazon EMR.	String
failureAndRerunMode	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
hadoopSchedulerType	Jenis penjadwal klaster. Jenis yang valid adalah: `PARALLEL_FAIR_SCHEDULING` , `PARALLEL_CAPACITY_SCHEDULING` , dan `DEFAULT_SCHEDULER` .	Pencacahan
httpProxy	Host proksi yang klien gunakan untuk ter-connect ke layanan AWS.	Objek Referensi, misalnya, “HttpProxy”: {"ref”:” myHttpProxy Id "}
initTimeout	Jumlah waktu untuk menunggu sumber daya dimulai.	Periode
keyPair	Amazon EC2 key pair yang akan digunakan untuk log on ke master node dari Amazon EMR cluster.	String
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke `ondemand`.	Periode

Bidang Opsional	Deskripsi	Jenis Slot
masterInstanceBidHarga	Harga Spot maksimum yang bersedia Anda bayarkan untuk EC2 instans Amazon. Ini adalah nilai desimal antara 0 dan 20,00, eksklusif. Ditentukan dalam USD. Menetapkan nilai ini mengaktifkan Instans Spot untuk simpul utama klaster Amazon EMR. Jika harga tawaran ditentukan, Amazon EMR menggunakan Instans Spot untuk grup instans.	String
masterInstanceType	Jenis EC2 instans Amazon yang digunakan untuk node master. Lihat EC2 Instans Amazon yang Didukung untuk Cluster EMR Amazon.	String
masterGroupConfiguration	Konfigurasi untuk grup instans utama klaster Amazon EMR. Parameter ini hanya berlaku untuk Amazon EMR versi 4.0 dan yang lebih baru.	Objek Referensi, misalnya `"configuration": {"ref": "myEmrConfigurationId"}`
masterEbsConfiguration	Konfigurasi untuk volume Amazon EBS yang akan melekat pada masing-masing simpul utama dalam grup utama di klaster Amazon EMR. Untuk informasi selengkapnya, lihat Jenis Instans yang Mendukung Optimasi EBS di Panduan EC2 Pengguna Amazon.	Objek Referensi, misalnya `"masterEbsConfiguration": {"ref": "myEbsConfiguration"}`
maxActiveInstances	Jumlah maksimum instans aktif bersamaan dari suatu komponen. Re-runs tidak dihitung terhadap jumlah instans aktif.	Bilangan Bulat
maximumRetries	Upaya jumlah maksimum mencoba lagi pada kegagalan.	Bilangan Bulat

Bidang Opsional	Deskripsi	Jenis Slot
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya, `"onFail": {"ref":"myActionId"}`
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya, `"onLateAction":{"ref":"myActionId"}`
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya, `"onSuccess":{"ref":"myActionId"}`
induk	Induk dari objek saat ini dari mana slot diwariskan.	Objek Referensi, misalnya. `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	URI Amazon S3 (seperti 's3://BucketName/Key/') untuk mengunggah log untuk pipeline.	String
region	Kode untuk wilayah tempat klaster Amazon EMR harus berjalan. Secara default, klaster berjalan di Wilayah yang sama dengan alur. Anda dapat menjalankan klaster di Wilayah yang sama sebagai set data bergantung.	Pencacahan
releaseLabel	Label rilis untuk klaster EMR.	String

Bidang Opsional	Deskripsi	Jenis Slot
reportProgressTimeout	Timeout untuk panggilan berurutan kerja jarak jauh ke `reportProgress` . Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
resourceRole	Peran IAM yang AWS Data Pipeline digunakan untuk membuat cluster EMR Amazon. Peran defaultya adalah `DataPipelineDefaultRole` .	String
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
peran	Peran IAM diteruskan ke Amazon EMR untuk membuat EC2 node.	String
runsOn	Bidang ini tidak diizinkan pada objek ini.	Objek Referensi, misalnya, `"runsOn": {"ref":"myResourceId"}`
SecurityConfiguration	Pengidentifikasi konfigurasi keamanan EMR yang akan diterapkan ke cluster. Parameter ini hanya berlaku untuk Amazon EMR versi 4.8.0 dan yang lebih baru.	String
serviceAccessSecurityGroupId	Pengenal grup keamanan akses layanan klaster Amazon EMR.	String. Ini mengikuti bentuk `sg-01XXXX6a` , misalnya, `sg-1234abcd` .

Bidang Opsional	Deskripsi	Jenis Slot
scheduleType	Jenis jadwal mengizinkan Anda untuk menentukan apakah objek dalam definisi alur Anda harus dijadwalkan pada awal interval, atau akhir interval. Nilai adalah: `cron`, `ondemand`, dan `timeseries` . Penjadwalan `timeseries` berarti bahwa instans dijadwalkan pada akhir setiap interval. Penjadwalan `cron` berarti bahwa instans dijadwalkan pada awal setiap interval. Jadwal `ondemand` mengizinkan Anda untuk menjalankan alur satu kali per aktivasi. Anda tidak perlu meng-klon atau membuat ulang alur untuk menjalankannya lagi. Jika Anda menggunakan jadwal `ondemand` itu harus ditentukan dalam objek default dan harus menjadi satu-satunya `scheduleType` yang ditentukan untuk objek dalam alur. Untuk menggunakan alur `ondemand`, panggil operasi `ActivatePipeline` untuk setiap putaran berikutnya.	Pencacahan
subnetId	Pengenal subnet ke tempat untuk meluncurkan klaster Amazon EMR.	String
supportedProducts	Parameter yang menginstal perangkat lunak pihak ke tiga pada klaster Amazon EMR, misalnya, distribusi pihak ke tiga Hadoop.	String
taskInstanceBidHarga	Harga Spot maksimum yang bersedia Anda bayarkan untuk EC2 instans. Nilai desimal antara 0 dan 20,00, eksklusif. Ditentukan dalam USD. Jika harga tawaran ditentukan, Amazon EMR menggunakan Instans Spot untuk grup instans.	String

Bidang Opsional	Deskripsi	Jenis Slot
taskInstanceCount	Jumlah simpul tugas untuk digunakan untuk klaster Amazon EMR.	Bilangan Bulat
taskInstanceType	Jenis EC2 instans Amazon yang digunakan untuk node tugas.	String
taskGroupConfiguration	Konfigurasi untuk grup instans tugas klaster Amazon EMR. Parameter ini hanya berlaku untuk Amazon EMR versi 4.0 dan yang lebih baru.	Objek Referensi, misalnya `“configuration”: {“ref”: “myEmrConfigurationId”}`
taskEbsConfiguration	Konfigurasi untuk volume Amazon EBS yang akan melekat pada masing-masing simpul tugas dalam grup tugas di klaster Amazon EMR. Untuk informasi selengkapnya, lihat Jenis Instans yang Mendukung Optimasi EBS di Panduan EC2 Pengguna Amazon.	Objek Referensi, misalnya `“taskEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
terminateAfter	Mengakhiri sumber daya setelah berjam-jam ini.	Bilangan Bulat

Bidang Opsional	Deskripsi	Jenis Slot
VolumeSpecification	Spesifikasi volume Amazon EBS, seperti tipe volume, IOPS, dan ukuran di Gigibytes (GiB) yang akan diminta untuk volume Amazon EBS yang dilampirkan ke instans EC2 Amazon di cluster EMR Amazon. Simpul bisa menjadi inti, utama atau simpul tugas. `VolumeSpecification` termasuk: • `iops()`Bilangan bulat. Jumlah I/O operasi per detik (IOPS) yang didukung oleh volume Amazon EBS, misalnya, 1000. Untuk informasi selengkapnya, lihat Karakteristik I/O EBS di EC2 Panduan Pengguna Amazon. • `sizeinGB()` . Bilangan bulat. Ukuran volume Amazon EBS, dalam gigibita (GiB), misalnya 500. Untuk informasi tentang kombinasi jenis volume dan ukuran hard drive yang valid, lihat Jenis Volume EBS di Panduan EC2 Pengguna Amazon. • `volumetType` . Rangkaian. Jenis volume Amazon EBS, misalnya, gp2. Jenis volume yang disupport termasuk standar, gp2, io1, st1, sc1, dan lain-lain. Untuk informasi selengkapnya, lihat Jenis Volume EBS di Panduan EC2 Pengguna Amazon.	Objek Referensi, misalnya `“VolumeSpecification”: {“ref”: “myVolumeSpecification”}`
useOnDemandOnLastAttempt	Pada upaya terakhir untuk meminta sumber daya, buat permintaan untuk Instans Sesuai Permintaan daripada Instans Spot. Hal ini memastikan bahwa jika semua upaya sebelumnya telah gagal, upaya terakhir tidak terganggu.	Boolean
workerGroup	Bidang tidak diizinkan pada objek ini.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya, “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya, "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah tersedia hanya pada upaya aktivitas Amazon EMR.	String
errorId	ID kesalahan jika objek ini gagal.	String
errorMessage	Pesan galat jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
@failureReason	Alasan kegagalan sumber daya.	String
@finishedTime	Waktu saat objek ini menyelesaikan eksekusin ya.	DateTime
hadoopJobLog	Log tugas Hadoop yang tersedia pada upaya untuk aktivitas Amazon EMR.	String
@healthStatus	Status kondisi objek yang mencerminkan keberhasilan atau kegagalan instans objek terakhir yang mencapai keadaan dihentikan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@healthStatusFromInstanceId	ID dari objek instans terakhir yang mencapai keadaan dihentikan.	String
@ healthStatusUpdated Waktu	Waktu di mana status kondisi diperbarui terakhir kali.	DateTime
hostname	Nama host klien yang mengambil upaya tugas.	String
@lastDeactivatedTime	Waktu di mana objek ini terakhir dinonaktifkan.	DateTime
@ latestCompletedRun Waktu	Waktu proses terakhir yang eksekusinya selesai.	DateTime
@latestRunTime	Waktu proses terakhir untuk eksekusi yang dijadwalkan.	DateTime
@nextRunTime	Waktu run yang akan dijadwalkan berikutnya.	DateTime
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur dengan objek yang dibuat.	String
@waitingOn	Deskripsi daftar dependensi yang menunggu objek ini.	Objek Referensi, misalnya, “WaitingOn”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Tempat objek dalam siklus hidup. objek komponen memunculkan objek instans, yang mengeksekusi objek percobaan.	String

Contoh

Berikut ini adalah contoh jenis objek ini.

Daftar Isi

- Meluncurkan klaster Amazon EMR dengan hadoopVersion
- Meluncurkan klaster Amazon EMR dengan label rilis emr-4.x atau lebih besar
- Menginstal perangkat lunak tambahan pada klaster Amazon EMR Anda
- Nonaktifkan enkripsi sisi server pada rilis 3.x
- Nonaktifkan enkripsi sisi server pada rilis 4.x
- Konfigurasikan Hadoop KMS ACLs dan buat zona enkripsi di HDFS
- Tentukan IAM role khusus
- Menggunakan EmrCluster Sumber Daya di AWS SDK for Java
- Mengonfigurasi klaster Amazon EMR di subnet privat
- Melampirkan volume EBS ke simpul klaster

Meluncurkan klaster Amazon EMR dengan hadoopVersion

Example

Contoh berikut meluncurkan sebuah klaster Amazon EMR menggunakan AMI versi 1.0 dan 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"]
}
```

Meluncurkan klaster Amazon EMR dengan label rilis emr-4.x atau lebih besar

Example

Contoh berikut meluncurkan klaster Amazon EMR menggunakan bidang `releaseLabel` yang lebih baru:

```
{
  "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"}
}
```

Menginstal perangkat lunak tambahan pada klaster Amazon EMR Anda

Example

`EmrCluster` menyediakan bidang `supportedProducts` yang menginstal perangkat lunak pihak ke tiga pada klaster Amazon EMR, misalnya, memungkinkan Anda menginstal distribusi kustom Hadoop, seperti MapR. Ia menerima daftar dipisahkan koma argumen untuk perangkat lunak pihak

ke tiga untuk membaca dan bertindak. Contoh berikut menunjukkan cara menggunakan bidang `supportedProducts` dari `EmrCluster` untuk membuat klaster edisi MapR M3 kustom dengan Karmasphere Analytics terinstal, dan menjalankan objek `EmrActivity` di atasnya.

```
{
    "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"
}
```

Nonaktifkan enkripsi sisi server pada rilis 3.x

Example

`EmrCluster`Aktivitas dengan Hadoop versi 2.x yang dibuat oleh AWS Data Pipeline mengaktifkan enkripsi sisi server secara default. Jika Anda ingin nonaktifkan enkripsi sisi server, Anda harus menentukan tindakan bootstrap dalam definisi objek klaster.

Contoh berikut membuat aktivitas `EmrCluster` dengan enkripsi sisi server dinonaktifkan:

```
{
   "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"]
}
```

Nonaktifkan enkripsi sisi server pada rilis 4.x

Example

Anda harus nonaktifkan enkripsi sisi server menggunakan objek `EmrConfiguration`.

Contoh berikut membuat aktivitas `EmrCluster` dengan enkripsi sisi server dinonaktifkan:

```
  {
     "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"
   }
```

Konfigurasikan Hadoop KMS ACLs dan buat zona enkripsi di HDFS

Example

Objek berikut dibuat ACLs untuk Hadoop KMS dan membuat zona enkripsi dan kunci enkripsi yang sesuai di 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"
    }
```

Tentukan IAM role khusus

Example

Secara default, AWS Data Pipeline masuk `DataPipelineDefaultRole` sebagai peran layanan EMR Amazon dan `DataPipelineDefaultResourceRole` sebagai profil EC2 instans Amazon untuk membuat sumber daya atas nama Anda. Namun, Anda dapat membuat peran layanan EMR Amazon khusus dan profil instans khusus dan menggunakannya sebagai gantinya. AWS Data Pipeline harus memiliki izin yang cukup untuk membuat cluster menggunakan peran khusus, dan Anda harus menambahkan AWS Data Pipeline sebagai entitas tepercaya.

Contoh berikut objek menentukan peran kustom untuk klaster 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"
}
```

Menggunakan EmrCluster Sumber Daya di AWS SDK for Java

Example

Contoh berikut menunjukkan cara menggunakan `EmrCluster` dan `EmrActivity` untuk membuat klaster Amazon EMR 4.x untuk menjalankan langkah Spark menggunakan 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);
```

```
    }

}
```

Mengonfigurasi klaster Amazon EMR di subnet privat

Example

Contoh ini mencakup konfigurasi yang meluncurkan klaster ke subnet privat di VPC. Untuk informasi selengkapnya, lihat Peluncuran Klaster Amazon EMR ke VPC di Panduan Pengelolaan Amazon EMR. Konfigurasi ini opsional. Anda dapat menggunakannya dalam setiap alur yang menggunakan objek `EmrCluster`.

