aws

Guide du développeur

AWS Data Pipeline

Version de l'API 2012-10-29

AWS Data Pipeline: Guide du développeur

Table of Contents

Qu'est-ce que AWS Data Pipeline ?

> **Note**
>
> AWS Data Pipelinele service est en mode maintenance et aucune nouvelle fonctionnalité ni extension régionale n'est prévue. Pour en savoir plus et pour savoir comment migrer vos charges de travail existantes, consultezMigration des charges de travail depuis AWS Data Pipeline.

AWS Data Pipeline est un service web que vous pouvez utiliser pour automatiser le transfert et la transformation des données. Avec AWS Data Pipeline, vous pouvez définir des flux de travail pilotés par les données afin que les tâches soient dépendantes de l'aboutissement des tâches précédentes. Vous définissez les paramètres de vos transformations de données et AWS Data Pipeline applique la logique que vous avez configurée.

Les composants suivants d'AWS Data Pipeline s'associent pour gérer vos données :

- Une définition de pipeline spécifie la logique métier de la gestion de vos données. Pour plus d'informations, veuillez consulter Syntaxe du fichier de définition du pipeline.
- Un pipeline planifie et exécute des tâches en créant des instances Amazon EC2 pour effectuer les activités de travail définies. Vous chargez votre définition de pipeline dans le pipeline, puis activez le pipeline. Vous pouvez modifier la définition d'un pipeline en cours d'exécution et réactiver le pipeline pour qu'il prenne effet. Vous pouvez désactiver le pipeline, modifier une source de données, puis réactiver le pipeline. Lorsque vous n'avez plus besoin de votre pipeline, vous pouvez le supprimer.
- Task Runner recherche des tâches, puis les exécute. Par exemple, Task Runner pourrait copier des fichiers journaux vers Amazon S3 et lancer des clusters Amazon EMR. Task Runner est installé et s'exécute automatiquement sur les ressources créées par les définitions de votre pipeline. Vous pouvez créer une application Task Runner personnalisée ou utiliser l'application Task Runner fournie parAWS Data Pipeline. Pour plus d'informations, veuillez consulter Exécuteurs de tâches.

Par exemple, vous pouvez archiver AWS Data Pipeline les journaux de votre serveur Web dans Amazon Simple Storage Service (Amazon S3) chaque jour, puis exécuter un cluster Amazon EMR (Amazon EMR) hebdomadaire sur ces journaux pour générer des rapports de trafic. AWS Data

Pipelineplanifie les tâches quotidiennes pour copier les données et la tâche hebdomadaire pour lancer le cluster Amazon EMR. AWS Data Pipelinegarantit également qu'Amazon EMR attend le chargement des données du dernier jour sur Amazon S3 avant de commencer son analyse, même en cas de retard imprévu dans le chargement des journaux.

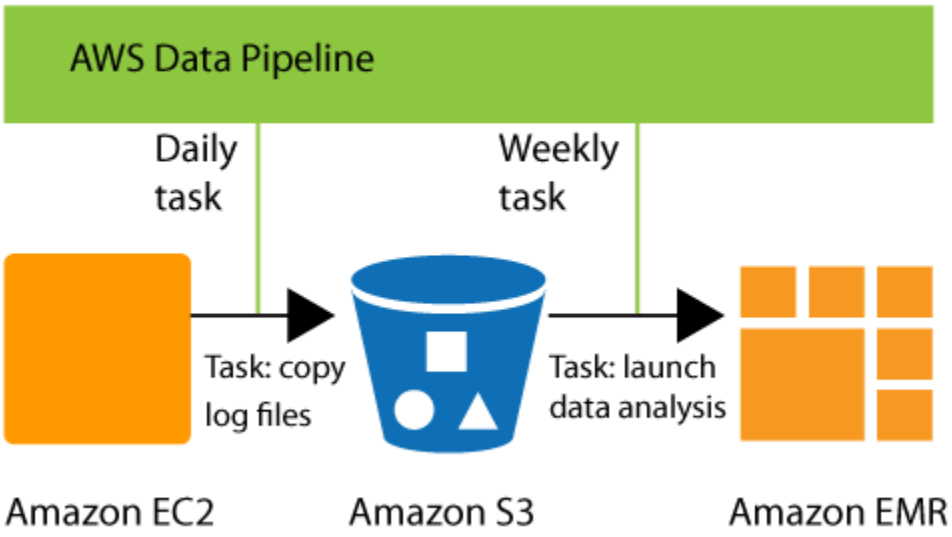


Table des matières

Migration des charges de travail depuis AWS Data Pipeline

AWSa lancé le AWS Data Pipeline service en 2012. À l'époque, les clients recherchaient un service qui les aiderait à déplacer des données de manière fiable entre différentes sources de données à l'aide de diverses options de calcul. Il existe désormais d'autres services qui offrent une meilleure expérience aux clients. Par exemple, vous pouvez utiliser AWS Glue to pour exécuter et orchestrer des applications Apache Spark, AWS Step Functions pour aider à orchestrer les composants du AWS service, ou Amazon Managed Workflows for Apache Airflow (Amazon MWAA) pour aider à gérer l'orchestration des flux de travail pour Apache Airflow.

Cette rubrique explique comment migrer AWS Data Pipeline vers d'autres options. L'option que vous choisissez dépend de votre charge de travail actuelle surAWS Data Pipeline. Vous pouvez migrer des cas d'AWS Data Pipelineutilisation classiques vers AWS Step Functions ou Amazon MWAA. AWS Glue

Migration des charges de travail vers AWS Glue

AWS Glueest un service d'intégration de données sans serveur qui facilite la découverte, la préparation, le déplacement et l'intégration de données provenant de plusieurs sources pour les utilisateurs d'analytique. Il inclut des outils pour la création, l'exécution de tâches et l'orchestration des flux de travail. avecAWS Glue, vous pouvez découvrir et vous connecter à plus de 70 sources de données diverses et gérer vos données dans un catalogue de données centralisé. Vous pouvez créer, exécuter et surveiller visuellement des pipelines d'extraction, de transformation et de chargement (ETL) pour charger les données dans vos lacs de données. Vous pouvez également rechercher et interroger immédiatement les données cataloguées à l'aide d'Amazon Athena, Amazon EMR et Amazon Redshift Spectrum.

Nous vous recommandons de migrer votre AWS Data Pipeline charge de travail AWS Glue lorsque :

- Vous recherchez un service d'intégration de données sans serveur prenant en charge diverses sources de données, des interfaces de création, notamment des éditeurs visuels et des blocs-notes, ainsi que des fonctionnalités avancées de gestion des données telles que la qualité des données et la détection des données sensibles.
- Votre charge de travail peut être migrée vers AWS Glue des flux de travail, des tâches (en Python ou Apache Spark) et des robots d'exploration (par exemple, votre pipeline existant est construit sur Apache Spark).
- Vous avez besoin d'une plateforme unique capable de gérer tous les aspects de votre pipeline de données, notamment l'ingestion, le traitement, le transfert, les tests d'intégrité et les contrôles de qualité.
- Votre pipeline existant a été créé à partir d'un modèle prédéfini sur la AWS Data Pipeline console, tel que l'exportation d'une table DynamoDB vers Amazon S3, et vous recherchez un modèle ayant le même objectif.
- Votre charge de travail ne dépend pas d'une application spécifique de l'écosystème Hadoop telle qu'Apache Hive.
- Votre charge de travail ne nécessite pas d'orchestrer des serveurs locaux.

AWSfacture un tarif horaire, facturé à la seconde, pour les robots d'exploration (découverte de données) et les tâches ETL (traitement et chargement de données). AWS Glue Studio est un moteur d'orchestration intégré pour les AWS Glue ressources et est proposé sans frais supplémentaires. Pour en savoir plus sur la tarification, consultez la section AWS GlueTarification.

Migration des charges de travail vers Step AWS Functions

AWSStep Functions est un service d'orchestration sans serveur qui vous permet de créer des flux de travail pour vos applications critiques. Avec Step Functions, vous utilisez un éditeur visuel pour créer des flux de travail et les intégrer directement à plus de 11 000 actions pour plus de 250 AWS services, tels qu'AWS Lambda, Amazon EMR, DynamoDB, etc. Vous pouvez utiliser Step Functions pour orchestrer les pipelines de traitement des données, gérer les erreurs et respecter les limites de limitation des services sous-jacents. AWS Vous pouvez créer des flux de travail qui traitent et publient des modèles d'apprentissage automatique, orchestrer des microservices, ainsi que des AWS services de contrôle, par exemple pour créer des flux de travail d'extraction, de transformation et de chargement (ETL). AWS Glue Vous pouvez également créer des flux de travail automatisés de longue durée pour les applications nécessitant une interaction humaine.

De mêmeAWS Data Pipeline, AWS Step Functions est un service entièrement géré fourni parAWS. Vous n'aurez pas à gérer l'infrastructure, à appliquer les correctifs, à gérer les mises à jour des versions du système d'exploitation ou à des tâches similaires.

Nous vous recommandons de migrer votre AWS Data Pipeline charge de travail vers AWS Step Functions lorsque :

- Vous recherchez un service d'orchestration de flux de travail sans serveur et hautement disponible.
- Vous recherchez une solution rentable qui facture en fonction de la granularité de l'exécution d'une seule tâche.
- Vos charges de travail orchestrent des tâches pour plusieurs autres AWS services, tels qu'Amazon EMR, Lambda ou DynamoDB. AWS Glue
- Vous recherchez une solution low-code qui intègre un concepteur drag-and-drop visuel pour la création de flux de travail et qui ne nécessite pas l'apprentissage de nouveaux concepts de programmation.
- Vous recherchez un service qui fournit des intégrations avec plus de 250 autres AWS services couvrant plus de 11 000 actionsout-of-the-box, tout en permettant des intégrations avec des activités et des AWS non-services personnalisés.

Step Functions AWS Data Pipeline et Step Functions utilisent le format JSON pour définir les flux de travail. Cela permet de stocker vos flux de travail dans le contrôle de source, de gérer les versions, de contrôler l'accès et d'automatiser avec CI/CD. Step Functions utilise une syntaxe appelée Amazon State Language qui est entièrement basée sur JSON et permet une transition fluide entre les représentations textuelles et visuelles du flux de travail.

Avec Step Functions, vous pouvez choisir la même version d'Amazon EMR que celle que vous utilisez actuellement. AWS Data Pipeline

Pour migrer des activités sur des ressources AWS Data Pipeline gérées, vous pouvez utiliser l'intégration des services du AWS SDK sur Step Functions pour automatiser le provisionnement et le nettoyage des ressources.

Pour migrer des activités sur des serveurs locaux, des instances EC2 gérées par l'utilisateur ou un cluster EMR géré par l'utilisateur, vous pouvez installer un agent SSM sur l'instance. Vous pouvez lancer la commande via la commande Exécuter de AWS Systems Manager à partir de Step Functions. Vous pouvez également lancer la machine d'état à partir du calendrier défini dans Amazon EventBridge.

AWSStep Functions propose deux types de flux de travail : les flux de travail standard et les flux de travail express. Pour les flux de travail standard, vous êtes facturé en fonction du nombre de transitions d'état requises pour exécuter votre application. Pour Express Workflows, vous êtes facturé en fonction du nombre de demandes pour votre flux de travail et de sa durée. Pour en savoir plus sur la tarification, consultez AWSStep Functions Pricing.

Migration des charges de travail vers Amazon MWAA

Amazon MWAA (Managed Workflows for Apache Airflow) est un service d'orchestration géré pour Apache Airflow qui facilite la configuration et l'exploitation de pipelines de données de bout en bout dans le cloud à grande échelle. Apache Airflow est un outil open source utilisé pour créer, planifier et surveiller par programmation des séquences de processus et de tâches appelées « workflows ». Avec Amazon MWAA, vous pouvez utiliser les langages de programmation Airflow et Python pour créer des flux de travail sans avoir à gérer l'infrastructure sous-jacente pour des raisons d'évolutivité, de disponibilité et de sécurité. Amazon MWAA adapte automatiquement sa capacité d'exécution des flux de travail pour répondre à vos besoins et est intégré aux services de AWS sécurité pour vous fournir un accès rapide et sécurisé à vos données.

De mêmeAWS Data Pipeline, Amazon MWAA est un service entièrement géré fourni parAWS. Bien que vous deviez apprendre plusieurs nouveaux concepts spécifiques à ces services, vous n'êtes pas

obligé de gérer l'infrastructure, les correctifs, les mises à jour des versions du système d'exploitation, etc.

Nous vous recommandons de migrer vos AWS Data Pipeline charges de travail vers Amazon MWAA lorsque :

- Vous recherchez un service géré à haute disponibilité pour orchestrer des flux de travail écrits en Python.
- Vous souhaitez passer à une technologie open source entièrement gérée et largement adoptée, Apache Airflow, pour une portabilité maximale.
- Vous avez besoin d'une plateforme unique capable de gérer tous les aspects de votre pipeline de données, notamment l'ingestion, le traitement, le transfert, les tests d'intégrité et les contrôles de qualité.
- Vous recherchez un service conçu pour l'orchestration du pipeline de données avec des fonctionnalités telles qu'une interface utilisateur riche pour l'observabilité, des redémarrages en cas d'échec des flux de travail, des remplissages et des nouvelles tentatives de tâches.
- Vous recherchez un service comprenant plus de 800 opérateurs et capteurs prédéfinis, couvrant AWS aussi bien des services que d'autres AWS services.

Les flux de travail Amazon MWAA sont définis comme des graphes acycliques dirigés (DAG) à l'aide de Python. Vous pouvez donc également les traiter comme du code source. Le framework Python extensible d'Airflow vous permet de créer des flux de travail connectés à pratiquement toutes les technologies. Il est doté d'une interface utilisateur riche permettant de visualiser et de surveiller les flux de travail et peut être facilement intégré aux systèmes de contrôle de version pour automatiser le processus CI/CD.

Avec Amazon MWAA, vous pouvez choisir la même version d'Amazon EMR que celle que vous utilisez actuellement. AWS Data Pipeline

AWSfacture en fonction du temps d'exécution de votre environnement Airflow, plus toute mise à l'échelle automatique supplémentaire visant à fournir davantage de capacité de travail ou de serveur Web. En savoir plus sur la tarification dans Amazon Managed Workflows for Apache Airflow Pricing.

Cartographie des concepts

Le tableau suivant contient le mappage des principaux concepts utilisés par les services. Il aidera les personnes familiarisées avec Data Pipeline à comprendre les fonctions Step et la terminologie de la MWAA.