Untuk meluncurkan klaster Amazon EMR di subnet privat, tentukan `SubnetId`, `emrManagedMasterSecurityGroupId`, `emrManagedSlaveSecurityGroupId`, dan `serviceAccessSecurityGroupId` di konfigurasi `EmrCluster` Anda.

```
{
  "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"
  }
}
```

Melampirkan volume EBS ke simpul klaster

Example

Anda dapat melampirkan volume EBS untuk semua jenis simpul di klaster EMR dalam alur Anda. Untuk melampirkan volume EBS ke simpul, gunakan `coreEbsConfiguration`, `masterEbsConfiguration`, dan `TaskEbsConfiguration` di konfigurasi `EmrCluster` Anda.

Contoh klaster Amazon EMR ini menggunakan volume Amazon EBS untuk master, tugas, dan simpul intinya. Untuk informasi selengkapnya, lihat Volume Amazon EBS di Amazon EMR di Panduan Pengelolaan Amazon EMR.

Konfigurasi ini bersifat opsional. Anda dapat menggunakannya dalam setiap alur yang menggunakan objek `EmrCluster`.

Dalam alur, klik konfigurasi objek `EmrCluster`, memilih Konfigurasi Utama EBS, Konfigurasi Inti EBS, atau Konfigurasi EBS Tugas, dan memasukkan detail konfigurasi yang serupa dengan contoh berikut.

```
{
  "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"
  }
}
```

Lihat Juga

- EmrActivity

HttpProxy

HttpProxy memungkinkan Anda untuk mengkonfigurasi proxy Anda sendiri dan membuat Task Runner mengakses AWS Data Pipeline layanan melalui itu. Anda tidak perlu mengonfigurasi Task Runner yang sedang berjalan dengan informasi ini.

Contoh dari sebuah HttpProxy in TaskRunner

Definisi alur berikut menunjukkan objek `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": []
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
hostname	Host dari proxy yang klien akan gunakan untuk connect ke Layanan AWS.	String
port	Port dari host proxy yang klien akan gunakan untuk connect ke Layanan AWS.	String

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}
*kata sandi	Kata sandi untuk proxy.	String
s3 NoProxy	Nonaktifkan proxy HTTP saat connect ke Amazon S3	Boolean
nama pengguna	Nama pengguna untuk proxy.	String
windowsDomain	Nama domain Windows untuk NTLM Proxy.	String
windowsWorkgroup	Nama grup kerja Windows untuk NTLM Proxy.	String

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Prasyarat

Berikut ini adalah objek AWS Data Pipeline prasyarat:

Objek

- Dynamo Ada DBData
- Dynamo Ada DBTable
- Exists
- S3 KeyExists
- S3 PrefixNotEmpty
- ShellCommandPrecondition

Dynamo Ada DBData

Prasyarat untuk memeriksa data yang ada di tabel DynamoDB.

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
peran	Menentukan peran yang akan digunakan untuk mengeksekusi prasyarat tersebut.	String

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
tableName	Tabel DynamoDB untuk memeriksa.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSucces

Bidang Opsional	Deskripsi	Jenis Slot
		s”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
preconditionTimeout	Periode dari awal setelah prasyarat ditandai sebagai gagal jika masih belum terpenuhi	Periode
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveInstances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai ketergantungan tempat objek gagal.	Objek Referensi, misalnya "cascadeFailedOn“: {" ref”:”

Bidang Runtime	Deskripsi	Jenis Slot
		myRunnableObject Id "}
currentRetryCount	Berapa kali prasyarat dicoba dalam upaya ini.	String
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
hostname	Nama host klien yang mengambil upaya tugas.	String
lastRetryTime	Terakhir kali ketika prasyarat dicoba dalam upaya ini.	String
simpul	Simpul yang prasyarat ini sedang dilakukan	Objek Referensi, misalnya “node”: {"ref”:” myRunnabl eObject Id "}
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Dynamo Ada DBTable

Prasyarat untuk memeriksa bahwa tabel DynamoDB ada.

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
peran	Menentukan peran yang akan digunakan untuk mengeksekusi prasyarat tersebut.	String
tableName	Tabel DynamoDB untuk memeriksa.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapka n dapat dicoba lagi.	Periode
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAc tion“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSucces s”: {"ref”:” “} myActionI d
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}

Bidang Opsional	Deskripsi	Jenis Slot
preconditionTimeout	Periode dari awal setelah prasyarat ditandai sebagai gagal jika masih belum terpenuhi	Periode
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveInstances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai ketergantungan tempat objek gagal.	Objek Referensi, misalnya "cascadeFailedOn“: {" ref”:” myRunnableObject Id "}
currentRetryCount	Berapa kali prasyarat dicoba dalam upaya ini.	String
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String

Bidang Runtime	Deskripsi	Jenis Slot
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
hostname	Nama host klien yang mengambil upaya tugas.	String
lastRetryTime	Terakhir kali ketika prasyarat dicoba dalam upaya ini.	String
simpul	Simpul yang prasyarat ini sedang dilakukan	Objek Referensi, misalnya “node”: {"ref”:” myRunnabl eObject Id "}
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Exists

Memeriksa apakah simpul data ada.

> **Note**
>
> Kami rekomendasikan agar Anda menggunakan prasyarat yang terkelola sistem. Untuk informasi selengkapnya, lihat Prasyarat.

Contoh

Berikut adalah contoh dari jenis objek ini. Objek `InputData` mereferensikan objek ini, `Ready`, ditambah objek lain yang akan Anda tetapkan dalam file definisi alur yang sama. `CopyPeriod` adalah objek `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"
}
```

Sintaksis

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke `ondemand`.	Periode
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}
preconditionTimeout	Periode dari awal setelah prasyarat ditandai sebagai gagal jika masih belum terpenuhi	Periode
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveInstances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeFailedOn“: {" ref”:” myRunnableObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
hostname	Nama host klien yang mengambil upaya tugas.	String
simpul	Simpul yang prasyarat ini sedang dilakukan.	Objek Referensi, misalnya “node”: {"ref”:” myRunnabl eObject Id "}
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Lihat Juga

- ShellCommandPrecondition

S3 KeyExists

Memeriksa apakah kunci ada di simpul data Amazon S3.

Contoh

Berikut adalah contoh dari jenis objek ini. Prasyarat akan memicu ketika kunci, `s3://amzn-s3-demo-bucket/mykey`, direferensikan oleh parameter `s3Key`, ada.

```
{
"id" : "InputReady",
"type" : "S3KeyExists",
"role" : "test-role",
"s3Key" : "s3://amzn-s3-demo-bucket/mykey"
}
```

Anda juga dapat menggunakan `S3KeyExists` sebagai prasyarat pada alur kedua yang menunggu alur pertama selesai. Untuk melakukannya:

1. Tulis file ke Amazon S3 pada akhir penyelesaian alur pertama ini.
2. Buat prasyarat `S3KeyExists` pada alur kedua.

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
peran	Menentukan peran yang akan digunakan untuk mengeksekusi prasyarat tersebut.	String
s3Key	Kunci Amazon S3.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout sebelum mencoba menyelesaikan pekerjaan jarak jauh sekali lagi. Jika disetel, maka aktivitas jarak jauh yang tidak lengkap dalam waktu mulai yang ditetapkan mungkin dicoba lagi.	Periode
failureAndRerunMod us	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali.	Pencacahan
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maximumRetries	Jumlah maksimum upaya yang dimulai pada kegagalan.	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAc

Bidang Opsional	Deskripsi	Jenis Slot
		tion“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSucces s”: {"ref”:” “} myActionI d
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}
preconditionTimeout	Periode dari awal setelah prasyarat ditandai sebagai gagal jika masih belum terpenuhi.	Periode
reportProgressTime out	Timeout untuk panggilan berurutan kerja jarak jauh ke `reportProgress` . Jika disetel, maka kegiatan jarak jauh yang tidak melaporka n kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya berturut-turut.	Periode

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeFailedOn“: {" ref”:” myRunnableObject Id "}
currentRetryCount	Berapa kali prasyarat dicoba dalam upaya ini.	String
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
hostname	Nama host klien yang mengambil upaya tugas.	String
lastRetryTime	Terakhir kali ketika prasyarat dicoba dalam upaya ini.	String
simpul	Simpul yang prasyarat ini sedang dilakukan	Objek Referensi, misalnya “node”: {"ref”:” myRunnableObject Id "}
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Lihat Juga

- ShellCommandPrecondition

S3 PrefixNotEmpty

Prasyarat untuk memeriksa bahwa objek Amazon S3 dengan prefiks yang diberikan (direpresentasikan sebagai URI) ada.

Contoh

Berikut ini adalah contoh dari jenis objek ini menggunakan bidang yang diperlukan, opsional, dan ekspresi.

```
{
  "id" : "InputReady",
  "type" : "S3PrefixNotEmpty",
  "role" : "test-role",
  "s3Prefix" : "#{node.filePath}"
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
peran	Menentukan peran yang akan digunakan untuk mengeksekusi prasyarat tersebut.	String
s3Prefix	Prefiks Amazon S3 untuk memeriksa keberadaan objek.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke `ondemand`.	Periode
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat

Bidang Opsional	Deskripsi	Jenis Slot
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
preconditionTimeout	Periode dari awal setelah prasyarat ditandai sebagai gagal jika masih belum terpenuhi	Periode
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveIn

Bidang Runtime	Deskripsi	Jenis Slot
		stances”: {"ref”:” Id "} myRunnableObject
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
currentRetryCount	Berapa kali prasyarat dicoba dalam upaya ini.	String
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
hostname	Nama host klien yang mengambil upaya tugas.	String
lastRetryTime	Terakhir kali ketika prasyarat dicoba dalam upaya ini.	String
simpul	Simpul yang prasyarat ini sedang dilakukan.	Objek Referensi, misalnya “node”: {"ref”:” myRunnabl eObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan akhir waktu untuk objek.	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek.	DateTime
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Lihat Juga

- ShellCommandPrecondition

ShellCommandPrecondition

Perintah Unix/Linux shell yang dapat dijalankan sebagai prasyarat.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "VerifyDataReadiness",
  "type" : "ShellCommandPrecondition",
  "command" : "perl check-data-ready.pl"
}
```

Sintaksis

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
perintah	Perintah yang akan dijalankan. Nilai ini dan setiap parameter terkait harus berfungsi di lingkungan dari mana Anda menjalankan Task Runner.	String
scriptUri	Sebuah jalur Amazon S3 URI untuk file untuk mengunduh dan menjalankan sebagai perintah shell. Hanya satu scriptURI atau bidang perintah yang harus ada. scriptURI tidak dapat menggunakan parameter, menggunakan perintah sebagai gantinya.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptStatus	Baru-baru ini melaporkan status dari aktivitas jarak jauh.	String

Bidang Opsional	Deskripsi	Jenis Slot
attemptTimeout	Timeout untuk penyelesaian pekerjaan jarak jauh. Jika disetel maka aktivitas jarak jauh yang tidak selesai dalam waktu mulai yang ditetapkan dapat dicoba lagi.	Periode
failureAndRerunModus	Menjelaskan perilaku simpul konsumen ketika dependensi gagal atau menjalankan kembali	Pencacahan
lateAfterTimeout	Waktu berlalu setelah alur mulai di mana objek harus menyelesaikan. Hal ini dipicu hanya ketika jenis jadwal tidak disetel ke ondemand.	Periode
maximumRetries	Jumlah maksimum percobaan ulang pada pelanggaran	Bilangan Bulat
onFail	Tindakan untuk dijalankan ketika objek saat ini gagal.	Objek Referensi, misalnya “onFail”: {"ref”:” “} myActionId
onLateAction	Tindakan yang harus dipicu jika objek belum dijadwalkan atau masih belum selesai.	Objek Referensi, misalnya "onLateAction“: {" ref”:” myActionId “}
onSuccess	Tindakan untuk dijalankan ketika objek saat ini berhasil.	Objek Referensi, misalnya “onSuccess”: {"ref”:” “} myActionId
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObjectId "}
preconditionTimeout	Periode dari awal setelah prasyarat ditandai sebagai gagal jika masih belum terpenuhi	Periode

Bidang Opsional	Deskripsi	Jenis Slot
reportProgressTimeout	Timeout untuk panggilan kerja jarak jauh berturut-turut ke reportProgress. Jika disetel, maka kegiatan jarak jauh yang tidak melaporkan kemajuan untuk jangka waktu tertentu dapat dianggap terhenti dan jadi dicoba lagi.	Periode
retryDelay	Durasi timeout antara dua upaya coba lagi.	Periode
scriptArgument	Argumen yang akan diteruskan ke script shell	String
stderr	Jalur Amazon S3 yang menerima olahpesan kesalahan sistem dialihkan dari perintah. Jika Anda menggunakan bidang `runsOn`, ini harus menjadi jalur Amazon S3 karena sifat sementara dari sumber daya yang menjalankan aktivitas Anda. Namun, jika Anda menentukan bidang `workerGroup` , jalur file lokal diizinkan .	String
stdout	Jalur Amazon S3 yang menerima output yang dialihkan dari perintah. Jika Anda menggunakan bidang `runsOn`, ini harus menjadi jalur Amazon S3 karena sifat sementara dari sumber daya yang menjalankan aktivitas Anda. Namun, jika Anda menentukan bidang `workerGroup` , jalur file lokal diizinkan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@activeInstances	Daftar objek instans aktif terjadwal saat ini.	Objek Referensi, misalnya “ActiveInstances”: {"ref”:” Id "} myRunnableObject

Bidang Runtime	Deskripsi	Jenis Slot
@actualEndTime	Waktu ketika eksekusi objek ini selesai.	DateTime
@actualStartTime	Waktu ketika eksekusi objek ini dimulai.	DateTime
cancellationReason	cancellationReason jika objek ini dibatalkan.	String
@cascadeFailedOn	Deskripsi rantai dependensi tempat objek gagal.	Objek Referensi, misalnya "cascadeF ailedOn“: {" ref”:” myRunnableObject Id "}
emrStepLog	Log langkah EMR hanya tersedia pada upaya aktivitas EMR	String
errorId	errorId jika objek ini gagal.	String
errorMessage	errorMessage jika objek ini gagal.	String
errorStackTrace	Jejak tumpukan kesalahan jika objek ini gagal.	String
hadoopJobLog	log tugas Hadoop tersedia pada upaya untuk kegiatan berbasis EMR.	String
hostname	Nama host klien yang mengambil upaya tugas.	String
simpul	Simpul yang prasyarat ini sedang dilakukan	Objek Referensi, misalnya “node”: {"ref”:” myRunnabl eObject Id "}
reportProgressTime	Waktu terbaru bahwa aktivitas jarak jauh melaporkan kemajuan.	DateTime
@scheduledEndTime	Jadwalkan waktu akhir untuk objek	DateTime
@scheduledStartTime	Jadwalkan waktu mulai untuk objek	DateTime

Bidang Runtime	Deskripsi	Jenis Slot
@status	Status objek ini.	String
@version	Versi alur objek dibuat dengan.	String
@waitingOn	Deskripsi daftar dependensi objek ini sedang menunggu.	Objek Referensi, misalnya “WaitingO n”: {"ref”:” Id "} myRunnableObject

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Lihat Juga

- ShellCommandActivity
- Exists

Basis Data

Berikut ini adalah objek AWS Data Pipeline database:

Objek

- JdbcDatabase
- RdsDatabase

- RedshiftDatabase

JdbcDatabase

Mendefinisikan basis data JDBC.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "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"
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
connectionString	rangkaian koneksi JDBC untuk mengakses basis data.	String
jdbcDriverClass	Kelas driver untuk memuat sebelum membangun koneksi JDBC.	String
*kata sandi	Kata sandi untuk memasok.	String
nama pengguna	Nama pengguna untuk memasok saat connect ke basis data.	String

Bidang Opsional	Deskripsi	Jenis Slot
databaseName	Nama basis data logis untuk dilampirkan	String
jdbcDriverJarUri	Lokasi di Amazon S3 dari file JAR driver JDBC yang digunakan untuk connect ke basis data. AWS Data Pipeline harus memiliki izin untuk baca file JAR ini.	String
jdbcProperties	Pasangan dari bentuk A=B yang akan ditetapkan sebagai properti pada koneksi JDBC untuk basis data ini.	String
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur tempat objek dibuat.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

RdsDatabase

Mendefinisikan basis data Amazon RDS.

> **Note**
>
> RdsDatabase tidak mendukung Aurora. Gunakan the section called “JdbcDatabase” untuk Aurora, sebagai gantinya.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "MyRdsDatabase",
  "type" : "RdsDatabase",
  "region" : "us-east-1",
  "username" : "user_name",
  "*password" : "my_password",
  "rdsInstanceId" : "my_db_instance_identifier"
}
```