Data Pipeline	Glue	Step Functions	Amazon MWA
Pipelines	Workflows	Workflows	Graphiques acryliques directs
Définition du pipeline JSON	Définition du flux de travail ou plans basés sur Python	JSON, langage d'État d'Amazon	Basé sur Python
Activités	Tâches	États et tâches	Tâches (opérateurs et capteurs)
instances	Exécutions de tâches	Exécutions	DAG fonctionne
Tentatives	Tentatives de nouvelle tentative	Catchers et retriers	Réessais
Calendrier du gazoduc	Déclencheurs de	EventBridgeTâches du planificateur	Cron, horaires, connaissance des données
Expressions et fonctions du pipeline	Bibliothèque de plans	Step Functions, fonctions intrinsèques et AWSLambda	Framework Python extensible

Exemples

Les sections suivantes répertorient des exemples publics auxquels vous pouvez vous référer pour effectuer la migration AWS Data Pipeline vers des services individuels. Vous pouvez vous y référer à titre d'exemples et créer votre propre pipeline sur les services individuels en le mettant à jour et en le testant en fonction de votre cas d'utilisation.

Exemples AWS Glue

La liste suivante contient des exemples d'implémentations pour les cas d'AWS Data Pipelineutilisation les plus courants avec. AWS Glue

- Exécution de tâches Spark

- Copier des données depuis JDBC vers Amazon S3 (y compris Amazon Redshift)
- Copier des données depuis Amazon S3 vers JDBC (y compris Amazon Redshift)
- Copier des données depuis Amazon S3 vers DynamoDB
- Déplacement de données vers et depuis Amazon Redshift
- Accès entre comptes et entre régions aux tables DynamoDB

AWSExemples de fonctions Step

La liste suivante contient des exemples d'implémentations pour les AWS Data Pipeline cas d'utilisation les plus courants avec AWS Step Functions.

- Gestion d'une tâche Amazon EMR
- Exécution d'une tâche de traitement de données sur Amazon EMR Serverless
- Exécution de tâches Hive/Pig/Hadoop
- Interrogation de grands ensembles de données (Amazon Athena, Amazon S3,) AWS Glue
- Exécution de flux de travail ETL à l'aide d'Amazon Redshift
- Orchestrer AWS Glue les robots d'exploration

Consultez des didacticiels supplémentaires et des exemples de projets sur l'utilisation de AWS Step Functions.

Échantillons Amazon MWAA

La liste suivante contient des exemples d'implémentations pour les AWS Data Pipeline cas d'utilisation les plus courants avec Amazon MWAA.

- Exécution d'une tâche Amazon EMR
- Création d'un plugin personnalisé pour Apache Hive et Hadoop
- Copier des données depuis Amazon S3 vers Redshift
- Exécution d'un script Shell sur une instance EC2 distante
- Orchestrer des flux de travail hybrides (sur site)

Consultez des didacticiels supplémentaires et des exemples de projets relatifs à l'utilisation d'Amazon MWAA.

Services connexes

AWS Data Pipeline s'associe aux services suivants pour stocker les données.

- Amazon DynamoDB : fournit une base de données NoSQL entièrement gérée offrant des performances rapides à moindre coût. Pour plus d'informations, consultez le guide du développeur Amazon DynamoDB.
- Amazon RDS : fournit une base de données relationnelle entièrement gérée qui s'adapte à de grands ensembles de données. Pour plus d'informations, consultez le guide du développeur Amazon Relational Database Service.
- Amazon Redshift : fournit un entrepôt de données rapide et entièrement géré d'une capacité de plusieurs pétaoctets qui permet d'analyser facilement et à moindre coût une grande quantité de données. Pour plus d'informations, consultez le guide du développeur de base de données Amazon Redshift.
- Amazon S3 : fournit un stockage d'objets sécurisé, durable et hautement évolutif. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon Simple Storage Service.

AWS Data Pipeline s'associe aux services de calcul suivants pour transformer les données.

- Amazon EC2 : fournit une capacité informatique redimensionnable (littéralement, des serveurs dans les centres de données d'Amazon) que vous utilisez pour créer et héberger vos systèmes logiciels. Pour plus d'informations, consultez le guide de l'utilisateur Amazon EC2 pour les instances Linux.
- Amazon EMR : vous permet de distribuer et de traiter facilement, rapidement et à moindre coût de grandes quantités de données sur les serveurs Amazon EC2, à l'aide d'un framework tel qu'Apache Hadoop ou Apache Spark. Pour plus d'informations, consultez le guide du développeur Amazon EMR.

Accès à AWS Data Pipeline

Vous pouvez créer vos pipelines, y accéder et les gérer à l'aide des interfaces suivantes :

- AWS Management Console— Fournit une interface Web à laquelle vous pouvez accéderAWS Data Pipeline.
- AWS Command Line Interface(AWS CLI) — Fournit des commandes pour un large éventail de services AWS, notamment WindowsAWS Data Pipeline, macOS et Linux, et est compatible avec

ces derniers. Pour plus d'informations sur l'installation de la AWS CLI, consultez le document AWS Command Line Interface. Pour voir la liste des commandes pour AWS Data Pipeline, consultez datapipeline.

- Kits de développement (SDK) AWS : fournissent des API propres au langage et se chargent de nombreux détails de connexion, tels que le calcul des signatures, la gestion des nouvelles tentatives de demande et la gestion des erreurs. Pour de plus amples informations, veuillez consulter SDK AWS.
- API de requête : fournit des API de bas niveau que vous appelez à l'aide de requêtes HTTPS. L'utilisation de l'API de demande est le moyen le plus direct d'accéder à AWS Data Pipeline, mais elle nécessite que votre application gère les détails de bas niveau, tels que la génération d'un hachage pour signer la demande et le traitement des erreurs. Pour plus d'informations, consultez la AWS Data Pipeline API Reference (Référence d'API).

Tarification

Avec Amazon Web Services, vous payez uniquement en fonction de votre utilisation. Pour AWS Data Pipeline, vous payez pour votre pipeline en fonction de l'environnement d'exécution et de la fréquence d'exécution planifiée de vos activités et conditions préalables. Pour plus d'informations, consultez AWS Data Pipeline Pricing (Tarification CTlong).

Si votre compte AWS a moins de 12 mois, vous pouvez bénéficier de l'offre gratuite. L'offre gratuite inclut trois conditions préalables à faible fréquence et cinq activités à faible fréquence gratuites par mois. Pour de plus amples informations, veuillez consulter Offre gratuite d'AWS.

Types d'instances prises en charge pour les activités de pipeline

Lorsqu'un pipeline est AWS Data Pipeline exécuté, il compile les composants du pipeline pour créer un ensemble d'instances Amazon EC2 exploitables. Chaque instance contient toutes les informations pour effectuer une tâche spécifique. L'ensemble complet d'instances correspond à la liste de tâches du pipeline. AWS Data Pipeline distribue les instances aux exécuteurs de tâches pour qu'ils les traitent.

Les instances EC2 se déclinent dans différentes configurations, également appelées types d'instance. Chaque type d'instance offre une capacité de CPU, d'entrée/sortie et de calcul différente. En plus de la spécification du type d'instance pour une activité, vous pouvez choisir différentes options d'achat. Tous les types d'instance ne sont pas disponibles dans toutes les régions AWS. Si

un type d'instance n'est pas disponible, votre pipeline risque de ne pas réussir la mise en service ou d'être bloqué pendant la mise en service. Pour plus d'informations sur la disponibilité des instances, consultez la page de tarification Amazon EC2. Ouvrez le lien correspondant à votre option d'achat d'instance et filtrez par Région pour voir si un type d'instance est disponible dans cette région. Pour plus d'informations sur ces types d'instances, ces familles et ces types de virtualisation, consultez Amazon EC2 Instances et Amazon Linux AMI Instances Type Matrix.

Les tableaux suivant décrivent les types d'instance pris en charge par AWS Data Pipeline. Vous pouvez l'utiliser AWS Data Pipeline pour lancer des instances Amazon EC2 dans n'importe quelle région, y compris les régions où ce service n'AWS Data Pipelineest pas pris en charge. Pour plus d'informations sur les régions dans lesquelles AWS Data Pipeline est pris en charge, consultez Régions et points de terminaison AWS.

Table des matières

Instances Amazon EC2 par défaut par région AWS

Si vous ne spécifiez pas de type d'instance dans votre définition de pipeline, AWS Data Pipeline lance une instance par défaut.

Le tableau suivant répertorie les instances Amazon EC2 AWS Data Pipeline utilisées par défaut dans les régions où il AWS Data Pipeline est pris en charge.

Nom de la région	Région	Type d'instance
US East (Virginie du Nord)	us-east-1	m1.small
USA Ouest (Oregon)	us-west-2	m1.small
Asie-Pacifique (Sydney)	ap-southeast-2	m1.small
Asie-Pacifique (Tokyo)	ap-northeast-1	m1.small
UE (Irlande)	eu-west-1	m1.small

Le tableau suivant répertorie les instances Amazon EC2 qui se AWS Data Pipeline lancent par défaut dans les régions où ce service n'AWS Data Pipelineest pas pris en charge.

Nom de la région	Région	Type d'instance
USA Est (Ohio)	us-east-2	t2.small
USA Ouest (Californie du Nord)	us-west-1	m1.small
Asie-Pacifique (Mumbai)	ap-south-1	t2.small
Asie-Pacifique (Singapour)	ap-southeast-1	m1.small
Asie-Pacifique (Séoul)	ap-northeast-2	t2.small
Canada (Centre)	ca-central-1	t2.small
UE (Francfort)	eu-central-1	t2.small
UE (Londres)	eu-west-2	t2.small
UE (Paris)	eu-west-3	t2.small
Amérique du Sud (São Paulo)	sa-east-1	m1.small

Instances Amazon EC2 supplémentaires prises en charge

Outre les instances par défaut qui sont créées si vous ne spécifiez pas de type d'instance dans votre définition de pipeline, les instances suivantes sont prises en charge.

Le tableau suivant répertorie les instances Amazon EC2 prises AWS Data Pipeline en charge et pouvant être créées, si cela est spécifié.

Classe d'instance	Types d'instances
Usage général	t2.nano | t2.micro | t2.small | t2.medium | t2.large
Calcul optimisé	c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge | c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge |

Classe d'instance	Types d'instances
	c5.xlarge | c5.9xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.18xlarge | c5d.xlarge | c5d.2xlarge | c5d.4xlarge | c5d.9xlarge | c5d.18xlarge
Mémoire optimisée	m3.medium | m3.large | m3.xlarge | m3.2xlarge | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge | m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge
Stockage optimisé	i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge | hs1.8xlarge | g2.2xlarge | g2.8xlarge | d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge

Instances Amazon EC2 prises en charge pour les clusters Amazon EMR

Ce tableau répertorie les instances Amazon EC2 qui prennent AWS Data Pipeline en charge et peuvent être créées pour les clusters Amazon EMR, si cela est spécifié. Pour plus d'informations, consultez la section Types d'instances pris en charge dans le Guide de gestion Amazon EMR.

Classe d'instance	Types d'instances
Usage général	m1.small | m1.medium | m1.large | m1.xlarge | m3.xlarge | m3.2xlarge
Calcul optimisé	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
Mémoire optimisée	m2.xlarge | m2.2xlarge | m2.4xlarge | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | cr1.8xlarge | m4.large | m4.xlarge |

Classe d'instance	Types d'instances
	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
Stockage optimisé	h1.4xlarge | hs1.2xlarge | hs1.4xlarge| hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4large | i2.8xlarge | d2.xlarge | d2.2xlarge| d2.4xlarge | d2.8xlarge
Calcul accéléré	g2.2xlarge | cg1.4xlarge

Concepts d'AWS Data Pipeline

Avant de commencer, prenez connaissance des concepts et composants clés d'AWS Data Pipeline.

Table des matières

Définition de pipeline

Une définition de pipeline est un moyen de communiquer votre logique métier à AWS Data Pipeline. Elle contient les informations suivantes :

- Noms, emplacements et formats de vos sources de données
- Activités qui transforment les données
- Planification de ces activités
- Ressources qui exécutent vos activités et conditions préalables
- Conditions préalables qui doivent être remplies pour que les activités puissent être planifiées
- Moyens de vous alerter avec des mises à jour de l'état au fur et à mesure de l'exécution du pipeline

A partir de votre définition de pipeline, AWS Data Pipeline détermine les tâches, les planifie et les affecte à des exécuteurs de tâches. Si une tâche n'est pas terminée avec succès, AWS Data Pipeline la relance conformément à vos instructions et, si nécessaire, la réaffecte à un autre exécuteur de tâches. Si la tâche échoue plusieurs fois, vous pouvez configurer le pipeline pour qu'il vous en informe.

Par exemple, dans la définition de votre pipeline, vous pouvez spécifier que les fichiers journaux générés par votre application sont archivés chaque mois en 2013 dans un compartiment Amazon S3. AWS Data Pipelinecréerait ensuite 12 tâches, chacune copiant l'équivalent de plus d'un mois de données, que le mois contienne 30, 31, 28 ou 29 jours.

Vous pouvez créer une définition de pipeline de différentes manières :

- Graphiquement, en utilisant la console AWS Data Pipeline
- Textuellement, en écrivant un fichier JSON au format utilisé par l'interface de ligne de commande
- Par programmation, en appelant le service web avec l'un des kits SDK AWS ou l'API AWS Data Pipeline

Une définition de pipeline peut contenir les types de composants suivants.

Composants de pipeline

Nœuds de données

Emplacement des données d'entrée pour une tâche ou emplacement où les données de sortie doivent être stockées.

Activités

Définition du travail à effectuer selon une planification donnée en utilisant une ressource de calcul et généralement des nœuds de données d'entrée et de sortie.

Conditions préalables

Instruction conditionnelle qui doit avoir la valeur true pour qu'une action puisse être exécutée.

Ressources

Ressource de calcul qui effectue le travail défini par un pipeline.

Actions

Action qui est déclenchée lorsque les conditions spécifiées sont remplies, par exemple, l'échec d'une activité.