Untuk mesin Oracle, bidang `jdbcDriverJarUri` diperlukan dan Anda dapat menentukan driver berikut: `http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html`. Untuk mesin SQL Server, bidang `jdbcDriverJarUri` diperlukan dan Anda dapat menentukan driver berikut: `https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774`. Untuk mesin MySQL dan PostgreSQL, bidang `jdbcDriverJarUri` bersifat opsional.

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
*kata sandi	Kata sandi untuk memasok.	String
rdsInstanceId	Properti `DBInstanceIdentifier` dari instans DB.	String

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
nama pengguna	Nama pengguna untuk memasok saat connect ke basis data.	String

Bidang Opsional	Deskripsi	Jenis Slot
databaseName	Nama basis data logis untuk dilampirkan	String
jdbcDriverJarUri	Lokasi di Amazon S3 dari file JAR driver JDBC yang digunakan untuk connect ke basis data. AWS Data Pipeline harus memiliki izin untuk baca file JAR ini. Untuk mesin MySQL dan PostgreSQL, driver default digunakan jika bidang ini tidak ditentukan, tetapi Anda dapat mengganti default menggunakan bidang ini. Untuk mesin Oracle dan SQL Server, bidang ini diperlukan.	String
jdbcProperties	Pasangan dari bentuk A=B yang akan ditetapkan sebagai properti pada koneksi JDBC untuk basis data ini.	String
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya, “induk”: {"ref”:” myBaseObject Id "}
region	Kode untuk wilayah di mana basis data ada. Misalnya, us-east-1.	String

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur tempat objek dibuat.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

RedshiftDatabase

Mendefinisikan basis data menggunakan Amazon Redshift. `RedshiftDatabase`emewakili properti basis data yang digunakan oleh alur Anda.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "MyRedshiftDatabase",
  "type" : "RedshiftDatabase",
  "clusterId" : "myRedshiftClusterId",
  "username" : "user_name",
  "*password" : "my_password",
  "databaseName" : "database_name"
}
```

Secara default, objek menggunakan driver Postgres, yang memerlukan bidang `clusterId`. Untuk menggunakan driver Amazon Redshift, tentukan rangkaian koneksi basis data Amazon Redshif dari

konsol Amazon Redshift (dimulai dengan "jdbc:redshift:") di bidang `connectionString` sebagai gantinya.

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
*kata sandi	Kata sandi untuk memasok.	String
nama pengguna	Nama pengguna untuk memasok saat connect ke basis data.	String

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
clusterId	Pengenal yang disediakan oleh pengguna ketika klaster Amazon Redshift dibuat. Misalnya, jika titik akhir untuk klaster Amazon Redshift Anda adalah mydb.example.us-east-1.redshift.amazonaws.com, pengenal yang benar adalah `mydb`. Dalam konsol Amazon Redshift, Anda bisa mendapatkan nilai ini dari Pengenal Klaster atau Nama Klaster.	String
connectionString	Titik akhir JDBC untuk connect ke instans Amazon Redshift yang dimiliki oleh akun yang berbeda dari alur. Anda tidak dapat menentukan `connectionString` dan `clusterId` sekaligus.	String

Bidang Opsional	Deskripsi	Jenis Slot
databaseName	Nama basis data logis untuk dilampirkan.	String
jdbcProperties	Pasangan dari bentuk A=B akan ditetapkan sebagai properti pada koneksi JDBC untuk basis data ini.	String
induk	Induk dari objek saat ini dari mana slot diwariskan.	Objek Referensi, misalnya, “induk”: {"ref”:” myBaseObject Id "}
region	Kode untuk wilayah di mana basis data ada. Misalnya, us-east-1.	Pencacahan

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur tempat objek dibuat.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Format Data

Berikut ini adalah objek format AWS Data Pipeline data:

Objek

- Format Data CSV
- Format Data Kustom
- Format Dinamo DBData
- Dinamo DBExport DataFormat
- RegEx Format Data
- Format Data TSV

Format Data CSV

Format data yang dibatasi koma di mana pemisah kolom adalah koma dan pemisah catatan adalah karakter baris baru.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "MyOutputDataType",
  "type" : "CSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintaksis

Bidang Opsional	Deskripsi	Jenis Slot
kolom	Nama kolom dengan jenis data yang ditentuka n oleh masing-masing bidang untuk data	String

Bidang Opsional	Deskripsi	Jenis Slot
	yang dijelaskan oleh simpul data ini. Contoh: hostname STRING Untuk beberapa nilai, menggunakan nama kolom dan tipe data yang dipisahkan oleh spasi.	
escapeChar	Sebuah karakter, misalnya "\", yang menginstruksikan parser untuk mengabaikan karakter berikutnya.	String
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Format Data Kustom

Format data kustom yang didefinisikan oleh kombinasi pemisah kolom tertentu, pemisah catatan, dan karakter escape.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "MyOutputDataType",
  "type" : "Custom",
  "columnSeparator" : ",",
  "recordSeparator" : "\n",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
columnSeparator	Sebuah karakter yang menunjukkan akhir kolom dalam file data.	String

Bidang Opsional	Deskripsi	Jenis Slot
kolom	Nama kolom dengan jenis data yang ditentukan oleh masing-masing bidang untuk data yang dijelaskan oleh simpul data ini. Contoh: hostname STRING Untuk beberapa nilai, menggunakan nama kolom dan tipe data yang dipisahkan oleh spasi.	String

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}
recordSeparator	Karakter yang menunjukkan akhir baris dalam file data, misalnya "\n". Hanya karakter tunggal yang disupport.	String

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Format Dinamo DBData

Berlaku skema untuk tabel DynamoDB untuk membuatnya dapat diakses oleh kueri Hive. `DynamoDBDataFormat` digunakan dengan objek `HiveActivity` dan input dan output `DynamoDBDataNode`. `DynamoDBDataFormat` mengharuskan Anda menentukan semua kolom dalam kueri Hive Anda. Untuk lebih banyak fleksibilitas untuk menentukan kolom tertentu dalam kueri Hive atau support Amazon S3, lihat Dinamo DBExport DataFormat.

> **Note**
>
> Jenis DynamoDB Boolean tidak dipetakan ke jenis Hive Boolean. Namun, adalah mungkin untuk memetakan nilai integer DynamoDB 0 atau 1 untuk jenis Hive Boolean.

Contoh

Contoh berikut menunjukkan cara menggunakan `DynamoDBDataFormat` untuk menetapkan skema untuk input `DynamoDBDataNode`, yang mengizinkan objek `HiveActivity` untuk mengakses data dengan kolom bernama dan menyalin data ke 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"
    }
  ]
}
```

Sintaksis

Bidang Opsional	Deskripsi	Jenis Slot
kolom	Nama kolom dengan jenis data yang ditentukan oleh masing-masing bidang untuk data yang dijelaskan oleh simpul data ini. Misalnya, `hostname STRING`. Untuk beberapa nilai, gunakan nama kolom dan tipe data yang dipisahkan oleh spasi.	String

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi , seperti “parent”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur digunakan untuk membuat objek.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Kesalahan yang menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Dinamo DBExport DataFormat

Berlaku skema untuk tabel DynamoDB untuk membuatnya dapat diakses oleh kueri Hive. Gunakan `DynamoDBExportDataFormat` dengan objek `HiveCopyActivity` dan `DynamoDBDataNode` atau `S3DataNode` input dan output. `DynamoDBExportDataFormat` memiliki manfaat berikut:

- Memberikan support DynamoDB dan Amazon S3
- Mengizinkan Anda untuk mem-filter data dengan kolom tertentu dalam kueri Hive Anda
- Ekspor semua atribut dari DynamoDB bahkan jika Anda memiliki skema tersebar

> **Note**
>
> Jenis DynamoDB Boolean tidak dipetakan ke jenis Hive Boolean. Namun, adalah mungkin untuk memetakan nilai integer DynamoDB 0 atau 1 untuk jenis Hive Boolean.

Contoh

Contoh berikut menunjukkan cara menggunakan `HiveCopyActivity` dan `DynamoDBExportDataFormat` untuk menyalin data dari satu `DynamoDBDataNode` ke yang lain, sambil mem-filter berdasarkan stempel waktu.

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

Sintaksis

Bidang Opsional	Deskripsi	Jenis Slot
kolom	Nama kolom dengan jenis data yang ditentuka n oleh masing-masing bidang untuk data yang dijelaskan oleh simpul data ini. Contoh: nama host STRING	String
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”:

Bidang Opsional	Deskripsi	Jenis Slot
		{"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

RegEx Format Data

Format data kustom yang didefinisikan oleh ekspresi reguler.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "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"
  ]
}
```

Sintaksis

Bidang Opsional	Deskripsi	Jenis Slot
kolom	Nama kolom dengan jenis data yang ditentuka n oleh masing-masing bidang untuk data yang dijelaskan oleh simpul data ini. Contoh: hostname STRING Untuk beberapa nilai, menggunakan nama kolom dan tipe data yang dipisahkan oleh spasi.	String
inputRegEx	Ekspresi reguler untuk mengurai file input S3. inputRegEx menyediakan cara untuk mengambil kolom dari data yang relatif tidak terstruktur dalam file.	String
outputFormat	Kolom kolom diambil oleh inputRegEx, tetapi direferensikan sebagai %1$s %2$s menggunak an sintaks pemformat Java.	String
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Format Data TSV

Format data yang dibatasi koma di mana pemisah kolom adalah karakter tab dan pemisah catatan adalah karakter baris baru.

Contoh

Berikut adalah contoh dari jenis objek ini.

```
{
  "id" : "MyOutputDataType",
  "type" : "TSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Sintaksis

Bidang Opsional	Deskripsi	Jenis Slot
kolom	Nama kolom dan tipe data untuk data yang dijelaskan oleh simpul data ini. Misalnya `"Name STRING"` menunjukkan sebuah kolom bernama `Name` dengan bidang tipe data `STRING`. Pisahkan beberapa nama kolom dan tipe data pasangan dengan koma (seperti yang ditunjukkan pada contoh).	String
columnSeparator	Karakter yang memisahkan bidang dalam satu kolom dari bidang di kolom berikutnya. Secara default ke '\t'.	String
escapeChar	Sebuah karakter, misalnya "\", yang menginstruksikan parser untuk mengabaikan karakter berikutnya.	String
induk	Induk dari objek saat ini dari mana slot diwariskan.	Objek Referensi, misalnya, “induk”: {"ref”:” myBaseObject Id "}
recordSeparator	Karakter yang memisahkan catatan. Secara default ke '\n'.	String

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur tempat objek dibuat.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects, yang mengeksekusi Attempt Objects.	String

Tindakan

Berikut ini adalah objek AWS Data Pipeline tindakan:

Objek

- SnsAlarm
- Mengakhiri

SnsAlarm

Mengirim olahpesan notifikasi Amazon SNS ketika aktivitas gagal atau selesai dengan sukses.

Contoh

Berikut adalah contoh dari jenis objek ini. Nilai untuk `node.input` dan `node.output` berasal dari simpul data atau aktivitas yang mereferensikan objek ini dalam bidang `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}."
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
pesan	Teks badan notifikasi Amazon SNS.	String
peran	IAM role untuk digunakan untuk membuat alarm Amazon SNS.	String
subjek	Baris subjek olahpesan notifikasi Amazon SNS.	String
topicArn	Tujuan Amazon SNS topik ARN untuk olahpesan.	String

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
simpul	Simpul untuk tempat tindakan ini sedang dilakukan.	Objek Referensi, misalnya “node”: {"ref”:” myRunnabl eObject Id "}
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	Id dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects.	String

Mengakhiri

Tindakan untuk memicu pembatalan aktivitas, sumber daya, atau node data yang tertunda atau belum selesai. AWS Data Pipeline mencoba untuk menempatkan aktivitas, sumber daya, atau node data ke dalam status CANCELLED jika tidak dimulai dengan `lateAfterTimeout` nilai.

Anda tidak dapat mengakhiri tindakan yang menyertakan sumber daya `onSuccess`, `OnFail`, atau `onLateAction`.

Contoh

Berikut adalah contoh dari jenis objek ini. Dalam contoh ini, bidang `onLateAction` dari `MyActivity` berisi referensi untuk tindakan `DefaultAction1`. Saat Anda memberikan tindakan untuk `onLateAction`, Anda juga harus menyediakan nilai `lateAfterTimeout` untuk menunjukkan periode waktu sejak awal dijadwalkan dari alur setelah aktivitas dianggap terlambat.

```
{
  "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"
}
```

Sintaksis

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
simpul	Simpul untuk tempat tindakan ini sedang dilakukan.	Objek Referensi, misalnya “node”: {"ref”:” myRunnabl eObject Id "}
@version	Versi alur tempat objek dibuat.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String

Bidang Sistem	Deskripsi	Jenis Slot
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects, yang mengeksekusi Attempt Objects.	String

Jadwal

Mendefinisikan waktu acara terjadwal, seperti ketika suatu aktivitas berjalan.