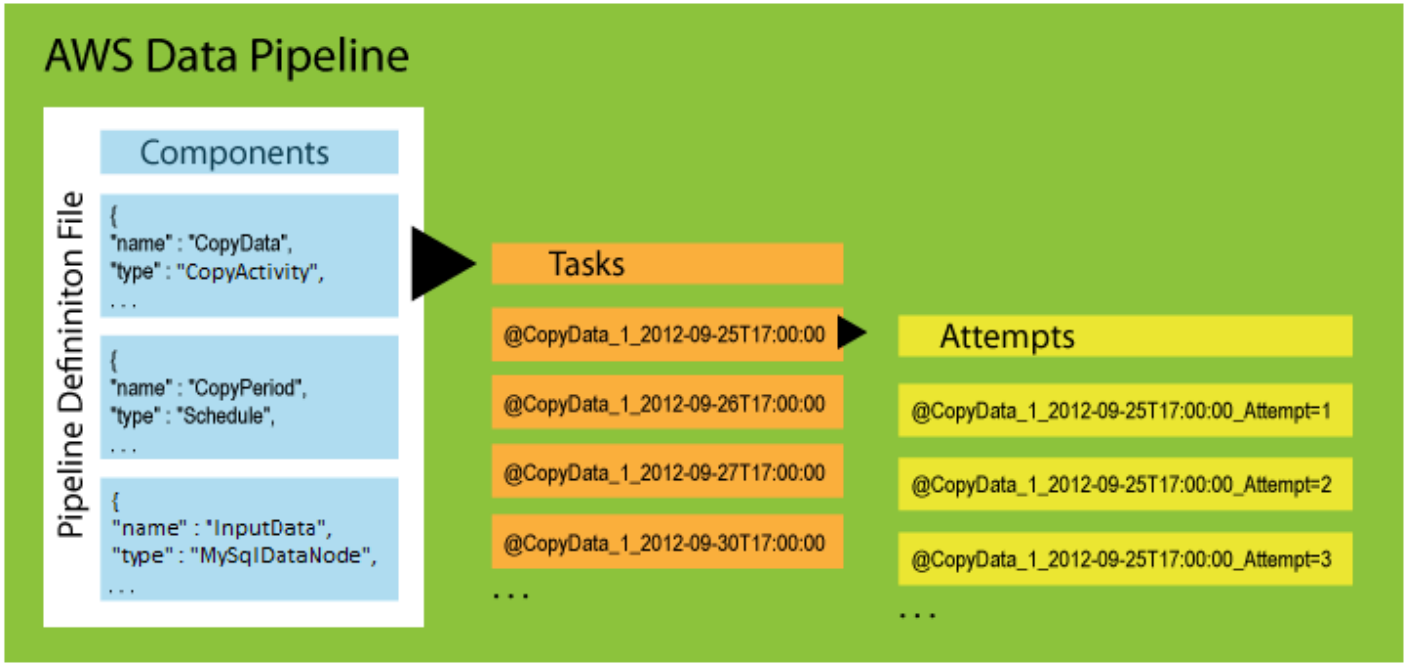
Pour plus d'informations, veuillez consulter Syntaxe du fichier de définition du pipeline.

Composants de pipeline, instances et tentatives

Trois types d'éléments sont associés à un pipeline planifié :

- Composants du pipeline : les composants du pipeline représentent la logique métier du pipeline et sont représentés par les différentes sections d'une définition du pipeline. Ils spécifient les sources de données, les activités, la planification et les conditions préalables du flux de travail. Ils peuvent hériter des propriétés de leurs composants parents. Les relations entre composants sont définies par une référence. Les composants de pipeline définissent les règles de gestion des données.
- Instances : lorsqu'AWS Data Pipeline exécute un pipeline, il compile les composants de pipeline pour créer un ensemble d'instances exploitables. Chaque instance contient toutes les informations pour effectuer une tâche spécifique. L'ensemble complet d'instances correspond à la liste de tâches du pipeline. AWS Data Pipeline distribue les instances aux exécuteurs de tâches pour qu'ils les traitent.
- Tentatives : pour assurer une gestion des données robuste, AWS Data Pipeline relance toute opération ayant échoué. Il continue jusqu'à ce que la tâche atteigne le nombre maximal de nouvelles tentatives autorisées. Les objets tentatives suivent les divers tentatives, résultats et motifs d'échec le cas échéant. Il s'agit essentiellement de l'instance dotée d'un compteur. AWS Data Pipelineeffectue de nouvelles tentatives en utilisant les mêmes ressources que lors des tentatives précédentes, telles que les clusters Amazon EMR et les instances EC2.

> **Note**
>
> La relance des tâches ayant échoué est une partie importante de toute stratégie de tolérance aux pannes, et les définitions AWS Data Pipeline fournissent les conditions et les seuils qui permettent de contrôler les nouvelles tentatives. Toutefois, un trop grand nombre de nouvelles tentatives peut retarder la détection d'une défaillance irrécupérable, car AWS Data Pipeline ne signale pas l'échec tant qu'il n'a pas épuisé le nombre total de nouvelles tentatives que vous avez spécifié. Les nouvelles tentatives supplémentaires peuvent occasionner des frais supplémentaires si elles sont exécutées sur les ressources AWS. En conséquence, déterminez soigneusement à quel moment il est approprié de dépasser les paramètres par défaut d'AWS Data Pipeline que vous utilisez pour contrôler les nouvelles tentatives et les paramètres associés.

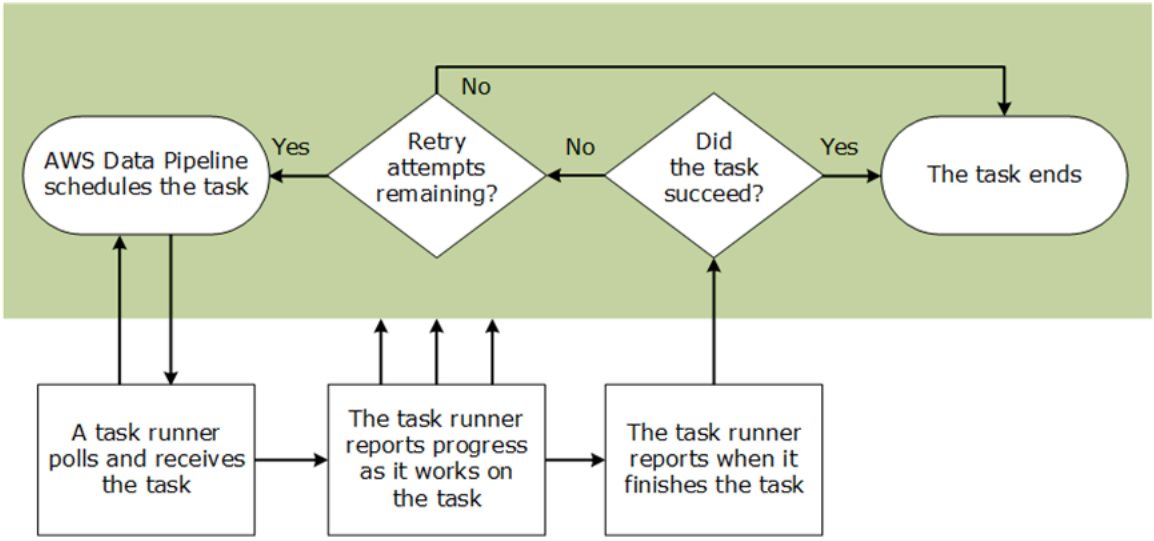


Exécuteurs de tâches

Un exécuteur de tâches est une application qui interroge AWS Data Pipeline pour rechercher les tâches, puis effectue ces tâches.

Task Runner est une implémentation par défaut d'un lanceur de tâches fourni parAWS Data Pipeline. Lorsque Task Runner est installé et configuré, il interroge AWS Data Pipeline les tâches associées aux pipelines que vous avez activés. Lorsqu'une tâche est attribuée à Task Runner, il l'exécute et indique son état àAWS Data Pipeline.

Le schéma suivant illustre la façon dont AWS Data Pipeline et un exécuteur de tâches interagissent pour traiter une tâche planifiée. Une tâche est une unité de travail discrète que le service AWS Data Pipeline partage avec un exécuteur de tâches. Elle diffère d'un pipeline, qui est une définition générale des activités et des ressources qui génère généralement plusieurs tâches.



Vous pouvez utiliser Task Runner pour traiter votre pipeline de deux manières :

- AWS Data Pipelineinstalle Task Runner pour vous sur les ressources lancées et gérées par le service AWS Data Pipeline Web.
- Vous installez Task Runner sur une ressource de calcul que vous gérez, telle qu'une instance EC2 de longue durée ou un serveur sur site.

Pour plus d'informations sur l'utilisation de Task Runner, consultez Utilisation de Task Runner.

Nœuds de données

Dans AWS Data Pipeline, un nœud de données définit l'emplacement et le type des données qu'une activité de pipeline utilise comme entrée ou sortie. AWS Data Pipeline prend en charge les types de nœuds de données suivants :

DynamoDB DataNode

Une table DynamoDB qui contient des données destinées HiveActivity ou EmrActivity à utiliser.

SqlDataNode

Requête de base de données et de table SQL qui représente les données qu'une activité de pipeline doit utiliser.

> **Note**
>
> Auparavant, MySqlDataNode était utilisé. Utilisez SqlDataNode à la place.

RedshiftDataNode

Un tableau Amazon Redshift contenant des données RedshiftCopyActivity à utiliser.

S3 DataNode

Un emplacement Amazon S3 qui contient un ou plusieurs fichiers destinés à être utilisés par une activité de pipeline.

Bases de données

AWS Data Pipeline prend en charge les types de bases de données suivants :

JdbcDatabase

Base de données JDBC.

RdsDatabase

Une base de données Amazon RDS.

RedshiftDatabase

Une base de données Amazon Redshift.

Activités

Dans AWS Data Pipeline, une activité est un composant de pipeline qui définit le travail à effectuer. AWS Data Pipeline fournit plusieurs activités pré-intégrées qui s'adaptent aux scénarios courants, tels que le transfert de données d'un emplacement vers un autre, l'exécution de requêtes Hive, etc. Les activités étant extensibles, vous pouvez exécuter vos propres scripts personnalisés pour prendre en charge une multitude de combinaisons.

AWS Data Pipeline prend en charge les types d'activités suivants :

CopyActivity

Copie les données d'un emplacement vers un autre.

EmrActivity

Exécute un cluster Amazon EMR.

HiveActivity

Exécute une requête Hive sur un cluster Amazon EMR.

HiveCopyActivity

Exécute une requête Hive sur un cluster Amazon EMR avec prise en charge du filtrage avancé des données et prise en charge de S3 DataNode et. DynamoDB DataNode

PigActivity

Exécute un script Pig sur un cluster Amazon EMR.

RedshiftCopyActivity

Copie les données depuis et vers les tables Amazon Redshift.

ShellCommandActivity

Exécute une commande shell UNIX/Linux personnalisée comme une activité.

SqlActivity

Exécute une requête SQL sur une base de données.

Certaines activités assurent la prise en charge spéciale des données et tables de base de données intermédiaires. Pour plus d'informations, veuillez consulter Copie intermédiaire des données et des tables avec les activités de pipeline.

Conditions préalables

Dans AWS Data Pipeline, une condition préalable est un composant de pipeline contenant des instructions conditionnelles qui doivent avoir la valeur true pour qu'une activité puisse être exécutée. Par exemple, une condition préalable peut vérifier si les données source sont présentes avant qu'une activité de pipeline ne tente de les copier. AWS Data Pipelinefournit plusieurs conditions préalables préconfigurées qui s'adaptent à des scénarios courants, tels que l'existence d'une table de base de données, la présence d'une clé Amazon S3, etc. Cependant, les conditions préalables sont

extensibles et vous permettent d'exécuter vos propres scripts personnalisés pour prendre en charge une multitude de combinaisons.

Il existe deux types de conditions préalables : les conditions préalables gérées par le système et les conditions préalables gérées par l'utilisateur. Les conditions préalables gérées par le système sont exécutées par le service web AWS Data Pipeline en votre nom et ne nécessitent aucune ressource de calcul. Les conditions préalables gérées par l'utilisateur s'exécutent uniquement sur la ressource de calcul que vous spécifiez à l'aide des champs `runsOn` ou `workerGroup`. La ressource `workerGroup` est dérivée de l'activité qui utilise la condition préalable.

Conditions préalables gérées par le système

DynamoDB DataExists

Vérifie si des données existent dans une table DynamoDB spécifique.

DynamoDB TableExists

Vérifie si une table DynamoDB existe.

S3 KeyExists

Vérifie si une clé Amazon S3 existe.

S3 PrefixNotEmpty

Vérifie si un préfixe Amazon S3 est vide.

Conditions préalables gérées par l'utilisateur

Existe

Vérifie si un nœud de données existe.

ShellCommandPrecondition

Exécute une commande shell Unix/Linux personnalisée en tant que condition préalable.

Ressources

Dans AWS Data Pipeline, une ressource est la ressource de calcul qui effectue le travail qu'une activité de pipeline spécifie. AWS Data Pipeline prend en charge les types de ressources suivants :

Ec2Resource

Instance EC2 qui exécute le travail défini par une activité de pipeline.

EmrCluster

Un cluster Amazon EMR qui exécute le travail défini par une activité de pipeline, tel queEmrActivity.

Les ressources peuvent s'exécuter dans la même région que leur ensemble de données de travail, même dans une région différente de celle d'AWS Data Pipeline. Pour plus d'informations, veuillez consulter Utilisation d'un pipeline avec des ressources dans plusieurs régions.

Limites des ressources

AWS Data Pipeline s'adapte pour prendre en charge un très grand nombre de tâches simultanées et vous pouvez le configurer de manière à créer automatiquement les ressources nécessaires pour gérer les charges de travail très importantes. Ces ressources créées automatiquement sont sous votre contrôle et prises en compte dans le calcul des limites des ressources de votre compte AWS. Par exemple, si vous configurez AWS Data Pipeline pour créer automatiquement un cluster Amazon EMR à 20 nœuds pour traiter les données et que la limite d'instances EC2 de votre compte AWS est fixée à 20, vous risquez d'épuiser par inadvertance vos ressources de remblayage disponibles. Par conséquent, tenez compte de ces restrictions de ressources dans votre conception ou augmentez les limites de votre compte en conséquence. Pour plus d'informations sur les limites du service, consultez Limites du service AWS dans le manuel AWS General Reference.

> **Note**
>
> La limite est de 1 instance par objet composant `Ec2Resource`.

Plateformes prises en charge

Les pipelines peuvent lancer vos ressources sur les plateformes suivantes :

EC2-Classic

Vos ressources s'exécutent sur un réseau plat unique que vous partagez avec d'autres clients.

EC2-VPC

Vos ressources s'exécutent dans un cloud privé virtuel (VPC) qui est logiquement isolé pour votre compte AWS.

Votre compte AWS peut lancer des ressources sur les deux plateformes, ou seulement sur EC2-VPC, région par région. Pour plus d'informations, consultez la section Plateformes prises en charge dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Si votre compte AWS prend uniquement en charge EC2-VPC, nous créons un VPC par défaut pour vous dans chaque région AWS. Par défaut, nous lançons vos ressources sur un sous-réseau par défaut de votre VPC par défaut. Sinon, vous pouvez créer un VPC personnalisé et spécifier l'un de ses sous-réseaux lorsque vous configurez vos ressources, et nous lancerons ensuite vos ressources sur le sous-réseau spécifié de ce VPC personnalisé.

Lorsque vous lancez une instance dans un VPC, vous devez spécifier un groupe de sécurité créé spécifiquement pour ce VPC. Vous ne pouvez pas spécifier un groupe de sécurité que vous avez créé pour EC2-Classic lorsque vous lancez une instance dans un VPC. En outre, vous devez utiliser l'ID de groupe de sécurité, et non le nom du groupe de sécurité, pour identifier un groupe de sécurité pour un VPC.

Instances Spot Amazon EC2 avec clusters Amazon EMR et AWS Data Pipeline

Les pipelines peuvent utiliser les instances Spot Amazon EC2 pour les nœuds de tâches de leurs ressources de cluster Amazon EMR. Par défaut, les pipelines utilisent des instances à la demande. Les instances Spot vous permettent d'utiliser des instances EC2 non utilisées et de les exécuter. Le modèle de tarification des instances Spot vient compléter les modèles de tarification des instances à la demande et réservées en permettant potentiellement d'offrir l'option la plus économique pour l'obtention d'une capacité de calcul, en fonction de votre application. Pour plus d'informations, consultez la page produit Instances Spot Amazon EC2.

Lorsque vous utilisez des instances Spot, AWS Data Pipeline soumet le prix maximum de votre instance Spot à Amazon EMR lorsque votre cluster est lancé. Celui-ci alloue automatiquement le travail du cluster au nombre de nœuds de tâches d'instance Spot que vous définissez à l'aide du champ `taskInstanceCount`. AWS Data Pipeline limite les instances Spot pour les nœuds de tâches afin de garantir que des nœuds principaux à la demande sont disponibles pour exécuter votre pipeline.

Vous pouvez modifier une instance de ressource de pipeline terminée ou ayant échoué pour ajouter des instances Spot. Lorsque le pipeline relance le cluster, il utilise les instances Spot pour les nœuds de tâches.

Considérations relatives aux instances Spot

Lorsque vous utilisez des instances Spot avec AWS Data Pipeline, les considérations suivantes s'appliquent :

- Vos instances Spot peuvent être résiliées lorsque le prix de l'instance Spot dépasse votre prix maximum pour l'instance ou pour des raisons de capacité Amazon EC2. Cependant, vous ne perdez pas vos données, car AWS Data Pipeline utilise des clusters avec des nœuds principaux qui sont toujours des instances à la demande et qui ne sont donc pas soumis à la résiliation.
- Les instances Spot peuvent prendre plus de temps à démarrer à mesure qu'elles atteignent leur capacité de manière asynchrone. Par conséquent, un pipeline d'instance Spot peut s'exécuter plus lentement qu'un pipeline d'instance à la demande équivalent.
- Votre cluster risque de ne pas s'exécuter si vous ne recevez pas vos instances Spot, par exemple, lorsque le prix de votre prix maximum est trop faible.

Actions

Les actions AWS Data Pipeline sont les étapes qu'un composant de pipeline réalise lorsque certains événements se produisent, tels que la réussite, l'échec ou le retard d'une activité. Le champ d'événement d'une activité fait référence à une action, par exemple une référence à `snsalarm` dans le champ `onLateAction` de l'activité `EmrActivity`.

AWS Data Pipelines'appuie sur les notifications Amazon SNS comme principal moyen d'indiquer l'état des pipelines et de leurs composants sans surveillance. Pour plus d'informations, consultez Amazon SNS. En plus des notifications SNS, vous pouvez utiliser la console AWS Data Pipeline et l'interface de ligne de commande pour obtenir des informations sur l'état du pipeline.

AWS Data Pipeline prend en charge les actions suivantes :

SnsAlarm

Action qui envoie une notification SNS à une rubrique en fonction d'événements `onSuccess`, `OnFail` et `onLateAction`.

Terminer

Action qui déclenche l'annulation d'une activité, d'une ressource ou d'un nœud de données inachevé ou en attente. Vous ne pouvez pas mettre fin à des actions qui comprennent `onSuccess`, `OnFail` ou `onLateAction`.

Surveillance proactive des pipelines

La meilleure façon de détecter les problèmes consiste à surveiller vos pipelines de manière proactive dès le début. Vous pouvez configurer les composants du pipeline pour vous informer de certaines situations ou de certains événements, par exemple lorsqu'un composant du pipeline tombe en panne ou ne démarre pas à l'heure de début prévue. AWS Data Pipelinefacilite la configuration des notifications en fournissant des champs d'événements sur les composants du pipeline que vous pouvez associer aux notifications Amazon SNS, tels que `onSuccessOnFail`, et`onLateAction`.

Configuration pour AWS Data Pipeline

Avant d'utiliser AWS Data Pipeline pour la première fois, exécutez les tâches suivantes :

Tâches

- Inscrivez-vous à AWS
- Création de rôles IAM pour les AWS Data Pipeline ressources et pipeline
- Autoriser les principaux IAM (utilisateurs et groupes) à effectuer les actions nécessaires
- Accorder un accès par programmation

Une fois ces tâches terminées, vous pouvez commencer à utiliser AWS Data Pipeline. Pour obtenir un didacticiel de base, consultez Démarrer avec AWS Data Pipeline.

Inscrivez-vous à AWS

Lorsque vous vous inscrivez à Amazon Web Services (AWS), votre compte AWS est automatiquement inscrit à tous les services d'AWS, y compris AWS Data Pipeline. Seuls les services que vous utilisez vous sont facturés. Pour plus d'informations sur les tarifs d'utilisation d'AWS Data Pipeline, consultez AWS Data Pipeline.

S’inscrire à un Compte AWS

Si vous n’avez pas de compte Compte AWS, procédez comme suit pour en créer un.

Pour s’inscrire à un Compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d’inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous souscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l’utilisateur root a accès à l’ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur administratif, et à uniquement utiliser l’utilisateur root pour effectuer tâches nécessitant un accès utilisateur root.

AWS vous envoie un e-mail de confirmation lorsque le processus d'inscription est terminé. Vous pouvez afficher l'activité en cours de votre compte et gérer votre compte à tout moment en accédant à https://aws.amazon.com/ et en cliquant sur Mon compte.

Création d'un utilisateur administratif

Après vous être inscrit à unCompte AWS, sécurisez Utilisateur racine d'un compte AWSAWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisation de votre Utilisateur racine d'un compte AWS

1. Connectez-vous à la AWS Management Console en tant que propriétaire du compte en sélectionnant Root user (Utilisateur racine) et en saisissant l'adresse e-mail de Compte AWS. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur root, consultez Connexion en tant qu'utilisateur root dans le Guide de l'utilisateur Connexion à AWS.
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur root.

   Pour obtenir des instructions, consultez Activation d'un dispositif MFA virtuel pour l'utilisateur root de votre Compte AWS (console) dans le Guide de l'utilisateur IAM.

Création d'un utilisateur administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez la section Activation AWS IAM Identity Center dans le guide de AWS IAM Identity Center l'utilisateur.
2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur administratif.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter à l'aide d'un utilisateur IAM Identity Center, consultez Connexion au portail d'accès AWS dans le Guide de l'utilisateur Connexion à AWS.

Création de rôles IAM pour les AWS Data Pipeline ressources et pipeline

AWS Data Pipelinenécessite des rôles IAM qui déterminent les autorisations pour effectuer des actions et accéder aux AWS ressources. Le rôle de pipeline détermine les autorisations dont il AWS Data Pipeline dispose, et un rôle de ressource détermine les autorisations dont disposent les applications exécutées sur des ressources de pipeline, telles que les instances EC2. Vous spécifiez ces rôles lorsque vous créez un pipeline. Même si vous ne spécifiez pas de rôle personnalisé et que vous utilisez les rôles `DataPipelineDefaultRole` par défaut`DataPipelineDefaultResourceRole`, vous devez d'abord créer les rôles et joindre des politiques d'autorisation. Pour plus d'informations, consultez Rôles IAM pour AWS Data Pipeline.

Autoriser les principaux IAM (utilisateurs et groupes) à effectuer les actions nécessaires

Pour utiliser un pipeline, un principal IAM (un utilisateur ou un groupe) de votre compte doit être autorisé à effectuer les AWS Data Pipelineactions requises et les actions pour les autres services tels que définis par votre pipeline.

Pour simplifier les autorisations, vous pouvez associer la politique AWSDataPipeline_FullAccessgérée aux principaux IAM. Cette politique gérée permet au principal d'effectuer toutes les actions requises par un utilisateur, ainsi que l'`iam:PassRole`action sur les rôles par défaut utilisés AWS Data Pipeline lorsqu'aucun rôle personnalisé n'est spécifié.

Nous vous recommandons vivement d'évaluer attentivement cette politique gérée et de limiter les autorisations uniquement à celles dont vos utilisateurs ont besoin. Si nécessaire, utilisez cette politique comme point de départ, puis supprimez les autorisations pour créer une politique d'autorisations intégrée plus restrictive que vous pouvez associer aux principaux IAM. Pour plus d'informations et des exemples de politiques d'autorisation, voir Exemples de stratégies pour AWS Data Pipeline

Une déclaration de politique similaire à l'exemple suivant doit être incluse dans une politique attachée à tout principal IAM qui utilise le pipeline. Cette instruction permet au principal IAM d'effectuer

l'`PassRole`action sur les rôles utilisés par un pipeline. Si vous n'utilisez pas de rôles par défaut, remplacez *`MyPipelineRole`* et *`MyResourceRole`* par les rôles personnalisés que vous créez.

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

La procédure suivante explique comment créer un groupe IAM, associer la politique AWSDataPipeline_FullAccessgérée au groupe, puis ajouter des utilisateurs au groupe. Vous pouvez utiliser cette procédure pour n'importe quelle politique intégrée

Pour créer un groupe d'utilisateurs **`DataPipelineDevelopers`** et associer la AWSDataPipeline_FullAccesspolitique

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, choisissez Groupes, puis Créer un groupe.
3. Entrez un nom de groupe, par exemple**`DataPipelineDevelopers`**, puis choisissez Next Step.
4. Entrez dans **`AWSDataPipeline_FullAccess`** le champ Filtre, puis sélectionnez-le dans la liste.
5. Cliquez sur Étape suivante, puis sur Créer un groupe.
6. Pour ajouter des utilisateurs au groupe :
    a. Sélectionnez le groupe que vous avez créé dans la liste des groupes.
    b. Choisissez Actions du groupe, Ajouter des utilisateurs au groupe.
    c. Sélectionnez les utilisateurs que vous souhaitez ajouter dans la liste, puis choisissez Ajouter des utilisateurs au groupe.

Accorder un accès par programmation

Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS en dehors de la AWS Management Console. La manière d'octroyer un accès par programmation dépend du type d'utilisateur qui accède à AWS.

Pour accorder aux utilisateurs un accès programmatique, choisissez l'une des options suivantes.

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Par
Identité de la main-d'œuvre (Utilisateurs gérés dans IAM Identity Center)	Utilisez des informations d'identification temporaires pour signer des demandes par programmation destinées à l'AWS CLI, aux kits SDK AWS ou aux API AWS.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour l'AWS CLI, consultez Configuration de l'AWS CLI pour l'utilisation d'AWS IAM Identity Center dans le Guide de l'utilisateur AWS Command Line Interface. • Pour les kits SDK et les outils AWS ainsi que les API AWS, veuillez consulter la rubrique Authentification IAM Identity Center dans le Guide de référence des kits SDK et des outils AWS.
IAM	Utilisez des informations d'identification temporaires pour signer des demandes par programmation destinées à l'AWS CLI, aux kits SDK AWS ou aux API AWS.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec des ressources AWS dans le Guide de l'utilisateur IAM.

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Par
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer des demandes par programmation destinées à l'AWS CLI, aux kits SDK AWS ou aux API AWS.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour l'AWS CLI, consultez Authentification à l'aide des informations d'identification d'utilisateur IAM dans le Guide de l'utilisateur AWS Command Line Interface. • Pour les kits SDK et les outils AWS, consultez Authentification à l'aide d'informations d'identification à long terme dans le Guide de référence des kits SDK et des outils AWS. • Pour les API AWS, consultez Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Démarrer avec AWS Data Pipeline

AWS Data Pipeline vous permet d'ordonner, de planifier, d'exécuter et de gérer les charges de travail récurrentes de traitement de données de manière fiable et rentable. Ce service vous permet de concevoir facilement des activités extract-transform-load (ETL) à l'aide de données structurées et non structurées, à la fois sur site et dans le cloud, en fonction de votre logique métier.

Pour utiliser AWS Data Pipeline, vous créez une définition de pipeline qui spécifie la logique métier du traitement de vos données. Une définition de pipeline classique comprend des activités qui définissent le travail à effectuer et des nœuds de données qui définissent l'emplacement et le type de données d'entrée et de sortie.

Dans ce tutoriel, vous exécutez un script de commande shell qui compte le nombre de demandes GET dans les journaux du serveur web Apache. Ce pipeline s'exécute toutes les 15 minutes pendant une heure et écrit la sortie sur Amazon S3 à chaque itération.

Prérequis

Avant de commencer, complétez les tâches détaillées dans Configuration pour AWS Data Pipeline.

Objets de pipeline

Le pipeline utilise les objets suivants :

ShellCommandActivity

Lit le fichier journal en entrée et compte le nombre d'erreurs.

S3 DataNode (input)

Compartiment S3 qui contient le fichier journal en entrée.

S3 DataNode (sortie)

Compartiment S3 de la sortie.

Ec2Resource

Ressource de calcul qu'AWS Data Pipeline utilise pour exécuter l'activité.

Notez que si vous avez une grande quantité de données de fichier journal, vous pouvez configurer votre pipeline afin d'utiliser un cluster EMR à la place d'une instance EC2 pour traiter les fichiers.

Planificateur

Définit que l'activité est exécutée toutes les 15 minutes pendant une heure.

Tâches

- Création du pipeline
- Surveillance de l'exécution du pipeline
- Affichage de la sortie
- Suppression du pipeline

Création du pipeline

Le moyen le plus rapide pour faire vos premiers pas avec AWS Data Pipeline consiste à utiliser une définition de pipeline appelée modèle.