> **Note**
>
> Ketika waktu mulai jadwal sudah berlalu, AWS Data Pipeline isi ulang pipeline Anda dan mulai penjadwalan berjalan segera dimulai pada waktu mulai yang ditentukan. Untuk pengujian/pengembangan, gunakan interval yang relatif singkat. Jika tidak, AWS Data Pipeline cobalah untuk mengantri dan menjadwalkan semua proses pipeline Anda untuk interval itu. AWS Data Pipeline upaya untuk mencegah timbunan ulang yang tidak disengaja jika komponen `scheduledStartTime` pipa lebih awal dari 1 hari yang lalu dengan memblokir aktivasi pipa.

Contoh

Berikut adalah contoh dari jenis objek ini. Ini mendefinisikan jadwal setiap jam mulai pukul 00:00:00 pada 2012-09-01 dan berakhir pada jam 00:00:00 pada 2012-10-01. Periode pertama berakhir pada pukul 01:00:00 pada 2012-09-01.

```
{
  "id" : "Hourly",
  "type" : "Schedule",
  "period" : "1 hours",
  "startDateTime" : "2012-09-01T00:00:00",
  "endDateTime" : "2012-10-01T00:00:00"
}
```

Alur berikut akan dimulai pada FIRST_ACTIVATION_DATE_TIME dan berjalan setiap jam sehingga jam 22:00:00 pada 2014-04-25.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "endDateTime": "2014-04-25T22:00:00"
   }
```

Alur berikut akan dimulai pada FIRST_ACTIVATION_DATE_TIME dan berjalan setiap jam dan selesai setelah tiga kejadian.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

Alur berikut akan dimulai pukul 22:00:00 pada 2014-04-25, berjalan per jam, dan berakhir setelah tiga kejadian.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startDateTime": "2014-04-25T22:00:00",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

Sesuai permintaan menggunakan objek Default

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
```

```
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
}
```

Sesuai permintaan dengan objek Jadwal eksplisit

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
},
{
  "name": "DefaultSchedule",
  "type": "Schedule",
  "id": "DefaultSchedule",
  "period": "ONDEMAND_PERIOD",
  "startAt": "ONDEMAND_ACTIVATION_TIME"
},
```

Contoh berikut menunjukkan bagaimana Jadwal dapat diwariskan dari objek default, secara eksplisit disetel untuk objek itu, atau diberikan oleh referensi induk:

Jadwal yang diwarisi dari objek Default

```
{
  "objects": [
  {
      "id": "Default",
      "failureAndRerunMode":"cascade",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "s3://myLogsbucket",
      "scheduleType": "cron",
      "schedule": {
        "ref": "DefaultSchedule"
      }
   },
   {
      "type": "Schedule",
      "id": "DefaultSchedule",
      "occurrences": "1",
      "period": "1 Day",
```

```
      "startAt": "FIRST_ACTIVATION_DATE_TIME"
    },
    {
      "id": "A_Fresh_NewEC2Instance",
      "type": "Ec2Resource",
      "terminateAfter": "1 Hour"
    },
    {
      "id": "ShellCommandActivity_HelloWorld",
      "runsOn": {
        "ref": "A_Fresh_NewEC2Instance"
      },
      "type": "ShellCommandActivity",
      "command": "echo 'Hello World!'"
    }
  ]
}
```

Jadwal eksplisit pada objek

```
{
  "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!'"
    }
  ]
}
```

Jadwal dari referensi Orang Tua

```
{
  "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!'"
    }
  ]
}
```

Sintaksis

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
periode	Seberapa sering alur harus berjalan. Formatnya adalah "N [menit|jam|hari|minggu|bulan]", di mana N adalah nomor diikuti oleh salah satu penentu waktu. Misalnya, "15 menit", menjalankan alur setiap 15 menit. Periode minimum adalah 15 menit dan periode maksimum adalah 3 tahun.	Periode

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
startAt	Tanggal dan waktu untuk mulai alur terjadwal. Nilai yang valid adalah FIRST_ACTIVATION_D	Pencacahan

Grup yang diperlukan (Salah satu dari berikut ini diperlukan)	Deskripsi	Jenis Slot
	ATE_TIME, yang tidak lagi digunakan untuk membuat alur sesuai permintaan.	
startDateTime	Tanggal dan waktu untuk mulai proses terjadwal. Anda harus menggunakan salah satu startDateTime atau StartAt tetapi tidak keduanya.	DateTime

Bidang Opsional	Deskripsi	Jenis Slot
endDateTime	Tanggal dan waktu untuk mengakhiri proses terjadwal. Harus tanggal dan waktu lebih lambat dari nilai startDateTime atau startAt. Perilaku default adalah untuk menjadwalkan proses berjalan sampai alur dimatikan.	DateTime
kejadian	Berapa kali mengeksekusi alur setelah diaktifkan. Anda tidak dapat menggunakan kejadian dengan endDateTime.	Bilangan Bulat
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@firstActivationTime	Waktu pembuatan objek.	DateTime
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Utilitas

Objek utilitas berikut mengonfigurasi objek alur lainnya:

Topik

- ShellScriptConfig
- EmrConfiguration
- Properti

ShellScriptConfig

Gunakan dengan Aktivitas untuk menjalankan skrip shell untuk preActivityTask Config dan Config postActivityTask. Objek ini tersedia untuk HadoopActivity, HiveActivity, HiveCopyActivity, dan PigActivity. Anda menentukan URI S3 dan daftar argumen untuk script.

Contoh

A ShellScriptConfig dengan argumen:

```
{
   "id" : "ShellScriptConfig_1”,
   "name" : “prescript”,
   "type" : "ShellScriptConfig",
   "scriptUri": “s3://my-bucket/shell-cleanup.sh”,
```

```
  "scriptArgument" : ["arg1","arg2"]
}
```

Sintaksis

Objek ini mencakup bidang berikut.

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot diwariskan.	Objek Referensi, misalnya, “induk”: {"ref”:” myBaseObject Id "}
scriptArgument	Daftar argumen untuk digunakan dengan script shell.	String
scriptUri	Script URI di Amazon S3 yang harus diunduh dan dijalankan.	String

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur tempat objek dibuat.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects, yang mengeksekusi Attempt Objects.	String

EmrConfiguration

EmrConfiguration Objek adalah konfigurasi yang digunakan untuk cluster EMR dengan rilis 4.0.0 atau lebih besar. Konfigurasi (sebagai daftar) adalah parameter untuk panggilan RunJobFlow API. API konfigurasi untuk Amazon EMR mengambil klasifikasi dan properti. AWS Data Pipeline menggunakan EmrConfiguration dengan objek Properti yang sesuai untuk mengkonfigurasi EmrCluster aplikasi seperti Hadoop, Hive, Spark, atau Pig pada cluster EMR diluncurkan dalam eksekusi pipeline. Karena konfigurasi hanya dapat diubah untuk cluster baru, Anda tidak dapat menyediakan EmrConfiguration objek untuk sumber daya yang ada. Untuk informasi selengkapnya, lihat https://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/.

Contoh

Objek konfigurasi berikut menetapkan properti `io.file.buffer.size` dan `fs.s3.block.size` di `core-site.xml`:

```
[
   {
      "classification":"core-site",
      "properties":
      {
         "io.file.buffer.size": "4096",
         "fs.s3.block.size": "67108864"
      }
   }
]
```

Definisi objek pipeline yang sesuai menggunakan EmrConfiguration objek dan daftar objek Properti di `property` bidang:

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

Contoh berikut adalah konfigurasi bersarang yang digunakan untuk mengatur lingkungan Hadoop dengan klasifikasi `hadoop-env`:

```
[
  {
    "classification": "hadoop-env",
    "properties": {},
    "configurations": [
      {
        "classification": "export",
        "properties": {
          "YARN_PROXYSERVER_HEAPSIZE": "2396"
```

```
        }
      }
    ]
  }
]
```

Objek definisi alur yang sesuai yang menggunakan konfigurasi ini adalah di bawah ini:

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

Contoh berikut memodifikasi properti khusus Hive untuk klaster 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"
        }
    ]
}
```

Sintaksis

Objek ini mencakup bidang berikut.

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
klasifikasi	Klasifikasi untuk konfigurasi.	String

Bidang Opsional	Deskripsi	Jenis Slot
konfigurasi	Sub-konfigurasi untuk konfigurasi ini.	Objek Referensi, misalnya “konfigur asi”: {"ref”:” myEmrConfiguration Id "}
induk	Induk dari objek saat ini dari mana slot akan diwariskan.	Objek Referensi, misalnya “induk”: {"ref”:” myBaseObject Id "}
properti	Properti konfigurasi.	Objek Referensi, misalnya “properti”: {"ref”:” myPropertyId “}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur objek dibuat dengan.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat mendeskripsikan obyek yang tidak terbentuk	String
@pipelineId	Id dari alur tempat objek ini berada	String
@sphere	Lingkup dari sebuah objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects yang mengeksekusi Attempt Objects	String

Lihat Juga

- EmrCluster
- Properti
- Panduan Rilis Amazon EMR

Properti

Sebuah properti kunci-nilai tunggal untuk digunakan dengan objek EmrConfiguration .

Contoh

Definisi pipeline berikut menunjukkan objek dan EmrConfiguration objek Properti yang sesuai untuk meluncurkan 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"
    }
  ]
}
```

Sintaksis

Objek ini mencakup bidang berikut.

Bidang yang Wajib Diisi	Deskripsi	Jenis Slot
kunci	kunci	String
nilai	nilai	String

Bidang Opsional	Deskripsi	Jenis Slot
induk	Induk dari objek saat ini dari mana slot diwariskan.	Objek Referensi, misalnya, “induk”: {"ref”:” myBaseObject Id "}

Bidang Runtime	Deskripsi	Jenis Slot
@version	Versi alur tempat objek dibuat.	String

Bidang Sistem	Deskripsi	Jenis Slot
@error	Galat menggambarkan objek yang tidak terbentuk.	String
@pipelineId	ID dari alur tempat objek ini berada.	String
@sphere	Lingkup objek menunjukkan tempatnya dalam siklus hidup: Component Objects memunculkan Instance Objects, yang mengeksekusi Attempt Objects.	String

Lihat Juga

- EmrCluster
- EmrConfiguration
- Panduan Rilis Amazon EMR

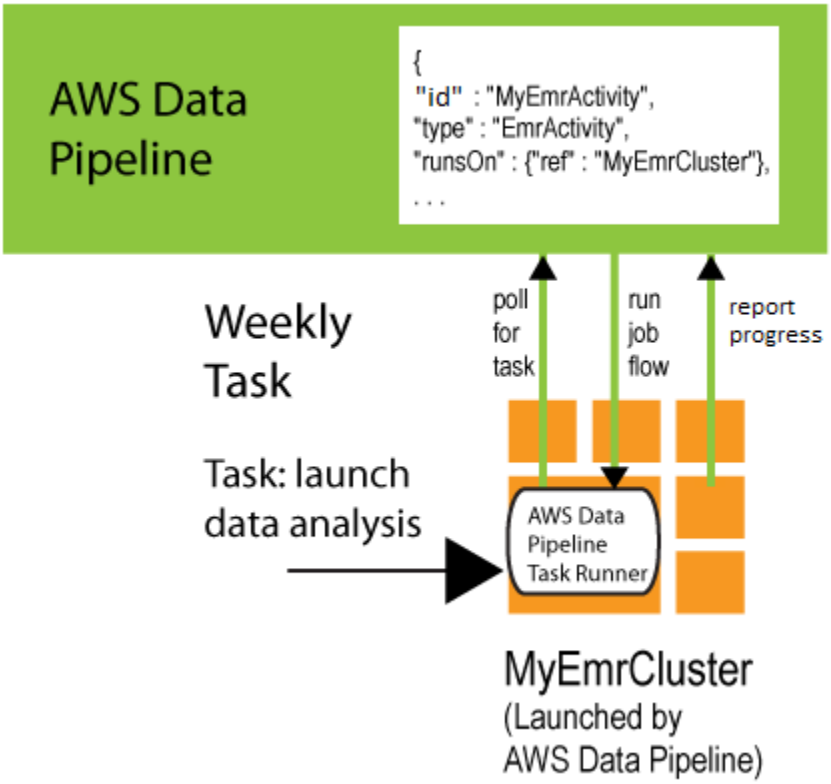
Bekerja dengan Runner Tugas

Task Runner adalah aplikasi agen tugas yang melakukan polling AWS Data Pipeline untuk tugas terjadwal dan menjalankannya di EC2 instans Amazon, kluster EMR Amazon, atau sumber daya komputasi lainnya, melaporkan status saat melakukannya. Tergantung pada aplikasi Anda, Anda dapat memilih untuk:

- Izinkan AWS Data Pipeline untuk menginstal dan mengelola satu atau lebih aplikasi Task Runner untuk Anda. Saat pipeline diaktifkan, default `Ec2Instance` atau `EmrCluster` objek yang direferensikan oleh bidang RunSon aktivitas akan dibuat secara otomatis. AWS Data Pipeline menangani instalasi Task Runner pada EC2 instance atau pada node master dari cluster EMR. Dalam pola ini, AWS Data Pipeline dapat melakukan sebagian besar instance atau manajemen cluster untuk Anda.
- Jalankan semua atau sebagian alur pada sumber daya yang Anda kelola. Sumber daya potensial termasuk EC2 instans Amazon yang berjalan lama, kluster EMR Amazon, atau server fisik. Anda dapat menginstal task runner (yang dapat berupa Task Runner atau agen tugas khusus yang Anda rancang sendiri) hampir di mana saja, asalkan dapat berkomunikasi dengan layanan web. AWS Data Pipeline Dalam pola ini, Anda mengasumsikan kendali hampir penuh atas sumber daya mana yang digunakan dan bagaimana sumber daya tersebut dikelola, dan Anda harus memasang dan mengonfigurasi Runner Tugas secara manual. Untuk melakukannya, gunakan prosedur di bagian ini, seperti yang dijelaskan di Menjalankan Pekerjaan pada Sumber Daya yang Ada Menggunakan Runner Tugas.

Pelari Tugas pada Sumber Daya yang AWS Data Pipeline Dikelola

Ketika sumber daya diluncurkan dan dikelola oleh AWS Data Pipeline, layanan web secara otomatis menginstal Task Runner pada sumber daya tersebut untuk memproses tugas dalam pipeline. Anda menentukan sumber daya komputasi (baik EC2 instance Amazon atau kluster EMR Amazon) untuk `runs0n` bidang objek aktivitas. Saat AWS Data Pipeline meluncurkan sumber daya ini, ia akan memasang Runner Tugas pada sumber daya tersebut dan mengonfigurasinya untuk memproses semua objek aktivitas yang bidang `runs0n`-nya diatur ke sumber daya tersebut. Saat AWS Data Pipeline mengakhiri sumber daya, log Task Runner dipublikasikan ke lokasi Amazon S3 sebelum dimatikan.



Misalnya, jika Anda menggunakan `EmrActivity` di alur, dan menentukan sumber daya `EmrCluster` di bidang `runsOn`. Saat AWS Data Pipeline memproses aktivitas tersebut, ia meluncurkan klaster EMR Amazon dan menginstal Task Runner ke node master. Runner Tugas ini kemudian memproses tugas untuk aktivitas yang bidang `runsOn`-nya disetel ke objek `EmrCluster` itu. Kutipan berikut dari definisi alur menunjukkan hubungan antara dua objek ini.