Pour créer le pipeline

1. Ouvrez la AWS Data Pipeline console à l'adresse https://console.aws.amazon.com/datapipeline/.
2. Dans la barre de navigation, sélectionnez une région. Vous pouvez sélectionner n'importe quelle région disponible, quel que soit votre emplacement. De nombreuses ressources AWS sont spécifiques à une région, mais AWS Data Pipeline vous permet d'utiliser les ressources d'une autre région que celle du pipeline.
3. Le premier écran qui s'affiche varie selon que vous avez créé ou non un pipeline dans la région actuelle.

   a. Si vous n'avez pas créé de pipeline dans cette région, la console affiche un écran d'introduction. Sélectionnez Pour commencer.

   b. Si vous avez déjà créé un pipeline dans cette région, la console affiche une page qui répertorie vos pipelines pour la région. Choisissez Create new pipeline.
4. Dans Nom, entrez le nom de votre pipeline.
5. (Facultatif) Dans Description, entrez une description pour votre pipeline.
6. Pour Source, sélectionnez Créer à l'aide d'un modèle, puis sélectionnez le modèle suivant : Commencer à utiliser ShellCommandActivity.
7. Dans la section Parameters, qui s'est ouverte quand vous avez sélectionné le modèle, conservez les valeurs par défaut de S3 input folder et de Shell command to run. Cliquez sur l'icône de

dossier en regard de S3 output folder, sélectionnez l'un de vos compartiments ou dossiers, puis cliquez sur Select.

8. Sous Schedule, conservez les valeurs par défaut. Lorsque vous activez le pipeline, le pipeline exécute le démarrage, puis poursuit toutes les 15 minutes pendant une heure.

   Si vous préférez, vous pouvez sélectionner Run once on pipeline activation.

9. Sous Configuration du pipeline, laissez la journalisation activée. Choisissez l'icône du dossier sous l'emplacement S3 pour les journaux, sélectionnez l'un de vos compartiments ou dossiers, puis choisissez Sélectionner.

   Si vous préférez, vous pouvez désactiver la journalisation à la place.

10. Sous Sécurité/Accès, laissez les rôles IAM définis sur Par défaut.
11. Cliquez sur Activate.

    Si vous préférez, vous pouvez choisir Modifier dans Architect pour modifier ce pipeline. Par exemple, vous pouvez ajouter des conditions préalables.

Surveillance de l'exécution du pipeline

Une fois que vous avez activé votre pipeline, vous êtes redirigé vers la page Execution details où vous pouvez surveiller la progression de votre pipeline.

Pour surveiller la progression de votre pipeline

1. Cliquez sur Update ou appuyez sur F5 pour mettre à jour le statut affiché.

   > **Tip**
   >
   > Si aucune exécution n'est affichée, assurez-vous que les valeurs Start (in UTC) et End (in UTC) couvrent les début et fin planifiés de votre pipeline, puis cliquez sur Update.

2. Lorsque le statut de tous les objets de votre pipeline est `FINISHED`, votre pipeline a terminé avec succès l'exécution de tâches planifiées.
3. Si votre pipeline ne s'est pas terminé avec succès, vérifiez les paramètres de votre pipeline à la recherche d'éventuels problèmes. Pour plus d'informations sur le dépannage des exécutions d'instance en échec ou incomplètes de votre pipeline, consultez Résolution des problèmes courants.

Affichage de la sortie

Ouvrez la console Amazon S3 et accédez à votre compartiment. Si vous avez exécuté votre pipeline toutes les 15 minutes pendant une heure, quatre sous-dossiers horodatés s'affichent. Chaque sous-dossier contient la sortie dans un fichier nommé `output.txt`. Dans la mesure où nous avons exécuté le script sur le même fichier d'entrée à chaque fois, les fichiers de sortie sont identiques.

Suppression du pipeline

Pour ne plus encourir de frais, supprimez votre pipeline. La suppression de votre pipeline entraîne la suppression de la définition du pipeline et de tous les objets associés.

Pour supprimer votre pipeline

1. Sur la page Liste des pipelines, sélectionnez votre pipeline.
2. Cliquez sur Actions, puis choisissez Supprimer.
3. Lorsque vous êtes invité à confirmer l'opération, choisissez Supprimer.

Si vous avez terminé le résultat de ce didacticiel, supprimez les dossiers de sortie de votre compartiment Amazon S3.

Utilisation des pipelines

Vous pouvez administrer, créer et modifier des pipelines à l'aide de l'interface de ligne de commande (CLI) ou du AWS SDK. Les sections suivantes présentent les concepts AWS Data Pipeline fondamentaux et vous expliquent comment utiliser les pipelines.

> **Important**
>
> Avant de commencer, consultez Configuration pour AWS Data Pipeline.

Table des matières

Création d'un pipeline

AWS Data Pipeline fournit plusieurs façons de créer des pipelines :

- Utilisez le AWS Command Line Interface (CLI) avec un modèle fourni pour vous faciliter la tâche. Pour plus d'informations, veuillez consulter Création d'un pipeline à partir de modèles de pipeline de données à l'aide de l'interface de ligne de commande.
- Utilisez l'AWS Command Line Interface avec un fichier de définition de pipeline au format JSON.

- Utilisez un kit SDK AWS avec une API spécifique au langage. Pour plus d'informations, veuillez consulter Utilisation de l'API.

Création d'un pipeline à partir de modèles de pipeline de données à l'aide de l'interface de ligne de commande

Data Pipeline fournit plusieurs définitions de pipeline préconfigurées, appelées modèles. Vous pouvez utiliser ces modèles pour commencer à utiliser rapidement AWS Data Pipeline. Ces modèles sont disponibles dans un compartiment public sur l'emplacement Amazon S3 :`s3://datapipeline-us-east-1/templates/`. Ces modèles prédéfinis sont créés pour répondre à des cas d'utilisation spécifiques et peuvent être utilisés pour créer des pipelines. Vous pouvez l'utiliser `aws s3 ls --recursive "s3://datapipeline-us-east-1/templates/"` pour répertorier tous les modèles disponibles.

Création d'un pipeline à partir d'un modèle à l'aide de l'interface de ligne de commande

Supposons que vous souhaitiez créer un pipeline qui exporte une table DynamoDB vers Amazon S3. Le modèle à utiliser dans ce cas se trouve à l'adresse :`s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json`.

Pour télécharger le modèle JSON et créer un pipeline à l'aide de l'interface de ligne de commande

1. Téléchargez le modèle à l'aide de l'`aws s3 cp`interface de ligne de commande ou de curl. Par exemple :

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

2. Apportez les modifications nécessaires au modèle téléchargé. Par exemple, pour utiliser la dernière version d'EMR, modifiez le `releaseLabel` champ dans `EmrClusterForBackup` l'objet, modifiez les types d'instance principale et principale et modifiez les valeurs par défaut des paramètres du modèle.
3. Créez un pipeline à l'aide de l'`create-pipeline`interface de ligne de commande. Par exemple :

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

4. Notez l'ID du pipeline créé.
5. `put-pipeline-definition`À utiliser pour télécharger la définition. Fournissez les valeurs des paramètres dont vous souhaitez remplacer les valeurs par défaut à l'aide de `--parameter-values` cette option.

Pour plus d'informations sur les modèles, consultez Choisir un modèle.

Choisir un modèle

Les modèles suivants sont disponibles au téléchargement depuis le compartiment Amazon S3 : `s3://datapipeline-us-east-1/templates/`

Modèles

- Mise en route avec ShellCommandActivity
- Exécuter la AWS commande CLI
- Exporter la table DynamoDB vers S3
- Importer des données de sauvegarde DynamoDB depuis S3
- Exécuter la tâche sur un cluster Amazon EMR
- Copie complète de la table MySQL Amazon RDS pour Amazon S3
- Copie incrémentielle de la table MySQL Amazon RDS vers Amazon S3
- Charger les données S3 dans la table MySQL Amazon RDS
- Copie complète de la table MySQL Amazon RDS vers Amazon Redshift
- Copie incrémentielle d'une table MySQL Amazon RDS vers Amazon Redshift
- Charger des données depuis Amazon S3 dans Amazon Redshift

Mise en route avec ShellCommandActivity

Le ShellCommandActivity modèle Getting Started using exécute un script de commande shell pour compter le nombre de requêtes GET dans un fichier journal. La sortie est écrite dans un emplacement Amazon S3 horodaté à chaque exécution planifiée du pipeline.

Le modèle utilise les objets de pipeline suivants :

- ShellCommandActivity
- S3 InputNode

- S3 OutputNode
- Ec2Resource

Exécuter la AWS commande CLI

Ce modèle exécute une commande d'AWS CLI spécifiée par l'utilisateur à intervalles planifiés.

Exporter la table DynamoDB vers S3

Le modèle Exporter la table DynamoDB vers S3 planifie un cluster Amazon EMR pour exporter les données d'une table DynamoDB vers un compartiment Amazon S3. Ce modèle utilise un cluster Amazon EMR, dont la taille est proportionnelle à la valeur du débit disponible pour la table DynamoDB. Même si vous pouvez augmenter les IOPS sur une table, cette opération peut entraîner des coûts supplémentaires lors de l'importation et l'exportation. Auparavant, l'export utilisait un format natif HiveActivity mais utilise désormais un format natifMapReduce.

Le modèle utilise les objets de pipeline suivants :

- EmrActivity
- EmrCluster
- DynamoDB DataNode
- S3 DataNode

Importer des données de sauvegarde DynamoDB depuis S3

Le modèle Importer des données de sauvegarde DynamoDB depuis S3 planifie un cluster Amazon EMR pour charger une sauvegarde DynamoDB créée précédemment dans Amazon S3 dans une table DynamoDB. Les éléments existants de la table DynamoDB sont mis à jour avec ceux issus des données de sauvegarde et de nouveaux éléments sont ajoutés au tableau. Ce modèle utilise un cluster Amazon EMR, dont la taille est proportionnelle à la valeur du débit disponible pour la table DynamoDB. Même si vous pouvez augmenter les IOPS sur une table, cette opération peut entraîner des coûts supplémentaires lors de l'importation et l'exportation. Auparavant, l'import utilisait un HiveActivity mais utilise désormais le natifMapReduce.

Le modèle utilise les objets de pipeline suivants :

- EmrActivity

- EmrCluster
- DynamoDB DataNode
- S3 DataNode
- S3 PrefixNotEmpty

Exécuter la tâche sur un cluster Amazon EMR

Le modèle Run Job on an Elastic MapReduce Cluster lance un cluster Amazon EMR en fonction des paramètres fournis et commence à exécuter des étapes en fonction du calendrier spécifié. Une fois la tâche terminée, le cluster EMR est arrêté. Des actions d'amorçage facultatives peuvent être spécifiées pour installer un logiciel supplémentaire ou modifier la configuration de l'application sur le cluster.

Le modèle utilise les objets de pipeline suivants :

- EmrActivity
- EmrCluster

Copie complète de la table MySQL Amazon RDS pour Amazon S3

Le modèle Copie complète de la table RDS MySQL vers S3 copie l'intégralité d'une table MySQL Amazon RDS et stocke la sortie dans un emplacement Amazon S3. La sortie est stockée sous forme de fichier CSV dans un sous-dossier horodaté à l'emplacement Amazon S3 spécifié.

Le modèle utilise les objets de pipeline suivants :

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Copie incrémentielle de la table MySQL Amazon RDS vers Amazon S3

Le modèle de copie incrémentielle de la table MySQL RDS vers S3 effectue une copie incrémentielle des données d'une table MySQL Amazon RDS et stocke la sortie dans un emplacement Amazon S3. La table MySQL Amazon RDS doit comporter une colonne Dernière modification.

Ce modèle copie les modifications qui ont été apportées à la table entre les intervalles planifiés à partir de l'heure de début planifiée. Le type de calendrier est une série chronologique. Ainsi, si une copie était planifiée pour une heure donnée, AWS Data Pipeline copie les lignes du tableau dont l'horodatage de la dernière modification se situe dans l'heure. Les suppressions physiques de la table ne sont pas copiées. La sortie est écrite dans un sous-dossier horodaté situé sous l'emplacement Amazon S3 à chaque exécution planifiée.

Le modèle utilise les objets de pipeline suivants :

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Charger les données S3 dans la table MySQL Amazon RDS

Le modèle Charger les données S3 dans la table RDS MySQL planifie une instance Amazon EC2 pour copier le fichier CSV depuis le chemin de fichier Amazon S3 spécifié ci-dessous vers une table MySQL Amazon RDS. Le fichier CSV ne doit pas contenir de ligne d'en-tête. Le modèle met à jour les entrées existantes de la table MySQL Amazon RDS avec celles des données Amazon S3 et ajoute de nouvelles entrées issues des données Amazon S3 à la table Amazon RDS MySQL. Vous pouvez charger les données dans une table existante ou fournir une requête SQL pour créer une table.

Le modèle utilise les objets de pipeline suivants :

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

Modèles Amazon RDS vers Amazon Redshift

Les deux modèles suivants copient des tables d'Amazon RDS MySQL vers Amazon Redshift à l'aide d'un script de traduction, qui crée une table Amazon Redshift à l'aide du schéma de la table source avec les mises en garde suivantes :

- Si aucune clé de distribution n'est spécifiée, la première clé primaire de la table Amazon RDS est définie comme clé de distribution.
- Vous ne pouvez pas ignorer une colonne présente dans une table MySQL Amazon RDS lorsque vous effectuez une copie vers Amazon Redshift.
- (Facultatif) Vous pouvez fournir un mappage des types de données de colonne Amazon RDS MySQL vers Amazon Redshift comme l'un des paramètres du modèle. Si cela est spécifié, le script l'utilise pour créer la table Amazon Redshift.

Si le mode d'insertion d'`Overwrite_Existing`Amazon Redshift est utilisé :

- Si aucune clé de distribution n'est fournie, une clé primaire de la table MySQL Amazon RDS est utilisée.
- S'il existe des clés primaires composites sur la table, la première est utilisée comme la clé de distribution si la clé de distribution n'est pas fournie. Seule la première clé composée est définie comme clé primaire dans le tableau Amazon Redshift.
- Si aucune clé de distribution n'est fournie et qu'il n'y a pas de clé primaire sur la table MySQL Amazon RDS, l'opération de copie échoue.

Pour plus d'informations sur Amazon Redshift, consultez les rubriques suivantes :

- Cluster Amazon Redshift
- COPIE D'Amazon Redshift
- Styles de distribution et DISTKEY (exemples)
- Clés de tri

La table suivante décrit la manière dont le script convertit les types de données :

Traductions de types de données entre MySQL et Amazon Redshift

Type de données MySQL	Type de données Amazon Redshift	Remarques
TINYINT, TINYINT (taille)	SMALLINT	MySQL : -128 à 127. Le nombre maximal de chiffres peut être spécifié entre parenthèses.