```
{
  "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"
}
```

Untuk informasi dan contoh menjalankan aktivitas ini, lihat EmrActivity.

Jika Anda memiliki beberapa sumber daya yang AWS Data Pipeline dikelola dalam pipeline, Task Runner diinstal pada masing-masing sumber daya tersebut, dan mereka semua melakukan polling AWS Data Pipeline untuk tugas yang akan diproses.

Menjalankan Pekerjaan pada Sumber Daya yang Ada Menggunakan Runner Tugas

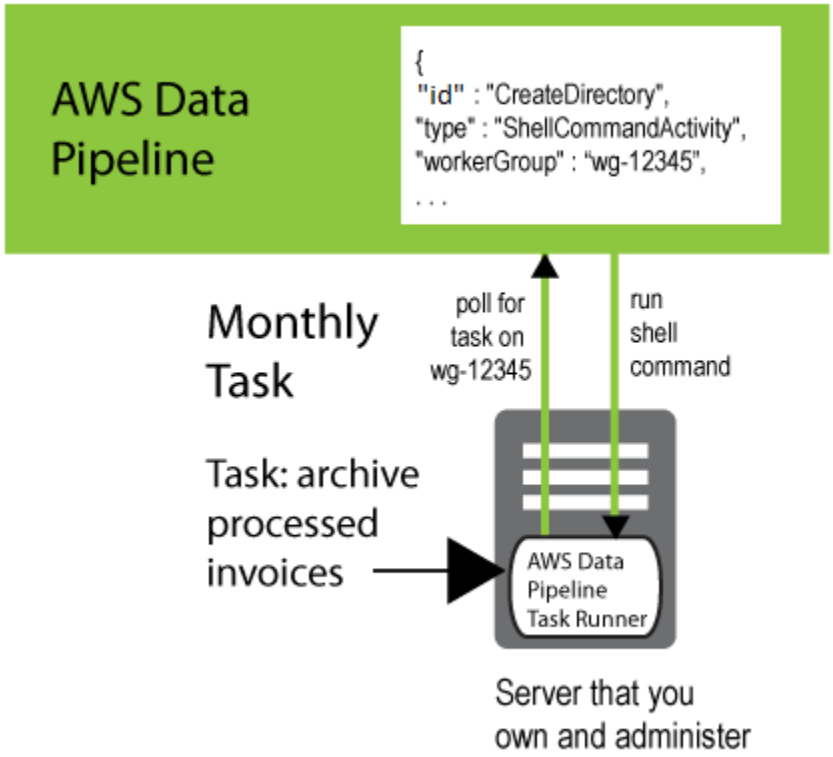
Anda dapat menginstal Task Runner pada sumber daya komputasi yang Anda kelola, seperti EC2 instans Amazon, atau server fisik atau workstation. Task Runner dapat diinstal di mana saja, pada perangkat keras atau sistem operasi yang kompatibel, asalkan dapat berkomunikasi dengan layanan AWS Data Pipeline web.

Pendekatan ini dapat berguna ketika, misalnya, Anda ingin menggunakan AWS Data Pipeline untuk memproses data yang disimpan di dalam firewall organisasi Anda. Dengan menginstal Task Runner di server di jaringan lokal, Anda dapat mengakses database lokal dengan aman dan kemudian melakukan polling AWS Data Pipeline untuk tugas berikutnya yang akan dijalankan. Saat AWS Data Pipeline selesai memproses atau menghapus pipeline, instance Task Runner tetap berjalan di sumber daya komputasi hingga Anda mematikannya secara manual. Log Runner Tugas tetap ada setelah eksekusi alur selesai.

Untuk menggunakan Runner Tugas pada sumber daya yang Anda kelola, Anda harus mengunduh Runner Tugas terlebih dahulu, lalu memasangnya pada sumber daya komputasi Anda, menggunakan prosedur di bagian ini.

> **Note**
>
> Anda hanya dapat memasang Runner Tugas di Linux, UNIX, atau macOS. Runner Tugas tidak didukung pada sistem operasi Windows.
> Untuk menggunakan Task Runner 2.0, versi Java minimum yang dibutuhkan adalah 1.7.

Untuk menghubungkan Runner Tugas yang telah Anda pasang ke aktivitas alur yang harus diproses, tambahkan bidang `workerGroup` ke objek, dan konfigurasikan Runner Tugas untuk melakukan polling untuk nilai grup pekerja tersebut. Anda melakukan ini dengan meneruskan string grup pekerja sebagai parameter (misalnya, `--workerGroup=wg-12345`) saat Anda menjalankan file JAR Runner Tugas.




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

Pemasangan Runner Tugas

Bagian ini menjelaskan cara memasang dan mengonfigurasi Runner Tugas dan prasyaratnya. Pemasangan adalah proses manual yang mudah.

Untuk memasang Runner Tugas

1. Runner Tugas memerlukan Java versi 1.6 atau 1.8. Untuk menentukan apakah Java telah terpasang, dan versi yang sedang berjalan, gunakan perintah berikut:

   ```
   java -version
   ```

   Jika Anda tidak menginstal Java 1.6 atau 1.8 di komputer Anda, unduh salah satu versi ini dari http://www.oracle. com/technetwork/java/index.html. Unduh dan pasang Java, lalu lanjutkan ke langkah berikutnya.
2. Unduh `TaskRunner-1.0.jar` dari https://s3.amazonaws.com/datapipeline-us-east-1/us-east-1/software/latest/TaskRunner/TaskRunner -1.0.jar dan kemudian salin ke folder pada sumber daya komputasi target. Untuk klaster Amazon EMR yang menjalankan tugas `EmrActivity`, pasang Runner Tugas di simpul utama dari klaster.
3. Saat menggunakan Task Runner untuk terhubung ke layanan AWS Data Pipeline web untuk memproses perintah Anda, pengguna memerlukan akses terprogram ke peran yang memiliki izin untuk membuat atau mengelola pipeline data. Untuk informasi selengkapnya, lihat Memberikan akses terprogram.
4. Task Runner terhubung ke layanan AWS Data Pipeline web menggunakan HTTPS. Jika Anda menggunakan AWS sumber daya, pastikan HTTPS diaktifkan di tabel routing dan subnet ACL yang sesuai. Jika Anda menggunakan firewall atau proxy, pastikan port 443 terbuka.

(Opsional) Memberikan Akses Runner Tugas ke Amazon RDS

Amazon RDS memungkinkan Anda mengendalikan akses ke instans DB Anda menggunakan grup keamanan basis data (grup keamanan DB). Grup keamanan DB bertindak seperti firewall yang mengendalikan akses jaringan ke instans DB Anda. Secara default, akses jaringan dimatikan untuk instans DB Anda. Anda harus memodifikasi grup keamanan DB Anda agar Runner Tugas dapat mengakses instans Amazon RDS Anda. Runner Tugas mendapatkan akses Amazon RDS dari

instans yang dijalankannya, sehingga akun dan grup keamanan yang Anda tambahkan ke instans Amazon RDS bergantung pada tempat Anda menginstal Runner Tugas.

Untuk memberikan akses ke Task Runner di EC2 -Classic

1. Buka konsol Amazon RDS.
2. Di panel navigasi, pilih Instans, lalu pilih instans DB Anda.
3. Di bawah Keamanan dan Jaringan, pilih grup keamanan, yang membuka halaman Grup Keamanan dengan grup keamanan DB ini dipilih. Pilih ikon detail untuk grup keamanan DB.
4. Di bawah Detail Grup Keamanan, buat aturan dengan Tipe Koneksi dan Detail yang sesuai. Bidang ini tergantung pada di mana Runner Tugas berjalan, seperti yang dijelaskan di sini:

   - `Ec2Resource`
     - Tipe koneksi: `EC2 Security Group`

       Detail: *`my-security-group-name`* (nama grup keamanan yang Anda buat untuk EC2 instance)
   - `EmrResource`
     - Tipe koneksi: `EC2 Security Group`

       Rincian: `ElasticMapReduce-master`
     - Tipe koneksi: `EC2 Security Group`

       Rincian: `ElasticMapReduce-slave`
   - Lingkungan lokal Anda (on-premise)
     - Tipe koneksi: `CIDR/IP`:

       Detail: *`my-ip-address`* (alamat IP komputer Anda atau rentang alamat IP jaringan Anda, jika komputer Anda berada di belakang firewall)
5. Klik Tambahkan.

Untuk memberikan akses ke Task Runner di EC2 -VPC

1. Buka konsol Amazon RDS.
2. Di panel navigasi, pilih Instans.

3. Pilih ikon detail untuk instans DB. Di bawah Keamanan dan Jaringan, buka tautan ke grup keamanan, yang membawa Anda ke EC2 konsol Amazon. Jika Anda menggunakan desain konsol lama untuk grup keamanan, alihkan ke desain konsol baru dengan memilih ikon yang ditampilkan di bagian atas halaman konsol tersebut.
4. Pada tab Masuk, pilih Edit, Tambahkan Peraturan. Tentukan port basis data yang Anda gunakan saat meluncurkan instans DB. Sumbernya bergantung pada tempat Runner Tugas dijalankan, seperti yang dijelaskan di sini:

   - `Ec2Resource`
     - *`my-security-group-id`*(ID grup keamanan yang Anda buat untuk EC2 instance)
   - `EmrResource`
     - *`master-security-group-id`*(ID grup `ElasticMapReduce-master` keamanan)
     - *`slave-security-group-id`*(ID grup `ElasticMapReduce-slave` keamanan)
   - Lingkungan lokal Anda (on-premise)
     - *`ip-address`*(alamat IP komputer Anda atau rentang alamat IP jaringan Anda, jika komputer Anda berada di belakang firewall)
5. Klik Simpan.

Memulai Runner Tugas

Di jendela prompt perintah baru yang diatur ke direktori tempat Anda memasang Runner Tugas, mulai Runner Tugas dengan perintah berikut.

```
java -jar TaskRunner-1.0.jar --config ~/credentials.json --workerGroup=myWorkerGroup --
region=MyRegion --logUri=s3://amzn-s3-demo-bucket/foldername
```

Opsi `--config` menunjuk ke file kredensial Anda.

Opsi `--workerGroup` menentukan nama grup pekerja Anda, yang harus memiliki nilai yang sama seperti yang ditentukan dalam alur Anda agar tugas dapat diproses.

Opsi `--region` menentukan wilayah layanan tempat menarik tugas untuk dieksekusi.

Opsi `--logUri` digunakan untuk mendorong log terkompresi Anda ke lokasi di Amazon S3.

Saat Runner Tugas aktif, ia mencetak jalur ke tempat berkas log ditulis di jendela terminal. Berikut adalah contohnya.