Type de données MySQL	Type de données Amazon Redshift	Remarques
		Amazon Redshift : INT2. Entier signé sur deux octets
TINYINT UNSIGNED, TINYINT (taille) UNSIGNED	SMALLINT	MySQL : 0 à 255 UNSIGNED. Le nombre maximal de chiffres peut être spécifié entre parenthèses. Amazon Redshift : INT2. Entier signé sur deux octets
SMALLINT, SMALLINT(taille)	SMALLINT	MySQL : -32768 à 32767 normal. Le nombre maximal de chiffres peut être spécifié entre parenthèses. Amazon Redshift : INT2. Entier signé sur deux octets
SMALLINT UNSIGNED, SMALLINT(taille) UNSIGNED,	INTEGER	MySQL : 0 à 65535 UNSIGNED*. Le nombre maximal de chiffres peut être spécifié entre parenthèses Amazon Redshift : INT4. Entier signé sur quatre octets
MEDIUMINT, MEDIUMINT(taille)	INTEGER	MySQL : 388608 à 8388607. Le nombre maximal de chiffres peut être spécifié entre parenthèses Amazon Redshift : INT4. Entier signé sur quatre octets

Type de données MySQL	Type de données Amazon Redshift	Remarques
MEDIUMINT UNSIGNED, MEDIUMINT(taille) UNSIGNED	INTEGER	MySQL : 0 à 16777215. Le nombre maximal de chiffres peut être spécifié entre parenthèses Amazon Redshift : INT4. Entier signé sur quatre octets
INT, INT(taille)	INTEGER	MySQL : 147483648 à 2147483647 Amazon Redshift : INT4. Entier signé sur quatre octets
INT UNSIGNED, INT(taille) UNSIGNED	BIGINT	MySQL : 0 à 4294967295 Amazon Redshift : INT8. Entier signé sur huit octets
BIGINT BIGINT(taille)	BIGINT	Amazon Redshift : INT8. Entier signé sur huit octets
BIGINT UNSIGNED BIGINT(taille) UNSIGNED	VARCHAR(20*4)	MySQL : 0 à 18446744073709551615 Amazon Redshift : pas d'équivalent natif, donc utilisez un tableau de caractères.

Type de données MySQL	Type de données Amazon Redshift	Remarques
FLOAT FLOAT(taille,d) FLOAT(taille,d) UNSIGNED	REAL	Le nombre maximal de chiffres peut être spécifié dans le paramètre de la taille. Le nombre maximal de chiffres après la virgule est spécifié dans le paramètre d. Amazon Redshift : FLOAT4
DOUBLE(taille,d)	DOUBLE PRECISION	Le nombre maximal de chiffres peut être spécifié dans le paramètre de la taille. Le nombre maximal de chiffres après la virgule est spécifié dans le paramètre d. Amazon Redshift : FLOAT8
DECIMAL(taille,d)	DECIMAL(taille,d)	Un DOUBLE stocké sous la forme de chaîne, permettant ainsi une décimale fixe. Le nombre maximal de chiffres peut être spécifié dans le paramètre de la taille. Le nombre maximal de chiffres après la virgule est spécifié dans le paramètre d. Amazon Redshift : aucun équivalent natif.

Type de données MySQL	Type de données Amazon Redshift	Remarques
CHAR(taille)	VARCHAR(taille*4)	Contient une chaîne de longueur fixe, qui peut être composée de lettres, de chiffres et de caractère s spéciaux. La taille fixe est spécifiée en tant que paramètre entre parenthès es. Peut stocker jusqu'à 255 caractères. Complété à droite par des espaces. Amazon Redshift : le type de données CHAR ne prend pas en charge les caractère s multioctets, c'est pourquoi VARCHAR est utilisé. Le nombre maximal d'octets par caractère est 4 selon la norme RFC3629, qui limite la table de caractères à U +10FFFF.

Type de données MySQL	Type de données Amazon Redshift	Remarques
VARCHAR(taille)	VARCHAR(taille*4)	Peut stocker jusqu'à 255 caractères. VARCHAR ne prend pas en charge les points de code UTF-8 non valides suivants : 0xD800 - 0xDFFF, (Séquences d'octets : ED A0 80 - ED BF BF), 0xFDD0 - 0xFDEF, 0xFFFE et 0xFFFF, (Séquences d'octets : EF B7 90 - EF B7 AF, EF BF BE et EF BF BF)
TINYTEXT	VARCHAR(255*4)	Contient une chaîne d'une longueur maximale de 255 caractères
TEXT	VARCHAR(max)	Contient une chaîne d'une longueur maximale de 65 535 caractères.
MEDIUMTEXT	VARCHAR(max)	0 à 16 777 215 caractères
LONGTEXT	VARCHAR(max)	0 à 4 294 967 295 caractères
BOOLEAN BOOL TINYINT(1)	BOOLEAN	MySQL : ces types sont des synonymes de TINYINT(1). Une valeur de zéro est considérée comme false. Les valeurs autres que zéro sont considérées comme true.
BINARY[(M)]	varchar(255)	M est compris entre 0 et 255 octets, FIXED

Type de données MySQL	Type de données Amazon Redshift	Remarques
VARBINARY(M)	VARCHAR(max)	0 à 65 535 octets
TINYBLOB	VARCHAR(255)	0 à 255 octets
BLOB	VARCHAR(max)	0 à 65 535 octets
MEDIUMBLOB	VARCHAR(max)	0 à 16 777 215 octets
LONGBLOB	VARCHAR(max)	0 à 4 294 967 295 octets
ENUM	VARCHAR(255*2)	La limite n'est pas sur la longueur de la chaîne enum littérale, mais plutôt sur la définition de la table en fonction du nombre de valeurs enum.
SET	VARCHAR(255*2)	Comme enum.
DATE	DATE	(AAAA-MM-JJ) « 1000-01-01 » à « 9999-12-31 »
TIME	VARCHAR(10*4)	(hh:mm:ss) « -838:59:59 » à « 838:59:59 »
DATETIME	TIMESTAMP	(AAAA-MM-JJ hh:mm:ss) 1000-01-01 00:00:00 » à « 9999-12-31 23:59:59 »
TIMESTAMP	TIMESTAMP	(AAAAMMJJhhmmss) 19700101000000 à 2037+

Type de données MySQL	Type de données Amazon Redshift	Remarques
YEAR	VARCHAR(4*4)	(YYYY) 1900 à 2155
colonne SERIAL	Génération d'ID / cet attribut n'est pas nécessaire pour un entrepôt de données OLAP depuis que cette colonne est copiée. Le mot-clé SERIAL n'est pas ajouté lors de la conversion.	SERIAL est une entité nommée SEQUENCE. Il existe de manière indépenda nte sur le reste de votre table. colonne GENERATED BY DEFAULT équivalente à : CREATE SEQUENCE nom ; CREATE TABLE table (colonne INTEGER NOT NULL DEFAULT nextval(n om)) ;
colonne BIGINT UNSIGNED NOT NULL AUTO_INCR EMENT UNIQUE	Génération d'ID / cet attribut n'est pas nécessaire pour l'entrepôt de données OLAP depuis que cette colonne est copiée. Le mot-clé SERIAL n'est donc pas ajouté lors de la conversio n.	SERIAL est une entité nommée SEQUENCE. Il existe de manière indépenda nte sur le reste de votre table. colonne GENERATED BY DEFAULT équivalente à : CREATE SEQUENCE nom ; CREATE TABLE table (colonne INTEGER NOT NULL DEFAULT nextval(n om)) ;

Type de données MySQL	Type de données Amazon Redshift	Remarques
ZEROFILL	Le mot-clé ZEROFILL n'est pas ajouté lors de la conversion.	INT UNSIGNED ZEROFILL NOT NULL ZEROFILL rembourre la valeur du champ affiché avec des zéros jusqu'à la largeur d'affichage spécifiée dans la définition de la colonne. Les valeurs plus longues que la largeur d'affichage ne sont pas tronquées. Notez que l'utilisation de ZEROFILL implique également UNSIGNED.

Copie complète de la table MySQL Amazon RDS vers Amazon Redshift

La copie complète de la table Amazon RDS MySQL vers Amazon Redshift copie l'intégralité de la table Amazon RDS MySQL vers une table Amazon Redshift en stockant les données dans un dossier Amazon S3. Le dossier intermédiaire Amazon S3 doit se trouver dans la même région que le cluster Amazon Redshift. Une table Amazon Redshift est créée avec le même schéma que la table MySQL Amazon RDS source si elle n'existe pas déjà. Veuillez indiquer les remplacements de type de données de colonne Amazon RDS MySQL vers Amazon Redshift que vous souhaitez appliquer lors de la création de la table Amazon Redshift.

Le modèle utilise les objets de pipeline suivants :

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

Copie incrémentielle d'une table MySQL Amazon RDS vers Amazon Redshift

La copie incrémentielle de la table Amazon RDS MySQL vers le modèle Amazon Redshift copie les données d'une table Amazon RDS MySQL vers une table Amazon Redshift en stockant les données dans un dossier Amazon S3.

Le dossier intermédiaire Amazon S3 doit se trouver dans la même région que le cluster Amazon Redshift.

AWS Data Pipelineutilise un script de traduction pour créer une table Amazon Redshift avec le même schéma que la table MySQL Amazon RDS source si elle n'existe pas déjà. Vous devez fournir tous les remplacements de type de données de colonne Amazon RDS MySQL vers Amazon Redshift que vous souhaitez appliquer lors de la création d'une table Amazon Redshift.

Ce modèle copie les modifications apportées à la table MySQL Amazon RDS entre des intervalles planifiés, à partir de l'heure de début planifiée. Les suppressions physiques de la table MySQL Amazon RDS ne sont pas copiées. Vous devez fournir le nom de la colonne qui stocke la valeur de l'heure de la dernière modification.

Lorsque vous utilisez le modèle par défaut pour créer des pipelines pour une copie incrémentielle d'Amazon RDS, une activité portant le nom par défaut `RDSToS3CopyActivity` est créée. Vous pouvez la renommer.

Le modèle utilise les objets de pipeline suivants :

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

Charger des données depuis Amazon S3 dans Amazon Redshift

Le modèle Load data from S3 into Redshift copie les données d'un dossier Amazon S3 dans une table Amazon Redshift. Vous pouvez charger les données dans une table existante ou fournir une requête SQL pour créer la table.

Les données sont copiées en fonction des COPY options Amazon Redshift. La table Amazon Redshift doit avoir le même schéma que les données d'Amazon S3. Pour connaître COPY les options, consultez la section COPY du manuel Amazon Redshift Database Developer Guide.

Le modèle utilise les objets de pipeline suivants :

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

Création d'un pipeline à l'aide de modèles paramétrés

Vous pouvez utiliser un modèle paramétré pour personnaliser une définition de pipeline. Vous pouvez ainsi créer une définition de pipeline courante, mais fournir des paramètres différents lorsque vous ajoutez la définition de pipeline à un nouveau pipeline.

Table des matières

Ajouter MyVariables à la définition du pipeline

Lorsque vous créez le fichier de définition de pipeline, spécifiez les variables à l'aide de la syntaxe suivante : #{my*Variable*}. Il est nécessaire que le préfixe de la variable soit my. Par exemple, le fichier de définition de pipeline suivant inclut les variables suivantes : *myShellCmdMyS3 InputLoc* et *OutputLocMyS3*. pipeline-definition.json

> **Note**
>
> Une définition de pipeline est limitée à 50 paramètres au maximum.

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

Définir des objets de paramètres

Vous pouvez créer un fichier séparé avec les objets de paramètre qui définit les variables dans la définition votre pipeline. Par exemple, le fichier JSON suivant contient des objets de paramètres pour les *myShellCmd*OutputLocvariables *MyS3 et MyS3 InputLoc* issus de l'exemple de définition de pipeline ci-dessus. `parameters.json`

```
{
  "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**
>
> Vous pouvez ajouter ces objets directement au fichier de définition du pipeline au lieu d'utiliser un fichier séparé.

Le tableau suivant décrit les attributs des objets de paramètre.

Attributs des paramètres

Attribut	Type	Description
`id`	Chaîne	L'identifiant unique du paramètre. Pour masquer la valeur lorsqu'elle est saisie ou affichée, ajoutez un astérisque (« * ») comme préfixe. Par exemple, `*myVariable` —. Notez que ceci chiffre également la valeur avant qu'elle soit stockée par AWS Data Pipeline.
description	Chaîne	Description du paramètre.
type	Chaîne, entier, double ou AWS::S3::ObjectKey	Le type de paramètre qui définit la plage autorisée des valeurs d'entrée et des règles de validation. La valeur par défaut est une chaîne.
facultatif	Booléen	Indique si le paramètre est facultatif ou obligatoire. La valeur par défaut est `false`.

Attribut	Type	Description
allowedValues	Liste de chaînes	Enumère toutes les valeurs autorisées du paramètre.
default	Chaîne	La valeur par défaut du paramètre. Si vous spécifiez une valeur de ce paramètre à l'aide de valeurs de paramètre, cette valeur remplace la valeur par défaut.
isArray	Booléen	Indique si le paramètre est un tableau.

Définition des valeurs de paramètre

Vous pouvez créer un fichier séparé pour définir vos variables à l'aide de valeurs de paramètre. Par exemple, le fichier JSON suivant contient la valeur de la OutputLoc variable *MyS3* issue de l'exemple de définition de pipeline ci-dessus. `file://values.json`

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

Soumission de la définition du pipeline

Lorsque vous soumettez votre définition de pipeline, vous pouvez spécifier les paramètres, les objets de paramètre et les valeurs de paramètre. Par exemple, vous pouvez utiliser la [put-pipeline-definition](#)AWS CLIcommande comme suit :

```
$ 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**
>
> Une définition de pipeline est limitée à 50 paramètres au maximum. La taille du fichier pour `parameter-values-uri` est limitée à 15 Ko au maximum.

Affichage de vos pipelines

Vous pouvez consulter vos pipelines à l'aide de l'interface de ligne de commande (CLI).

Pour afficher vos pipelines à l'aide de l'AWS CLI

- Utilisez la commande list-pipelines pour répertorier vos pipelines :

```
aws datapipeline list-pipelines
```

Interprétation des codes d'état de pipeline

Les niveaux d'état affichés dans la console AWS Data Pipeline et l'interface de ligne de commande indiquent la condition d'un pipeline et de ses composants. L'état du pipeline n'est qu'une présentation d'un pipeline ; pour afficher plus d'informations, consultez l'état de chaque composant du pipeline.

L'état d'un pipeline est `SCHEDULED` s'il est prêt (la définition de pipeline a été validée), exécute actuellement ses tâches ou a terminé ses tâches. L'état d'un pipeline est `PENDING` s'il n'est pas activé ou n'est pas en mesure d'exécuter ses tâches (par exemple, la définition de pipeline n'a pas été validée).

Un pipeline est considéré comme inactif si son état est `PENDING`, `INACTIVE` ou `FINISHED`. Les pipelines inactifs sont facturés (pour plus d'informations, consultez la Tarification).

Codes de statut

`ACTIVATING`

Le composant ou la ressource est en cours de démarrage, par exemple une instance EC2.

`CANCELED`

Le composant a été annulé par un utilisateur ou AWS Data Pipeline avant son exécution. Cela peut se produire automatiquement lorsqu'une défaillance survient dans un composant ou une ressource différente dont dépend ce composant.

`CASCADE_FAILED`

Le composant ou la ressource a été annulé à la suite d'une défaillance en cascade liée à l'une de ses dépendances, mais le composant n'était probablement pas à l'origine de la panne.

`DEACTIVATING`

Le pipeline est en cours de désactivation.

`FAILED`

Le composant ou la ressource a rencontré une erreur et a cessé de fonctionner. Lorsqu'un composant ou une ressource tombe en panne, cela peut entraîner des annulations et des défaillances se répercuter sur d'autres composants qui en dépendent.

`FINISHED`

La composante a terminé le travail qui lui avait été assigné.

`INACTIVE`

Le pipeline a été désactivé.

`PAUSED`

Le composant a été suspendu et ne fonctionne pas actuellement.

`PENDING`

Le pipeline est prêt à être activé pour la première fois.

`RUNNING`

La ressource est en cours d'exécution et prête à recevoir du travail.

`SCHEDULED`

L'exécution de la ressource est planifiée.

`SHUTTING_DOWN`

La ressource s'arrête après avoir terminé son travail avec succès.

`SKIPPED`

Le composant a ignoré des intervalles d'exécution après l'activation du pipeline en utilisant un horodatage postérieur au calendrier actuel.

`TIMEDOUT`

La ressource a dépassé le `terminateAfter` seuil et a été bloquéeAWS Data Pipeline. Une fois que la ressource a atteint ce statut, AWS Data Pipeline ignore les `retryTimeout` valeurs `actionOnResourceFailureretryDelay`, et de cette ressource. Ce statut s'applique uniquement aux ressources.

`VALIDATING`

La définition du pipeline est en cours de validation parAWS Data Pipeline.

`WAITING_FOR_RUNNER`

Le composant attend que son client de travail récupère un élément de travail. La relation entre le composant et le travailleur-client est contrôlée par les `workerGroup` champs `runsOn` ou définis par ce composant.

`WAITING_ON_DEPENDENCIES`

Le composant vérifie que ses conditions préalables par défaut et configurées par l'utilisateur sont remplies avant d'effectuer son travail.

Interprétation de l'état de santé des pipelines et des composants

Chaque pipeline et composant au sein de ce pipeline renvoie un état de santé `HEALTHY`, `ERROR`, `"-"`, `No Completed Executions` ou `No Health Information Available`. Un pipeline affiche un état de santé uniquement après la première exécution d'un composant du pipeline ou si les conditions préalables du composant ont échoué. L'état de santé des composants s'intègre dans l'état de santé du pipeline dans ces états d'erreur qui sont visibles la première fois que vous consultez les détails d'exécution de votre pipeline.

États de santé d'un pipeline

`HEALTHY`

L'état de santé agrégé de tous les composants est `HEALTHY`. Cela signifie qu'au moins un composant doit avoir été exécuté avec succès. Vous pouvez cliquer sur l'état `HEALTHY` pour

afficher la dernière instance du composant de pipeline exécutée avec succès sur la page Execution Details.

`ERROR`

L'état de santé d'au moins un composant du pipeline est `ERROR`. Vous pouvez cliquer sur l'état `ERROR` pour afficher la dernière instance du composant de pipeline en échec sur la page Execution Details.

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

Aucun état de santé n'a été signalé pour ce pipeline.

> **Note**
>
> Même si la mise à jour de l'état de santé des composants est presque immédiate, celle d'un pipeline peut prendre jusqu'à cinq minutes.

États de santé des composants

`HEALTHY`

Un composant (`Activity` ou `DataNode`) affiche un état de santé `HEALTHY` s'il a été exécuté avec succès et que son état était `FINISHED` ou `MARK_FINISHED`. Vous pouvez cliquer sur le nom du composant ou sur l'état `HEALTHY` pour afficher les toutes dernières instances du composant de pipeline exécutées avec succès sur la page Execution Details.

`ERROR`

Une erreur s'est produite au niveau du composant ou l'une de ses conditions préalables a échoué. Les états `FAILED`, `TIMEOUT` ou `CANCELED` déclenchent cette erreur. Vous pouvez cliquer sur le nom du composant ou sur l'état `ERROR` pour afficher la toute dernière instance du composant de pipeline en échec sur la page Execution Details.

`No Completed Executions` ou `No Health Information Available`

Aucun état de santé n'a été signalé pour ce composant.

Affichage de vos définitions de pipeline

Utilisez l'interface de ligne de commande (CLI) pour afficher la définition de votre pipeline. La CLI imprime un fichier de définition de pipeline au format JSON. Pour plus d'informations sur la syntaxe et l'utilisation des fichiers de définition de pipeline, consultez Syntaxe du fichier de définition du pipeline.

Lorsque vous utilisez l'interface de ligne de commande, il est conseillé de récupérer la définition du pipeline avant de soumettre des modifications, car il est possible qu'un autre utilisateur ou processus ait modifié la définition du pipeline après votre dernière utilisation. En téléchargeant une copie de la définition effective et en l'utilisant comme base pour vos modifications, vous pouvez être sûr que vous utilisez la toute dernière définition de pipeline. Il est également conseillé de récupérer la définition de pipeline une nouvelle fois après l'avoir modifiée, vous pouvez ainsi vous assurer que la mise à jour a réussi.

Lorsque vous utilisez l'interface de ligne de commande, vous pouvez obtenir deux versions différentes de votre pipeline. La version `active` est celle du pipeline actuellement en cours d'exécution. La version `latest` est une copie qui est créée lorsque vous modifiez un pipeline en cours d'exécution. Lorsque vous téléchargez le pipeline modifié, la version devient `active` et la version précédente `active` n'est plus disponible.

Pour obtenir la définition d'un pipeline à l'aide de l'AWS CLI

Pour obtenir la définition complète du pipeline, utilisez la get-pipeline-definitioncommande. La définition du pipeline est imprimée dans la sortie standard (stdout).

L'exemple suivant permet d'obtenir la définition du pipeline pour le pipeline spécifié.

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

Pour récupérer une version spécifique d'un pipeline, utilisez l'option `--version`. L'exemple suivant récupère la version `active` du pipeline spécifié.

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

Affichage des détails des instances de pipeline

Vous pouvez surveiller la progression de votre pipeline. Pour plus d'informations sur l'état de l'instance, consultez Interprétation des détails sur l'état du pipeline. Pour plus d'informations sur le dépannage des exécutions d'instance en échec ou incomplètes de votre pipeline, consultez Résolution des problèmes courants.

Pour surveiller la progression d'un pipeline à l'aide de l'AWS CLI

Pour récupérer les détails d'instances de pipeline, comme un historique des exécutions d'un pipeline, utilisez la commande list-runs. Cette commande vous permet de filtrer la liste des exécutions renvoyées en fonction de leur état effectif ou de la plage de dates dans laquelle elles ont été lancées. Le filtrage des résultats est utile car, en fonction de l'âge et de la planification du pipeline, l'historique des exécutions peut être large.

L'exemple suivant récupère des informations pour toutes les exécutions.

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

L'exemple suivant récupère des informations pour toutes les exécutions terminées.

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

L'exemple suivant récupère des informations pour toutes les exécutions lancées dans la période spécifiée.

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

Affichage des journaux de pipelines

La journalisation au niveau du pipeline est prise en charge lors de la création du pipeline en spécifiant un emplacement Amazon S3 dans la console ou en utilisant un `pipelineLogUri` dans l'objet par défaut du SDK/CLI. La structure du répertoire de chaque pipeline au sein de cette URI est semblable à ce qui suit :

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

Pour le pipeline, `df-00123456ABC7DEF8HIJK`, la structure du répertoire est similaire à :

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

Pour l'activité `ShellCommandActivity`, les journaux `stderr` et `stdout` associés à ces activités sont stockés dans le répertoire de chaque tentative.

Pour les ressources telles que, `EmrCluster`, où une `emrLogUri` est définie, cette valeur est prioritaire. Sinon, les ressources (y compris TaskRunner les journaux de ces ressources) suivent la structure de journalisation du pipeline ci-dessus.

Pour afficher les journaux d'une exécution de pipeline donnée :

1. Récupérez le `ObjectId` en appelant `query-objects` pour obtenir l'ID exact de l'objet. Par exemple :

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

   `query-objects`est une CLI paginée qui peut renvoyer un jeton de pagination s'il y a plus d'exécutions pour la ligne donnée. `pipeline-id` Vous pouvez utiliser le jeton pour parcourir toutes les tentatives jusqu'à ce que vous trouviez l'objet attendu. Par exemple, un retour ObjectId ressemblerait à :`@TableBackupActivity_2023-05-020T18:05:18_Attempt=1`.
2. À l'aide duObjectId, récupérez l'emplacement du journal en utilisant :

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

Message d'erreur indiquant l'échec d'une activité

Pour obtenir le message d'erreur, commencez par ObjectId utiliser`query-objects`.

Après avoir récupéré le message d'échecObjectId, utilisez la `describe-objects` CLI pour obtenir le message d'erreur réel.

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

Annuler, exécuter à nouveau ou marquer un objet comme terminé

Utilisez l'`set-status`interface de ligne de commande pour annuler un objet en cours d'exécution, réexécuter un objet défaillant ou marquer un objet en cours d'exécution comme terminé.

Tout d'abord, récupérez l'ID de l'objet à l'aide de l'`query-objects`interface de ligne de commande. Par exemple :

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

Utilisez l'`set-status`interface de ligne de commande pour modifier l'état de l'objet souhaité. Par exemple :

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

Modification de votre pipeline

Pour modifier certains aspects de l'un de vos pipelines, vous pouvez mettre à jour sa définition de pipeline. Après la modification d'un pipeline en cours d'exécution, vous devez réactiver le pipeline pour que vos modifications prennent effet. De plus, vous pouvez exécuter une nouvelle fois un ou plusieurs composants du pipeline.

Table des matières

Limites

Tant que le pipeline est dans l'`PENDING`état et qu'il n'est pas activé, vous ne pouvez pas le modifier. Après l'activation d'un pipeline, vous pouvez le modifier avec les restrictions suivantes. Les modifications apportées s'appliquent à de nouvelles exécutions d'objets de pipeline une fois que vous les enregistrez, puis activez le pipeline une nouvelle fois.

- Vous ne pouvez pas supprimer un objet.
- Vous ne pouvez pas modifier la période de planification d'un objet existant.
- Vous ne pouvez pas ajouter, supprimer ou modifier les champs de référence d'un objet existant.
- Vous ne pouvez pas référencer un objet existant dans un champ de sortie de nouvel objet.
- Vous ne pouvez pas modifier la date de début planifiée d'un objet (au lieu de cela, activez le pipeline avec une date et une heure spécifiques)

Modification d'un pipeline à l'aide de l'AWS CLI

Vous pouvez modifier un pipeline à l'aide des outils de ligne de commande.

Tout d'abord, téléchargez une copie de la définition actuelle du pipeline à l'aide de la get-pipeline-definitioncommande. Vous pouvez ainsi être sûr que vous modifiez la toute dernière définition du pipeline. L'exemple suivant imprime la définition de pipeline dans la sortie standard (stdout).

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

Enregistrez la définition de pipeline dans un fichier et modifiez-la si nécessaire. Mettez à jour la définition de votre pipeline à l'aide de la put-pipeline-definitioncommande. L'exemple suivant télécharge le fichier de définition de pipeline mis à jour.

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

Vous pouvez récupérer la définition de pipeline une nouvelle fois à l'aide de la commande `get-pipeline-definition` pour vous assurer que la mise à jour a réussi. Pour activer le pipeline, utilisez la commande activate-pipeline :

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

Si vous préférez, vous pouvez activer le pipeline à partir d'une date et d'une heure spécifiques, à l'aide de l'option `--start-timestamp` comme suit :

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

Pour exécuter une nouvelle fois un ou plusieurs composants de pipeline, utilisez la commande set-status.

Clonage de votre pipeline

Le clonage effectue une copie d'un pipeline et vous permet de spécifier un nom pour le nouveau pipeline. Vous pouvez cloner un pipeline qui se trouve dans n'importe quel état, même si celui-ci contient des erreurs ; toutefois, le nouveau pipeline reste dans l'état `PENDING` tant que vous ne

l'activez pas manuellement. Pour le nouveau pipeline, l'opération de clonage utilise la dernière version de la définition de pipeline d'origine plutôt que la version active. Dans l'opération de clonage, la planification complète du pipeline d'origine n'est pas copiée dans le nouveau pipeline, uniquement le paramètre de période.