```
Logging to /Computer_Name/.../output/logs
```

Runner Tugas harus dijalankan terlepas dari shell login Anda. Jika Anda menggunakan aplikasi terminal untuk terhubung ke komputer Anda, Anda mungkin perlu menggunakan utilitas seperti nohup atau layar untuk mencegah aplikasi Runner Tugas keluar saat Anda log out. Untuk informasi selengkapnya tentang opsi baris perintah, lihat Opsi Konfigurasi Runner Tugas.

Memverifikasi Pencatatan Runner Tugas

Cara termudah untuk memverifikasi bahwa Runner Tugas berfungsi adalah dengan memeriksa apakah ia menulis berkas log. Runner Tugas menulis berkas log per jam ke direktori, `output/logs`, di bawah direktori tempat Runner Tugas dipasang. Nama filenya adalah `Task Runner.log.YYYY-MM-DD-HH`, di mana HH berjalan dari 00 hingga 23, dalam UDT. Untuk menghemat ruang penyimpanan, file log apa pun yang lebih tua dari delapan jam dikompresi. GZip

Thread dan Prasyarat Runner Tugas

Runner Tugas menggunakan kolam thread untuk setiap tugas, aktivitas, dan prasyarat. Pengaturan default untuk `--tasks` adalah 2, artinya ada dua utas yang dialokasikan dari kumpulan tugas dan setiap utas melakukan polling AWS Data Pipeline layanan untuk tugas baru. Dengan demikian, `--tasks` adalah atribut penyetelan performa yang dapat digunakan untuk membantu mengoptimalkan throughput alur.

Logika coba ulang alur untuk prasyarat terjadi di Runner Tugas. Dua utas prasyarat dialokasikan untuk polling AWS Data Pipeline untuk objek prasyarat. Runner Tugas menghormati objek prasyarat retryDelay dan bidang preconditionTimeout yang Anda tetapkan pada prasyarat.

Dalam banyak kasus, mengurangi batas waktu polling prasyarat dan jumlah percobaan ulang membantu meningkatkan performa aplikasi Anda. Demikian pula, aplikasi dengan prasyarat yang berjalan lama mungkin perlu meningkatkan nilai batas waktu dan percobaan lagi. Untuk informasi selengkapnya tentang objek prasyarat, lihat Prasyarat.

Opsi Konfigurasi Runner Tugas

Ini adalah opsi konfigurasi yang tersedia dari baris perintah saat Anda meluncurkan Runner Tugas.

Parameter Baris Perintah	Deskripsi
`--help`	Bantuan baris perintah. Contoh: `Java -jar TaskRunner-1.0.jar --help`
`--config`	Jalur dan nama file dari file `credentials.json` Anda.
`--accessId`	ID kunci AWS akses Anda untuk Task Runner untuk digunakan saat membuat permintaan. Opsi `--accessID` dan `--secretKey` memberikan alternatif untuk menggunakan file kredensial.json. Jika file `credentials.json` juga disediakan, opsi `--accessID` dan `--secretKey` akan diutamakan.
`--secretKey`	Kunci AWS rahasia Anda untuk Task Runner untuk digunakan saat membuat permintaan. Untuk informasi selengkapnya, lihat `--accessID` .
`--endpoint`	Titik akhir adalah URL yang merupakan titik masuk untuk layanan web. Titik akhir AWS Data Pipeline layanan di wilayah tempat Anda membuat permintaan. Tidak wajib. Secara umum, cukup untuk menentukan wilayah, dan Anda tidak perlu mengatur titik akhir. Untuk daftar AWS Data Pipeline wilayah dan titik akhir, lihat Wilayah dan Titik Akhir AWS Data Pipeline di bagian. Referensi Umum AWS
`--workerGroup`	Nama grup pekerja tempat Runner Tugas mengambil pekerjaannya. Wajib. Saat Runner Tugas mensurvei layanan web, ia menggunakan kredensial yang Anda berikan dan nilai `workerGroup` untuk memilih tugas

Parameter Baris Perintah	Deskripsi
	mana (jika ada) yang akan diambil. Anda dapat menggunakan nama apa pun yang berarti bagi Anda; satu-satunya persyaratan adalah bahwa string harus cocok antara Runner Tugas dan aktivitas alur yang sesuai. Nama grup pekerja terikat ke suatu wilayah. Bahkan jika ada nama grup pekerja yang identik di wilayah lain, Runner Tugas selalu mendapatkan tugas dari wilayah yang ditentukan di `--region`.
`--taskrunnerId`	ID runner tugas yang akan digunakan saat melaporkan kemajuan. Tidak wajib.
`--output`	Direktori Runner Tugas untuk file output log. Tidak wajib. Berkas log disimpan dalam direktori lokal hingga didorong ke Amazon S3. Opsi ini menimpa direktori default.
`--region`	Wilayah yang akan digunakan. Opsional, tetapi direkomendasikan untuk selalu mengatur wilayah. Jika Anda tidak menentukan wilayah, Runner Tugas mengambil tugas dari wilayah layanan default, `us-east-1` . Wilayah lain yang didukung adalah: `eu-west-1` , `ap-northeast-1` , `ap-southeast-2` , `us-west-2` .
`--logUri`	Jalur tujuan Amazon S3 untuk Runner Tugas untuk mencadangkan berkas log setiap jam. Saat Runner Tugas berakhir, log aktif di direktori lokal didorong ke folder tujuan Amazon S3.
`--proxyHost`	Host proxy yang digunakan oleh klien Runner Tugas untuk terhubung ke layanan AWS.

Parameter Baris Perintah	Deskripsi
`--proxyPort`	Port host proxy yang digunakan oleh klien Runner Tugas untuk terhubung ke layanan AWS.
`--proxyUsername`	Nama pengguna untuk proxy.
`--proxyPassword`	Kata sandi untuk proxy.
`--proxyDomain`	Nama domain Windows untuk NTLM Proxy.
`--proxyWorkstation`	Nama workstation Windows untuk NTLM Proxy.

Menggunakan Runner Tugas dengan Proxy

Jika Anda menggunakan host proxy, Anda dapat menentukan konfigurasinya saat menjalankan Runner Tugas atau mengatur variabel lingkungan, HTTPS_PROXY. Variabel lingkungan yang digunakan dengan Runner Tugas menerima konfigurasi yang sama yang digunakan untuk Antarmuka Baris Perintah AWS.

Pelari Tugas dan Kustom AMIs

Saat Anda menentukan `Ec2Resource` objek untuk pipeline, buat AWS Data Pipeline EC2 instance untuk Anda, menggunakan AMI yang menginstal dan mengonfigurasi Task Runner untuk Anda. Tipe instans yang kompatibel dengan PV diperlukan dalam kasus ini. Atau, Anda dapat membuat AMI kustom dengan Runner Tugas, lalu menentukan ID AMI ini menggunakan bidang `imageId` objek `Ec2Resource`. Untuk informasi selengkapnya, lihat Ec2Resource.

AMI kustom harus memenuhi persyaratan berikut agar berhasil menggunakannya AWS Data Pipeline untuk Task Runner:

- Buat AMI di wilayah yang sama tempat instans akan dijalankan. Untuk informasi selengkapnya, lihat Membuat AMI Anda Sendiri di Panduan EC2 Pengguna Amazon.
- Pastikan bahwa jenis virtualisasi AMI didukung oleh tipe instans yang Anda rencanakan untuk digunakan. Misalnya, tipe instans I2 dan G2 memerlukan AMI HVM dan tipe instans T1, C1, M1, dan M2 memerlukan AMI PV. Untuk informasi selengkapnya, lihat Jenis Virtualisasi AMI Linux di Panduan EC2 Pengguna Amazon.

- Pasang perangkat lunak berikut:
  - Linux
  - Bash
  - wget
  - unzip
  - Java 1.6 atau 1.8
  - cloud-init
- Buat dan konfigurasikan pengguna bernama`ec2-user`.

Pemecahan Masalah

Ketika Anda memiliki masalah dengan AWS Data Pipeline, gejala yang paling umum adalah bahwa pipa tidak berjalan. Anda dapat menggunakan data yang disediakan oleh konsol tersebut dan CLI untuk mengidentifikasi masalah dan menemukan solusi.

Daftar Isi

- Menemukan Kesalahan dalam Alur
- Mengidentifikasi Klaster Amazon EMR yang Melayani Alur Anda
- Menafsirkan Detail Status Alur
- Menemukan Log Kesalahan
- Menyelesaikan Masalah Umum

Menemukan Kesalahan dalam Alur

AWS Data Pipeline Konsol adalah alat yang nyaman untuk memantau status saluran pipa Anda secara visual dan dengan mudah menemukan kesalahan apa pun yang terkait dengan proses pipa yang gagal atau tidak lengkap.

Untuk menemukan kesalahan tentang proses yang gagal atau tidak lengkap dengan konsol tersebut

1. Pada halaman Daftar Alur, jika kolom Status dari salah satu instans alur Anda menunjukkan status selain SELESAI, alur Anda menunggu beberapa prasyarat terpenuhi atau ia gagal dan Anda perlu memecahkan masalah alur tersebut.
2. Pada halaman Daftar Alur, cari instans alur dan pilih segitiga di sebelah kirinya, untuk memperluas detailnya.
3. Di bagian bawah panel ini, pilih Lihat detail eksekusi; panel Ringkasan instans terbuka untuk menampilkan detail instans yang terpilih.
4. Di panel Ringkasan Instans, pilih segitiga di sebelah instans untuk melihat detail tambahan dari instans, dan pilih Detail, Lagi... Jika status instans yang Anda pilih adalah GAGAL, kotak detail memiliki entri untuk pesan kesalahan, `errorStackTrace`, dan informasi lainnya. Anda dapat menyimpan informasi ini ke dalam file. Pilih OK.
5. Di panel Ringkasan instans, pilih Percobaan, untuk melihat detail setiap baris percobaan.

6. Untuk mengambil tindakan pada instans Anda yang tidak lengkap atau gagal, pilih kotak centang di sebelah instans. Ini mengaktifkan tindakan. Kemudian, pilih tindakan (`Rerun|Cancel|Mark Finished`).

Mengidentifikasi Klaster Amazon EMR yang Melayani Alur Anda

Jika ada `EMRCluster` atau `EMRActivity` gagal dan informasi kesalahan yang diberikan oleh AWS Data Pipeline konsol tidak jelas, Anda dapat mengidentifikasi klaster EMR Amazon yang melayani pipeline Anda menggunakan konsol Amazon EMR. Ini membantu Anda menemukan log yang disediakan Amazon EMR untuk mendapatkan detail selengkapnya tentang kesalahan yang terjadi.

Untuk melihat informasi kesalahan Amazon EMR yang lebih detail

1. Di AWS Data Pipeline konsol, pilih segitiga di sebelah instance pipeline, untuk memperluas detail instance.

2. Pilih Lihat detail eksekusi dan pilih segitiga di sebelah komponen.
3. Pada kolom Detail, pilih Lagi.... Layar informasi membuka daftar rincian komponen. Cari dan salin nilai instanceParent dari layar, seperti `@EmrActivityId_xiFDD_2017-09-30T21:40:13`:
4. Navigasikan ke konsol Amazon EMR, cari klaster dengan nilai instanceParent yang cocok dalam namanya, lalu pilih Debug.

   > **Note**
   >
   > Agar tombol Debug berfungsi, definisi pipeline Anda harus menyetel EmrActivity `enableDebugging` opsi ke `true` dan `EmrLogUri` opsi ke jalur yang valid.

5. Sekarang setelah Anda mengetahui klaster Amazon EMR mana yang berisi kesalahan yang menyebabkan kegagalan alur Anda, ikuti Tips Pemecahan Masalah di Panduan Developer Amazon EMR.

Menafsirkan Detail Status Alur

Berbagai tingkat status yang ditampilkan di AWS Data Pipeline konsol dan CLI menunjukkan kondisi pipa dan komponennya. Status alur hanyalah gambaran umum dari alur; untuk melihat informasi

lebih lanjut, lihat status masing-masing komponen alur. Anda dapat melakukannya dengan mengklik melalui alur di konsol tersebut atau mengambil detail komponen alur menggunakan CLI.

Kode Status

`ACTIVATING`

Komponen atau sumber daya sedang dimulai, seperti EC2 instance.

`CANCELED`

Komponen dibatalkan oleh pengguna atau AWS Data Pipeline sebelum dapat dijalankan. Hal ini dapat terjadi secara otomatis ketika terjadi kegagalan pada komponen atau sumber daya yang berbeda yang bergantung pada komponen ini.

`CASCADE_FAILED`

Komponen atau sumber daya dibatalkan sebagai akibat dari kegagalan kaskade dari salah satu dependensinya, tetapi komponen tersebut mungkin bukan sumber asli kegagalan.

`DEACTIVATING`

Alur sedang dinonaktifkan.

`FAILED`

Komponen atau sumber daya mengalami kesalahan dan berhenti bekerja. Ketika komponen atau sumber daya gagal, itu dapat menyebabkan pembatalan dan kegagalan untuk mengalir ke komponen lain yang bergantung padanya.

`FINISHED`

Komponen menyelesaikan pekerjaan yang ditugaskan.

`INACTIVE`

Alur dinonaktifkan.

`PAUSED`

Komponen dijeda dan saat ini tidak menjalankan tugasnya.

`PENDING`

Alur siap untuk diaktifkan untuk pertama kalinya.

`RUNNING`

Sumber daya sedang berjalan dan siap menerima pekerjaan.

`SCHEDULED`

Sumber daya dijadwalkan untuk berjalan.

`SHUTTING_DOWN`

Sumber daya dimatikan setelah berhasil menyelesaikan pekerjaannya.

`SKIPPED`

Komponen melewatkan interval eksekusi setelah alur diaktifkan menggunakan stempel waktu yang lebih lambat dari jadwal saat ini.

`TIMEDOUT`

Sumber daya melebihi `terminateAfter` ambang batas dan dihentikan. AWS Data Pipeline Setelah sumber daya mencapai status ini, AWS Data Pipeline mengabaikan nilai `actionOnResourceFailure`, `retryDelay`, dan `retryTimeout` untuk sumber daya tersebut. Status ini hanya berlaku untuk sumber daya.

`VALIDATING`

Definisi pipeline sedang divalidasi oleh AWS Data Pipeline.

`WAITING_FOR_RUNNER`

Komponen sedang menunggu klien pekerjanya untuk mengambil item pekerjaan. Hubungan klien komponen dan pekerja dikendalikan oleh bidang `runsOn` atau `workerGroup` yang ditentukan oleh komponen tersebut.

`WAITING_ON_DEPENDENCIES`

Komponen sedang memverifikasi bahwa prakondisi default dan yang dikonfigurasi pengguna terpenuhi sebelum melakukan pekerjaannya.

Menemukan Log Kesalahan

Bagian ini menjelaskan cara menemukan berbagai log yang AWS Data Pipeline menulis, yang dapat Anda gunakan untuk menentukan sumber kegagalan dan kesalahan tertentu.

Log Alur

Kami merekomendasikan Anda mengonfigurasi alur untuk membuat file log di lokasi persisten, seperti dalam contoh berikut di mana Anda menggunakan bidang `pipelineLogUri` pada objek `Default` alur untuk menyebabkan semua komponen alur menggunakan lokasi log Amazon S3

secara default (Anda dapat mengganti ini dengan mengonfigurasi lokasi log di komponen alur tertentu).

> **Note**
>
> Runner Tugas menyimpan log-nya di lokasi yang berbeda secara default, yang mungkin tidak tersedia saat alur selesai dan instans yang menjalankan Runner Tugas berakhir. Untuk informasi selengkapnya, lihat Memverifikasi Pencatatan Runner Tugas.

Untuk mengonfigurasi lokasi log menggunakan AWS Data Pipeline CLI dalam file JSON pipeline, mulailah file pipeline Anda dengan teks berikut:

```
{ "objects": [
{
  "id":"Default",
  "pipelineLogUri":"s3://amzn-s3-demo-bucket/error_logs"
},
...
```

Setelah Anda mengonfigurasi direktori log alur, Runner Tugas membuat salinan log di direktori Anda, dengan format dan nama file yang sama seperti yang dijelaskan di bagian sebelumnya tentang log Runner Tugas.

Tugas Hadoop dan Log Langkah Amazon EMR

Dengan aktivitas berbasis Hadoop sepertiHadoopActivity,HiveActivity, atau PigActivity Anda dapat melihat log pekerjaan Hadoop di lokasi yang dikembalikan di slot runtime,. hadoopJobLog EmrActivitymemiliki fitur logging sendiri dan log tersebut disimpan menggunakan lokasi yang dipilih oleh Amazon EMR dan dikembalikan oleh slot runtime,. emrStepLog Untuk informasi selengkapnya, lihat Lihat Berkas Log di Panduan Developer Amazon EMR.

Menyelesaikan Masalah Umum

Topik ini memberikan berbagai gejala AWS Data Pipeline masalah dan langkah-langkah yang disarankan untuk menyelesaikannya.

Daftar Isi

- Alur Terjebak dalam Status Tertunda

- Komponen Alur Terjebak dalam Menunggu Status Runner
- Komponen Alur Terjebak dalam Status WAITING_ON_DEPENDENCIES
- Jalankan Tidak Mulai Saat Dijadwalkan
- Komponen Alur Berjalan dalam Urutan yang Salah
- Klaster EMR Gagal Dengan Kesalahan: Token keamanan yang disertakan dalam permintaan tidak valid
- Izin Tidak Memadai untuk Mengakses Sumber Daya
- Kode Status: 400 Kode Kesalahan: PipelineNotFoundException
- Membuat Alur Menyebabkan Kesalahan Token Keamanan
- Tidak Dapat Melihat Detail Alur di Konsol Tersebut
- Kesalahan dalam Kode Status runner jarak jauh: 404, Layanan AWS: Amazon S3
- Akses Ditolak - Tidak Diotorisasi untuk Melakukan Fungsi datapipeline:
- Amazon EMR Lama AMIs Dapat Membuat Data Palsu untuk File CSV Besar
- Meningkatkan AWS Data Pipeline Batas

Alur Terjebak dalam Status Tertunda

Alur yang terlihat macet dalam status PENDING menunjukkan bahwa alur belum diaktifkan, atau aktivasi gagal karena kesalahan dalam definisi alur. Pastikan Anda tidak menerima kesalahan apa pun saat mengirimkan pipeline menggunakan AWS Data Pipeline CLI atau saat mencoba menyimpan atau mengaktifkan pipeline menggunakan konsol. AWS Data Pipeline Selain itu, periksa apakah alur Anda memiliki definisi yang valid.

Untuk melihat definisi alur Anda di layar menggunakan CLI:

```
aws datapipeline --get-pipeline-definition --pipeline-id df-EXAMPLE_PIPELINE_ID
```

Pastikan definisi alur selesai, periksa tanda kurung Anda, verifikasi koma yang diperlukan, periksa referensi yang hilang, dan kesalahan sintaks lainnya. Lebih baik menggunakan editor teks yang secara visual dapat memvalidasi sintaks file JSON.

Komponen Alur Terjebak dalam Menunggu Status Runner

Jika alur Anda dalam status TERJADWAL dan satu atau beberapa tugas tampak macet dalam status WAITING_FOR_RUNNER, pastikan Anda mengatur nilai yang valid untuk bidang runOn atau

workerGroup untuk tugas tersebut. Jika kedua nilai kosong atau hilang, tugas tidak dapat dimulai karena tidak ada asosiasi antara tugas dan pekerja untuk melakukan tugas. Dalam situasi ini, Anda telah mendefinisikan pekerjaan tetapi belum menentukan komputer apa yang melakukan pekerjaan. Jika dapat diaplikasikan, verifikasi bahwa nilai workerGroup yang ditetapkan ke komponen alur adalah nama dan kasus yang sama persis dengan nilai workerGroup yang Anda konfigurasi untuk Runner Tugas.

> **Note**
>
> Jika Anda memberikan nilai runsOn dan workerGroup ada, workerGroup diabaikan.

Penyebab potensial lain dari masalah ini adalah bahwa titik akhir dan kunci akses yang disediakan untuk Task Runner tidak sama dengan AWS Data Pipeline konsol atau komputer tempat alat AWS Data Pipeline CLI diinstal. Anda mungkin telah membuat alur baru tanpa kesalahan yang terlihat, tetapi Runner Tugas melakukan polling lokasi yang salah karena perbedaan kredensial, atau polling lokasi yang benar dengan izin yang tidak memadai untuk mengidentifikasi dan menjalankan pekerjaan yang ditentukan oleh definisi alur.

Komponen Alur Terjebak dalam Status WAITING_ON_DEPENDENCIES

Jika alur Anda berada dalam status `SCHEDULED` dan satu atau beberapa tugas terlihat macet dalam status `WAITING_ON_DEPENDENCIES`, pastikan prasyarat awal alur Anda telah terpenuhi. Jika prasyarat objek pertama dalam rantai logika tidak terpenuhi, tidak ada objek yang bergantung pada objek pertama tersebut yang dapat keluar dari status `WAITING_ON_DEPENDENCIES`.

Sebagai contoh, perhatikan kutipan berikut dari definisi alur. Dalam hal ini, `InputData` objek memiliki prasyarat 'Siap' yang menentukan bahwa data harus ada sebelum InputData objek selesai. Jika data tidak ada, InputData objek tetap dalam `WAITING_ON_DEPENDENCIES` keadaan, menunggu data yang ditentukan oleh bidang jalur tersedia. Benda apa pun yang bergantung pada InputData juga tetap dalam `WAITING_ON_DEPENDENCIES` keadaan menunggu InputData objek mencapai `FINISHED` negara.

```
{
    "id": "InputData",
    "type": "S3DataNode",
    "filePath": "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
    "schedule":{"ref":"MySchedule"},
    "precondition": "Ready"
```

```
},
{
    "id": "Ready",
    "type": "Exists"
...
```

Juga, periksa apakah objek Anda memiliki izin yang tepat untuk mengakses data. Dalam contoh sebelumnya, jika informasi di bidang kredensyal tidak memiliki izin untuk mengakses data yang ditentukan di bidang jalur, InputData objek akan macet dalam `WAITING_ON_DEPENDENCIES` keadaan karena tidak dapat mengakses data yang ditentukan oleh bidang jalur, bahkan jika data itu ada.

Mungkin juga sumber daya yang berkomunikasi dengan Amazon S3 tidak memiliki alamat IP publik yang terkait dengannya. Misalnya, `Ec2Resource` dalam subnet publik harus memiliki alamat IP publik yang terasosiasi dengannya.

Terakhir, dalam kondisi tertentu, instans sumber daya dapat mencapai status `WAITING_ON_DEPENDENCIES` jauh lebih awal daripada aktivitas terkait yang dijadwalkan untuk dimulai, yang mungkin memberi kesan bahwa sumber daya atau aktivitas tersebut gagal.

Jalankan Tidak Mulai Saat Dijadwalkan

Periksa apakah Anda memilih jenis jadwal yang benar yang menentukan apakah tugas Anda dimulai pada awal interval jadwal (Tipe Jadwal Gaya Cron) atau di akhir interval jadwal (Tipe Jadwal Deret Waktu).

Selain itu, periksa apakah Anda telah menentukan tanggal dengan benar dalam objek jadwal Anda startDateTime dan bahwa endDateTime nilai dan dalam format UTC, seperti dalam contoh berikut:

```
{
    "id": "MySchedule",
    "startDateTime": "2012-11-12T19:30:00",
    "endDateTime":"2012-11-12T20:30:00",
    "period": "1 Hour",
    "type": "Schedule"
},
```

Komponen Alur Berjalan dalam Urutan yang Salah

Anda mungkin memperhatikan bahwa waktu mulai dan berakhir untuk komponen alur Anda berjalan dalam urutan yang salah, atau dalam urutan yang berbeda dari yang Anda harapkan. Penting

untuk dipahami bahwa komponen alur dapat mulai berjalan secara bersamaan jika prasyaratnya terpenuhi pada saat memulai. Dengan kata lain, komponen alur tidak dijalankan secara berurutan secara default; jika Anda memerlukan perintah eksekusi tertentu, Anda harus mengendalikan urutan eksekusi dengan prasyarat dan bidang `dependsOn`.

Verifikasi bahwa Anda menggunakan bidang `dependsOn` yang diisi dengan referensi ke komponen alur prasyarat yang benar, dan bahwa semua petunjuk yang diperlukan antara komponen ada untuk mencapai urutan yang Anda butuhkan.

Klaster EMR Gagal Dengan Kesalahan: Token keamanan yang disertakan dalam permintaan tidak valid

Verifikasi peran, kebijakan, dan hubungan kepercayaan IAM Anda seperti yang dijelaskan di Peran IAM untuk AWS Data Pipeline.

Izin Tidak Memadai untuk Mengakses Sumber Daya

Izin yang Anda tetapkan pada peran IAM menentukan apakah AWS Data Pipeline dapat mengakses kluster dan EC2 instans EMR untuk menjalankan pipeline Anda. Selain itu, IAM memberikan konsep hubungan kepercayaan yang lebih jauh untuk mengizinkan pembuatan sumber daya atas nama Anda. Misalnya, saat Anda membuat pipeline yang menggunakan EC2 instance untuk menjalankan perintah untuk memindahkan data, AWS Data Pipeline dapat menyediakan EC2 instance ini untuk Anda. Jika Anda mengalami masalah, terutama yang melibatkan sumber daya yang dapat Anda akses secara manual tetapi AWS Data Pipeline tidak dapat, verifikasi peran, kebijakan, dan hubungan kepercayaan IAM Anda seperti yang dijelaskan dalamPeran IAM untuk AWS Data Pipeline.

Kode Status: 400 Kode Kesalahan: PipelineNotFoundException

Kesalahan ini berarti bahwa peran default IAM Anda mungkin tidak memiliki izin yang diperlukan AWS Data Pipeline untuk berfungsi dengan benar. Untuk informasi selengkapnya, lihat Peran IAM untuk AWS Data Pipeline.

Membuat Alur Menyebabkan Kesalahan Token Keamanan

Anda menerima kesalahan berikut ini saat mencoba membuat alur:

Gagal membuat alur dengan 'pipeline_name'. Kesalahan: UnrecognizedClientException - Token keamanan yang disertakan dalam permintaan tidak valid.

Tidak Dapat Melihat Detail Alur di Konsol Tersebut

Filter pipa AWS Data Pipeline konsol berlaku untuk tanggal mulai yang dijadwalkan untuk pipa, tanpa memperhatikan kapan pipa dikirimkan. Dimungkinkan untuk mengirimkan alur baru menggunakan tanggal mulai terjadwal yang terjadi di masa lalu, yang mungkin tidak ditampilkan oleh filter tanggal default. Untuk melihat detail alur, ubah filter tanggal Anda untuk memastikan tanggal mulai alur yang dijadwalkan sesuai dengan filter rentang tanggal.

Kesalahan dalam Kode Status runner jarak jauh: 404, Layanan AWS: Amazon S3

Kesalahan ini berarti bahwa Runner Tugas tidak dapat mengakses file Anda di Amazon S3. Verifikasi bahwa:

- Anda memiliki kredensial yang diatur dengan benar
- Bucket Amazon S3 yang Anda coba akses ada
- Anda diberi otorisasi untuk mengakses bucket Amazon S3

Akses Ditolak - Tidak Diotorisasi untuk Melakukan Fungsi datapipeline:

Di log Runner Tugas, Anda mungkin melihat kesalahan yang mirip dengan berikut ini:

- Kode Status ERROR: 403
- Layanan AWS: DataPipeline
- Kode Kesalahan AWS: AccessDenied
- Pesan Kesalahan AWS: Pengguna: arn:aws:sts: :xxxxxxxxxxxx:federated-user/i-xxxxxxxx tidak diizinkan untuk melakukan: datapipeline:. PollForTask

> **Note**
>
> Dalam pesan kesalahan ini, PollForTask dapat diganti dengan nama AWS Data Pipeline izin lainnya.

Pesan galat ini menunjukkan bahwa peran IAM yang Anda tentukan memerlukan izin tambahan yang diperlukan untuk berinteraksi. AWS Data Pipeline Pastikan kebijakan peran IAM Anda berisi baris

berikut, yang diganti dengan nama izin yang ingin Anda tambahkan (gunakan* untuk memberikan semua izin). PollForTask Untuk informasi selengkapnya tentang cara membuat IAM role baru dan menerapkan kebijakan padanya, lihat Mengelola Kebijakan IAM di panduan Menggunakan IAM.

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

Amazon EMR Lama AMIs Dapat Membuat Data Palsu untuk File CSV Besar

Di Amazon EMR AMIs sebelum 3,9 (3,8 dan di bawah) AWS Data Pipeline menggunakan kustom InputFormat untuk membaca dan menulis file CSV untuk digunakan dengan pekerjaan. MapReduce Ini digunakan saat layanan menyusun tabel ke dan dari Amazon S3. Masalah dengan ini InputFormat ditemukan di mana membaca catatan dari file CSV besar dapat menghasilkan tabel yang tidak disalin dengan benar. Masalah ini telah diperbaiki di rilis Amazon EMR selanjutnya. Harap gunakan Amazon EMR AMI 3.9 atau rilis Amazon EMR 4.0.0 atau yang lebih baru.

Meningkatkan AWS Data Pipeline Batas

Terkadang, Anda dapat melebihi batas AWS Data Pipeline sistem tertentu. Misalnya, batas alur default adalah 20 alur dengan masing-masing 50 objek. Jika Anda menemukan bahwa Anda membutuhkan lebih banyak alur daripada batasnya, pertimbangkan untuk menggabungkan beberapa alur untuk membuat lebih sedikit alur dengan lebih banyak objek di masing-masing. Untuk informasi lebih lanjut tentang batas AWS Data Pipeline , lihat AWS Data Pipeline Batas. Namun, jika Anda tidak dapat mengatasi batas menggunakan teknik penggabungan alur, minta peningkatan kapasitas Anda menggunakan formulir ini: Peningkatan Batas Data Pipeline.

AWS Data Pipeline Batas

Untuk memastikan bahwa ada kapasitas untuk semua pengguna, berikan AWS Data Pipeline batasan pada sumber daya yang dapat Anda alokasikan dan tingkat di mana Anda dapat mengalokasikan sumber daya.

Daftar Isi

- Batasan Akun
- Batas Panggilan Layanan Web
- Pertimbangan Penskalaan

Batasan Akun

Batasan berikut berlaku untuk satu AWS akun. Jika Anda memerlukan kapasitas tambahan, Anda dapat menggunakan Formulir permintaan Pusat Dukungan Amazon Web Services untuk meningkatkan kapasitas Anda.

Atribut	Kuota	Dapat Disesuaikan
Jumlah alur	100	Ya
Jumlah objek per alur	100	Ya
Jumlah instans aktif per objek	5	Ya
Jumlah bidang per objek	50	Tidak
Jumlah UTF8 byte per nama bidang atau pengenal	256	Tidak
Jumlah UTF8 byte per bidang	10,240	Tidak

Atribut	Kuota	Dapat Disesuaikan
Jumlah UTF8 byte per objek	15.360 (termasuk nama bidang)	Tidak
Tingkat pembuatan instans dari sebuah objek	1 per 5 menit	Tidak
Coba lagi aktivitas alur	5 per tugas	Tidak
Penundaan minimum antara upaya coba lagi	2 menit	Tidak
Interval penjadwalan minimum	15 menit	Tidak
Jumlah maksimum roll-up ke dalam satu objek	32	Tidak
Jumlah maksimum EC2 instance per objek Ec2Resource	1	Tidak

Batas Panggilan Layanan Web

AWS Data Pipeline membatasi tingkat di mana Anda dapat memanggil API layanan web. Batasan ini juga berlaku untuk AWS Data Pipeline agen yang memanggil API layanan web atas nama Anda, seperti konsol, CLI, dan Task Runner.

Batasan berikut berlaku untuk satu AWS akun. Ini berarti total penggunaan pada akun, termasuk yang oleh pengguna, tidak dapat melebihi batas ini.

Tingkat ledakan memungkinkan Anda menyimpan panggilan layanan web selama periode tidak aktif dan menghabiskan mereka semua dalam waktu singkat. Misalnya, CreatePipeline memiliki tingkat

reguler satu panggilan setiap lima detik. Jika Anda tidak menelepon layanan selama 30 detik, Anda memiliki enam panggilan disimpan. Anda kemudian bisa memanggil layanan web enam kali dalam satu detik. Karena ini adalah di bawah batas meledak dan terus panggilan rata-rata Anda pada batas tarif reguler, panggilan Anda tidak terhalang.

Jika Anda melebihi batas tingkat dan batas meledak, panggilan layanan web Anda gagal dan mengembalikan pengecualian throttling. Implementasi default pekerja, Task Runner, secara otomatis mencoba API panggilan yang gagal dengan pelambatan throttling. Task Runner memiliki mundur sehingga upaya berikutnya untuk memanggil API terjadi pada interval semakin lama. Jika Anda menulis pekerja, sebaiknya Anda menerapkan logika coba lagi yang serupa.

Batasan ini diterapkan terhadap AWS akun individu.

API	Batas tarif reguler	Batas burst
ActivatePipeline	1 panggilan per detik	100 panggilan
CreatePipeline	1 panggilan per detik	100 panggilan
DeletePipeline	1 panggilan per detik	100 panggilan
DescribeObjects	2 panggilan per detik	100 panggilan
DescribePipelines	1 panggilan per detik	100 panggilan
GetPipelineDefinition	1 panggilan per detik	100 panggilan
PollForTask	2 panggilan per detik	100 panggilan
ListPipelines	1 panggilan per detik	100 panggilan
PutPipelineDefinition	1 panggilan per detik	100 panggilan
QueryObjects	2 panggilan per detik	100 panggilan
ReportTaskProgress	10 panggilan per detik	100 panggilan
SetTaskStatus	10 panggilan per detik	100 panggilan
SetStatus	1 panggilan per detik	100 panggilan

API	Batas tarif reguler	Batas burst
ReportTaskRunnerHeartbeat	1 panggilan per detik	100 panggilan
ValidatePipelineDefinition	1 panggilan per detik	100 panggilan

Pertimbangan Penskalaan

AWS Data Pipeline skala untuk mengakomodasi sejumlah besar tugas bersamaan dan Anda dapat mengonfigurasinya untuk secara otomatis membuat sumber daya yang diperlukan untuk menangani beban kerja yang besar. Sumber daya yang dibuat secara otomatis ini berada di bawah kendali Anda dan memperhitungkan batas sumber daya akun AWS Anda. Misalnya, jika Anda mengonfigurasi AWS Data Pipeline untuk secara otomatis membuat klaster EMR Amazon 20 simpul untuk memproses data dan akun AWS Anda memiliki batas instans EC2 yang disetel ke 20, Anda mungkin secara tidak sengaja menghabiskan sumber daya pengisian ulang yang tersedia. Sebagai hasilnya, pertimbangkan pembatasan sumber daya ini dalam desain Anda atau tingkatkan batas akun Anda dengan sesuai.

Jika Anda memerlukan kapasitas tambahan, Anda dapat menggunakan Formulir permintaan Pusat Dukungan Amazon Web Services untuk meningkatkan kapasitas Anda.

AWS Data Pipeline Sumber Daya

Berikut ini adalah sumber daya untuk membantu Anda menggunakan AWS Data Pipeline.

- Informasi Produk AWS Data Pipeline–Halaman web utama untuk informasi tentang AWS Data Pipeline.
- AWS Data Pipeline FAQ Teknis - Meliputi 20 pertanyaan teratas yang diajukan pengembang tentang produk ini.
- Catatan rilis – Memberikan gambaran umum tingkat tinggi tentang rilis terkini. Secara khusus, catatan tersebut mencatat fitur, koreksi, dan masalah yang diketahui baru.
- Forum Diskusi AWS Data Pipeline – Forum berbasis komunitas bagi developer untuk mendiskusikan pertanyaan teknis terkait Amazon Web Services.

- Kelas & Lokakarya - Tautan ke kursus berbasis peran dan khusus, selain laboratorium mandiri untuk membantu mempertajam keterampilan Anda AWS dan mendapatkan pengalaman praktis.
- AWS Pusat Pengembang — Jelajahi tutorial, unduh alat, dan pelajari tentang acara AWS pengembang.
- AWS Alat Pengembang — Tautan ke alat pengembang SDKs, toolkit IDE, dan alat baris perintah untuk mengembangkan dan mengelola AWS aplikasi.
- Memulai Pusat Sumber Daya — Pelajari cara menyiapkan Akun AWS, bergabung dengan AWS komunitas, dan meluncurkan aplikasi pertama Anda.
- Hands-On Tutorial - Ikuti step-by-step tutorial untuk meluncurkan aplikasi pertama Anda. AWS
- AWS Whitepaper — Tautan ke daftar lengkap AWS whitepaper teknis, yang mencakup topik-topik seperti arsitektur, keamanan, dan ekonomi dan ditulis oleh AWS Solutions Architects atau pakar teknis lainnya.
- AWS Support Pusat — Hub untuk membuat dan mengelola AWS Support kasus Anda. Juga termasuk tautan ke sumber daya bermanfaat lainnya, seperti forum, teknis FAQs, status kesehatan layanan, dan AWS Trusted Advisor.
- Support— Halaman web utama untuk informasi tentang Support, saluran dukungan respons cepat untuk membantu Anda membangun dan menjalankan aplikasi di cloud. one-on-one
- Hubungi Kami – Titik kontak pusat untuk pertanyaan tentang tandaihan AWS , akun, peristiwa, penyalahgunaan, dan masalah lainnya.

- AWS Ketentuan Situs — Informasi terperinci tentang hak cipta dan merek dagang kami; akun, lisensi, dan akses situs Anda; dan topik lainnya.

Riwayat Dokumen

Dokumentasi ini dikaitkan dengan versi 2012-10-29 dari. AWS Data Pipeline

Ubah	Deskripsi	Tanggal Rilis
AWS Data Pipeline tidak lagi tersedia untuk pelanggan baru	AWS Data Pipeline tidak lagi tersedia untuk pelanggan baru. Pelanggan yang sudah ada AWS Data Pipeline dapat terus menggunakan layanan seperti biasa. Pelajari selengkapnya	25 Juli 2025
Menambahkan dokumentasi untuk melakukan prosedur tertentu menggunakan AWS CLI. Prosedur terkait AWS Data Pipeline konsol yang dihapus.	Lihat informasi selengkapnya di Mengkloning Alur Anda, Melihat Log Alur, dan Buat pipeline dari template Data Pipeline menggunakan CLI.	26 Mei 2023
Menambahkan lebih banyak konten dan sampel untuk bermigrasi dari AWS Data Pipeline ke layanan alternatif lainnya.	Memperbarui topik untuk migrasi AWS Data Pipeline ke AWS Step Functions AWS Glue, atau Amazon MWAA dengan informasi lebih lanjut tentang setiap alternatif, pemetaan konsep antara layanan, dan sampel. Untuk informasi selengkapnya, lihat Migrasi beban kerja dari AWS Data Pipeline.	31 Maret 2023
Menambahkan informasi tentang AWS Data Pipeline dukungan IMDSv2.	AWS Data Pipeline mendukung IMDSv2 sumber daya Amazon EMR dan Amazon EC2 . Lihat informasi selengkapnya di Perlindungan Data di AWS Data Pipeline, EmrCluster, dan Ec2Resource.	16 Desember 2022
Menambahkan topik untuk bermigrasi dari AWS Data Pipeline	Sekarang ada AWS layanan lain yang menawarka n pengalaman integrasi data yang lebih baik kepada pelanggan. Anda dapat memigrasikan kasus	16 Desember 2022

Ubah	Deskripsi	Tanggal Rilis
ke layanan alternatif lainnya.	penggunaan umum AWS Data Pipeline ke salah satu AWS Glue, AWS Step Functions, atau Amazon MWAA. Untuk informasi selengkapnya, lihat Migrasi beban kerja dari AWS Data Pipeline.	
Memperbarui daftar instans EMR Amazon EC2 dan Amazon yang didukung. Memperbarui daftar IDs HVM (Hardware Virtual Machine) yang AMIs digunakan untuk instance.	Memperbarui daftar instans EMR Amazon EC2 dan Amazon yang didukung. Untuk informasi selengkapnya, lihat Tipe Instans yang Didukung untuk Aktivitas Kerja Alur. Memperbarui daftar IDs HVM (Hardware Virtual Machine) yang AMIs digunakan untuk instance. Untuk informasi selengkapnya, lihat Sintaksis dan cari `imageId`.	9 November 2018

Ubah	Deskripsi	Tanggal Rilis
Menambahkan konfigurasi untuk melampirkan volume Amazon EBS ke simpul klaster, dan untuk meluncurk an klaster Amazon EMR ke dalam subnet subnet privat.	Menambahkan opsi konfigurasi ke objek `EMRcluste r` . Anda dapat menggunakan opsi ini dalam alur yang menggunakan klaster Amazon EMR. Gunakan bidang `coreEbsConfiguration` , `masterEbsConfiguration` , dan `TaskEbsCo nfiguration` untuk mengonfigurasi lampiran volume Amazon EBS ke simpul tugas, utama, dan inti di klaster Amazon EMR. Untuk informasi selengkap nya, lihat Melampirkan volume EBS ke simpul klaster. Gunakan bidang `emrManagedMasterSe curityGroupId` , `emrManagedSlaveSec urityGroupId` , dan `ServiceAccessSecur ityGroupId` untuk mengonfigurasi klaster Amazon EMR di subnet privat. Untuk informasi selengkapnya, lihat Mengonfigurasi klaster Amazon EMR di subnet privat. Untuk informasi selengkapnya tentang sintaks `EMRcluster` , lihat EmrCluster.	19 April 2018
Menambahkan daftar instans EMR Amazon EC2 dan Amazon yang didukung.	Menambahkan daftar AWS Data Pipeline instance yang dibuat secara default, jika Anda tidak menentuka n jenis instance dalam definisi pipeline. Menambahk an daftar instans EMR Amazon EC2 dan Amazon yang didukung. Untuk informasi selengkapnya, lihat Tipe Instans yang Didukung untuk Aktivitas Kerja Alur.	22 Maret 2018
Menambahkan dukungan untuk alur Sesuai Permintaan.	• Menambahkan dukungan untuk alur Sesuai Permintaan, yang memungkinkan Anda menjalankan ulang alur dengan mengaktifkannya lagi.	22 Februari 2016

Ubah	Deskripsi	Tanggal Rilis
Dukungan tambahan untuk basis data RDS	• Menambahkan `rdsInstanceId` , `region`, dan `jdbcDriverJarUri` ke RdsDatabase. • Memperbarui `database` di SqlActivity untuk juga mendukung `RdsDatabase` .	17 Agustus 2015
Dukungan JDBC tambahan	• Memperbarui `database` di SqlActivity untuk juga mendukung `JdbcDatabase` . • Menambahkan `jdbcDriverJarUri` ke JdbcDatabase • Menambahkan `initTimeout` ke Ec2Resource dan EmrCluster. • Menambahkan `runAsUser` ke Ec2Resource.	7 Juli 2015
HadoopActivity, Availability Zone, dan Spot Support	• Menambahkan dukungan untuk mengirimkan pekerjaan paralel ke klaster Hadoop. Untuk informasi selengkapnya, lihat HadoopActivity. • Menambahkan kemampuan untuk meminta Instans Spot dengan Ec2Resource dan EmrCluster. • Menambahkan kemampuan untuk meluncurkan sumber daya `EmrCluster` di Availability Zone tertentu.	1 Juni 2015
Menonaktifkan alur	Menambahkan dukungan untuk menonaktifkan alur aktif. Untuk informasi selengkapnya, lihat Menonaktifkan Alur Anda.	7 April 2015
Templat dan konsol yang diperbarui	Menambahkan template baru. Memperbarui chapter Memulai untuk menggunakan ShellCommandActivity template Memulai dengan. Untuk informasi selengkapnya, lihat Buat pipeline dari template Data Pipeline menggunakan CLI.	25 November 2014

Ubah	Deskripsi	Tanggal Rilis
Dukungan VPC	Menambahkan dukungan untuk meluncurkan sumber daya ke virtual private cloud (VPC).	12 Maret 2014
Dukungan Wilayah	Menambahkan dukungan untuk beberapa wilayah layanan. Selain ituus-east-1 , AWS Data Pipeline didukung dalameu-west-1 ,ap-northeast-1 ,ap-southeast-2 , danus-west-2 .	20 Februari 2014
Dukungan Amazon Redshift	Menambahkan dukungan untuk Amazon Redshift di AWS Data Pipeline, termasuk template konsol baru (Salin ke Redshift) dan tutorial untuk mendemons trasikan template. Untuk informasi lebih lanjut, lihat Salin Data ke Amazon Redshift Menggunakan AWS Data Pipeline, RedshiftDataNode, RedshiftDatabase, dan RedshiftCopyActivity .	6 November 2013
PigActivity	Ditambahkan PigActivity, yang menyediakan dukungan asli untuk Babi. Untuk informasi selengkapnya, lihat PigActivity.	15 Oktober 2013
Templat konsol baru, aktivitas, dan format data	Menambahkan template konsol CrossRegion DynamoDB Copy baru, termasuk yang baru dan Dynamo HiveCopyActivity . DBExport DataFormat	21 Agustus 2013
Kegagalan dan tayangan ulang yang berulang	Menambahkan informasi tentang kegagalan AWS Data Pipeline cascading dan perilaku menjalankan kembali. Untuk informasi selengkapnya, lihat Kegagalan dan tayangan ulang yang berulang.	8 Agustus 2013
Video pemecahan masalah	Menambahkan video Pemecahan Masalah AWS Data Pipeline Dasar. Untuk informasi selengkapnya, lihat Pemecahan Masalah.	17 Juli 2013

Ubah	Deskripsi	Tanggal Rilis
Mengedit alur aktif	Menambahkan lebih banyak informasi tentang mengedit alur aktif dan menjalankan kembali komponen alur. Untuk informasi selengkapnya, lihat Mengedit Alur Anda.	17 Juli 2013
Menggunakan sumber daya di wilayah berbeda	Menambahkan lebih banyak informasi tentang penggunaan sumber daya di wilayah berbeda. Untuk informasi selengkapnya, lihat Menggunakan Alur dengan Sumber Daya di Beberapa Wilayah.	17 Juni 2013
Status WAITING_ON_DEPENDENCIES	Status CHECKING_PRECONDITIONS diubah menjadi WAITING_ON_DEPENDENCIES dan menambahkan bidang waktu aktif @waitingOn untuk objek alur.	20 Mei 2013
Format Dinamo DBData	Ditambahkan template Dynamo DBData Format.	23 April 2013
Memproses video Log Web dan dukungan Instans Spot	Memperkenalkan video “Memproses Log Web dengan AWS Data Pipeline, Amazon EMR, dan Hive,” dan dukungan Instans EC2 Spot Amazon.	21 Februari 2013
	Rilis awal Panduan AWS Data Pipeline Pengembang.	20 Desember 2012