Pour cloner un pipeline à l'aide de l'AWSinterface de ligne de commande :

1. Créez un nouveau pipeline avec un nouveau nom et un identifiant unique. Notez l'ID du pipeline renvoyé.
2. Utilisez l'`get-pipeline-definition`interface de ligne de commande pour obtenir la définition du pipeline existant à cloner et écrivez-la dans un fichier temporaire. Notez le chemin absolu du fichier.
3. Utilisez la `put-pipeline-definition` CLI pour copier la définition du pipeline du pipeline existant vers le nouveau pipeline.
4. Utilisez la `get-pipeline-definition` CLI pour obtenir la définition du nouveau pipeline afin de vérifier la définition du 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
```

Balisage de votre pipeline

Les balises sont des paires clé-valeur sensibles à la casse qui se composent d'une clé et d'une valeur facultative, toutes les deux définies par l'utilisateur. Vous pouvez appliquer jusqu'à dix balises

à chaque pipeline. Les clés de balise doivent être uniques pour chaque pipeline. Si vous ajoutez une balise avec une clé qui est déjà associée au pipeline, cela met à jour la valeur de cette balise.

L'application d'une balise à un pipeline propage également les balises vers ses ressources sous-jacentes (par exemple, les clusters Amazon EMR et les instances Amazon EC2). Toutefois, cela n'applique pas ces balises aux ressources dans un état `FINISHED` ou dans un état résilié. Vous pouvez utiliser l'interface de ligne de commande pour appliquer des balises à ces ressources, si nécessaire.

Lorsque vous avez fini avec une balise, vous pouvez la supprimer de votre pipeline.

Pour baliser votre pipeline à l'aide de l'interface de ligne de commande AWS

Pour ajouter des balises à un nouveau pipeline, ajouter l'option `--tags` à votre commande create-pipeline. Par exemple, l'option suivante crée un pipeline avec deux balises, une balise `environment` avec une valeur `production` et une balise `owner` avec une valeur `sales`.

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

Pour ajouter des balises à un pipeline existant, utilisez la commande add-tags comme suit :

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

Pour supprimer des balises d'un pipeline existant, utilisez la commande remove-tags comme suit :

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

Désactivation de votre pipeline

La désactivation d'un pipeline en cours d'exécution suspend l'exécution du pipeline. Pour reprendre l'exécution du pipeline, vous pouvez activer ce dernier. Vous pouvez ainsi apporter des modifications. Par exemple, si vous écrivez des données dans une base de données planifiée pour subir une maintenance, vous pouvez désactiver le pipeline, attendre la fin de la maintenance, puis activer le pipeline.

Lorsque vous désactivez un pipeline, vous pouvez spécifier ce qui se passe pour les activités en cours d'exécution. Par défaut, ces activités sont annulées immédiatement. Vous pouvez également

devoir faire patienter AWS Data Pipeline jusqu'à ce que les activités se terminent avant de désactiver le pipeline.

Lorsque vous activez la désactivation d'un pipeline, vous pouvez spécifier lorsqu'elle doit reprendre. A l'aide de l'AWS CLI ou de l'API, le pipeline reprend par défaut à la fin de la dernière exécution, ou vous pouvez spécifier la date et l'heure de reprise du pipeline.

Désactivation de votre pipeline à l'aide de l'AWS CLI

Utilisez la commande deactivate-pipeline suivante pour désactiver un pipeline :

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

Pour désactiver le pipeline uniquement à la fin de toutes les activités en cours d'exécution, ajoutez l'option `--no-cancel-active`, comme suit :

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

Lorsque vous êtes prêt, vous pouvez reprendre l'exécution du pipeline là où elle s'était arrêtée à l'aide de la commande activate-pipeline suivante :

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

Pour démarrer le pipeline à partir d'une date et d'une heure spécifiques, ajoutez l'option `--start-timestamp`, comme suit :

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

Suppression de votre pipeline

Lorsque vous n'avez plus besoin d'un pipeline, comme un pipeline créé lors des tests d'application, vous devez le supprimer pour qu'il ne puisse plus être utilisé activement. La suppression d'un pipeline le met dans un état de suppression. Lorsque le pipeline est dans l'état supprimé, sa définition de pipeline et l'historique des exécutions ont été supprimés. Par conséquent, vous ne pouvez plus effectuer d'opérations sur le pipeline, y compris le décrire.

> **Important**
>
> Vous ne pouvez pas restaurer un pipeline après l'avoir supprimé, soyez donc sûr que vous n'en aurez plus besoin à l'avenir avant de le supprimer.

Pour supprimer un pipeline à l'aide de l'AWS CLI

Pour supprimer un pipeline, utilisez la commande delete-pipeline. La commande suivante supprime le pipeline spécifié.

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

Copie intermédiaire des données et des tables avec les activités de pipeline

AWS Data Pipeline peut effectuer une copie intermédiaire des données entrantes et sortantes dans vos pipelines pour faciliter l'utilisation de certaines activités, telles que `ShellCommandActivity` et `HiveActivity`.

Une copie intermédiaire vous permet de copier des données du nœud d'entrée vers la ressource exécutant l'activité et, de la même manière, de la ressource vers le nœud de sortie.

Les données intermédiaires sur la ressource Amazon EMR ou Amazon EC2 sont disponibles à l'aide de variables spéciales dans les commandes shell ou les scripts Hive de l'activité.

La copie intermédiaire de tables est similaire au déploiement de données, à la différence près que les données ayant fait l'objet d'une copie intermédiaire prennent la forme de tables de base de données, tout spécialement.

AWS Data Pipeline prend en charge les scénarios de copie intermédiaire suivants :

- Copie intermédiaire de données avec `ShellCommandActivity`
- Copie intermédiaire de tables avec Hive et nœuds de données pris en charge par la copie intermédiaire
- Copie intermédiaire de tables avec Hive et nœuds de données non pris en charge par la copie intermédiaire

> **Note**
>
> La copie intermédiaire fonctionne uniquement lorsque le champ `stage` a la valeur `true` sur une activité, telle que `ShellCommandActivity`. Pour plus d'informations, veuillez consulter ShellCommandActivity.

En outre, les nœuds de données et les activités peuvent avoir des liens de quatre manières :

Copie intermédiaire de données localement sur une ressource

Les données d'entrée sont automatiquement copiées dans le système de fichiers local des ressources. Les données de sortie sont automatiquement copiées du système de fichiers local des ressources vers le nœud de données de sortie. Par exemple, lorsque vous configurez des entrées et des sorties `ShellCommandActivity` avec la copie intermédiaire = true, les données d'entrée sont disponibles en tant qu'INPUTx_STAGING_DIR et les données de sortie sont disponibles en tant qu'OUTPUTx_STAGING_DIR, où x correspond au nombre d'entrée ou de sortie.

Copie intermédiaire des définitions d'entrée et de sortie pour une activité

Le format des données d'entrée (noms des colonnes et noms des tables) est automatiquement copié dans la ressource de l'activité. Par exemple, lorsque vous configurez `HiveActivity` avec la copie intermédiaire = true. Le format de données spécifié sur l'entrée `S3DataNode` est utilisé pour effectuer une copie intermédiaire de la définition de table à partir de la table Hive.

Copie intermédiaire non activée

Les objets d'entrée et de sortie et leurs champs sont disponibles pour l'activité, mais les données elles-mêmes ne le sont pas. Par exemple, `EmrActivity` par défaut ou lorsque vous configurez d'autres activités avec copie intermédiaire = false. Dans cette configuration, les champs de données sont disponibles pour l'activité afin d'y faire référence à l'aide de la syntaxe des expressions AWS Data Pipeline, et ceci se produit uniquement lorsque la dépendance est satisfaite. Ceci sert à vérifier la dépendance uniquement. Le code de l'activité est responsable de la copie des données de l'entrée vers la ressource exécutant l'activité.

Relation de dépendance entre les objets

Il existe une relation de dépendance entre deux objets, ce qui donne lieu à une situation similaire à celle lorsque la copie intermédiaire n'est pas activée. Ceci entraîne un nœud de données ou une activité à agir en tant que condition préalable pour l'exécution d'une autre activité.

Stationnage des données avec ShellCommandActivity

Prenons l'exemple d'un scénario qui utilise une `ShellCommandActivity` avec des objets `S3DataNode` en tant qu'entrée et sortie de données. AWS Data Pipeline effectue automatiquement une copie intermédiaire des nœuds de données pour les rendre accessibles à la commande shell comme s'ils étaient des dossiers de fichiers locaux en utilisant les variables d'environnement `${INPUT1_STAGING_DIR}` et `${OUTPUT1_STAGING_DIR}`, comme indiqué dans l'exemple suivant. La partie numérique des variables nommées `INPUT1_STAGING_DIR` et `OUTPUT1_STAGING_DIR` s'incrémente en fonction du nombre de nœuds de données référencés par votre activité.

> **Note**
>
> Ce scénario fonctionne uniquement comme indiqué si vos entrées et sorties de données sont des objets `S3DataNode`. De plus, la copie intermédiaire des données de sortie est autorisée uniquement lorsque le `directoryPath` est défini sur l'objet `S3DataNode` de sortie.

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

Copie intermédiaire de tables avec Hive et nœuds de données pris en charge par la copie intermédiaire

Prenons l'exemple d'un scénario qui utilise une `HiveActivity` avec des objets `S3DataNode` en tant qu'entrée et sortie de données. AWS Data Pipeline effectue automatiquement une copie intermédiaire des nœuds de données pour les rendre accessibles au script Hive comme s'ils étaient des tables Hive en utilisant les variables `${input1}` et `${output1}`, comme indiqué dans l'exemple suivant pour `HiveActivity`. La partie numérique des variables nommées `input` et `output` s'incrémente en fonction du nombre de nœuds de données référencés par votre activité.

> **Note**
>
> Ce scénario fonctionne uniquement comme indiqué si vos entrées et sorties de données sont des objets `S3DataNode` ou `MySqlDataNode`. La copie intermédiaire de tables n'est pas prise en charge pour `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"
  }
},
...
```

Copie intermédiaire de tables avec Hive et nœuds de données non pris en charge par la copie intermédiaire

Prenons l'exemple d'un scénario qui utilise une `HiveActivity` avec `DynamoDBDataNode` en tant qu'entrée de données et un objet `S3DataNode` en tant que sortie. Aucun transfert de données n'est disponible. Vous devez donc d'abord créer la table manuellement dans votre script Hive, en utilisant le nom de la variable `#{input.tableName}` pour faire référence à la table DynamoDB. `DynamoDBDataNode` Une nomenclature similaire s'applique si la table DynamoDB est la sortie, sauf que vous utilisez une variable. `#{output.tableName}` La copie intermédiaire est disponible pour l'objet `S3DataNode` de sortie dans cet exemple, vous pouvez donc faire référence au nœud de données de sortie en tant que `${output1}`.

> **Note**
>
> Dans cet exemple, la variable du nom de la table comporte un préfixe # (hachage), car AWS Data Pipeline utilise des expressions pour accéder au `tableName` ou au `directoryPath`.

Pour plus d'informations sur le fonctionnement de l'évaluation des expressions dans AWS Data Pipeline, consultez Evaluation d'expression.

```
{
  "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"
  }
```

```
},
...
```

Utilisation d'un pipeline avec des ressources dans plusieurs régions

Par défaut, les ressources `Ec2Resource` et `EmrCluster` s'exécutent dans la même région qu'AWS Data Pipeline. Toutefois, AWS Data Pipeline prend en charge la possibilité d'orchestrer les flux de données dans plusieurs régions, comme l'exécution des ressources d'une région consolidant les données d'entrée d'une autre région. En permettant aux ressources de s'exécuter dans une région spécifiée, vous avez également la possibilité de colocaliser vos ressources avec leurs jeux de données dépendants et d'optimiser les performances en réduisant les latences et en évitant les frais de transfert de données entre régions. Vous pouvez configurer des ressources pour qu'elles s'exécutent dans une autre région que celle d'AWS Data Pipeline en utilisant le champ `region` sur `Ec2Resource` et `EmrCluster`.

L'exemple de fichier JSON de pipeline suivant montre comment exécuter une `EmrCluster` ressource dans la région Europe (Irlande), en supposant qu'une grande quantité de données sur lesquelles le cluster doit travailler existe dans la même région. Dans cet exemple, la seule différence avec un pipeline typique est que l'`EmrCluster` comporte une valeur de champ `region` définie sur `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"
    }
  ]
}
```

Le tableau suivant répertorie les régions que vous pouvez choisir et les codes de région associés à utiliser dans le champ `region`.

> **Note**
>
> La liste suivante inclut les régions dans lesquelles AWS Data Pipeline vous pouvez orchestrer des flux de travail et lancer des ressources Amazon EMR ou Amazon EC2. AWS Data Pipelinepeut ne pas être pris en charge dans ces régions. Pour plus d'informations sur les régions dans lesquelles AWS Data Pipeline est pris en charge, consultez Régions et points de terminaison AWS.

Nom de la région	Code région
US East (Virginie du Nord)	us-east-1
USA Est (Ohio)	us-east-2
USA Ouest (Californie du Nord)	us-west-1
US West (Oregon)	us-west-2
Canada (Centre)	ca-central-1

Nom de la région	Code région
Europe (Irlande)	eu-west-1
Europe (Londres)	eu-west-2
Europe (Francfort)	eu-central-1
Asie-Pacifique (Singapore)	ap-southeast-1
Asie-Pacifique (Sydney)	ap-southeast-2
Asie-Pacifique (Mumbai)	ap-south-1
Asie Pacifique (Tokyo)	ap-northeast-1
Asie-Pacifique (Séoul)	ap-northeast-2
Amérique du Sud (São Paulo)	sa-east-1

Mise en cascade des échecs et des réexécutions

AWS Data Pipeline vous permet de configurer la façon dont les objets de pipeline se comportent lorsqu'une dépendance échoue ou est annulée par un utilisateur. Vous pouvez vous assurer que les échecs se mettent en cascade vers d'autres objets de pipeline (consommateurs), afin d'éviter une attente indéfinie. Toutes les activités, tous les nœuds de données et toutes les conditions préalables comportent un champ nommé `failureAndRerunMode` doté d'une valeur `none` par défaut. Pour activer les échecs en cascade, définissez le champ `failureAndRerunMode` sur `cascade`.

Lorsque ce champ est activé, les échecs en cascade se produisent si un objet de pipeline est bloqué à l'état `WAITING_ON_DEPENDENCIES` et que toutes les dépendances ont échoué sans aucune commande en attente. Lors d'un échec en cascade, les événements suivants se produisent :

- Lorsqu'un objet échoue, ses consommateurs sont définis sur `CASCADE_FAILED`, et l'objet d'origine et les conditions préalables de ses consommateurs sont définis sur `CANCELED`.
- Tous les objets qui sont déjà définis sur `FINISHED`, `FAILED` ou `CANCELED` sont ignorés.

L'échec en cascade ne fonctionne pas sur les dépendances de l'objet en échec (en amont), à l'exception des conditions préalables associées à l'objet en échec d'origine. Les objets de pipeline affectés par un échec de cascade peuvent déclencher de nouvelles tentatives ou des actions ultérieures, par exemple `onFail`.

Les effets détaillés d'un échec en cascade dépendent du type d'objet.

Activités

Une activité passe à l'état `CASCADE_FAILED` si l'une de ses dépendances échoue, puis elle déclenche un échec en cascade pour les consommateurs de l'activité. Si une ressource échoue du fait que l'activité en dépend, l'état de l'activité est `CANCELED` et tous ses consommateurs passent à l'état `CASCADE_FAILED`.

Nœuds de données et conditions préalables

Si un nœud de données est configuré en tant que sortie d'une activité qui échoue, le nœud de données passe à l'état `CASCADE_FAILED`. L'échec d'un nœud de données se propage vers toutes les conditions préalables associées, qui passent à l'état `CANCELED`.

Ressources

Si l'état des objets qui dépendent d'une ressource est `FAILED` et celui de la ressource elle-même est `WAITING_ON_DEPENDENCIES`, la ressource passe à l'état `FINISHED`.

Réexécution d'objets ayant échoué en cascade

Par défaut, la réexécution toute activité ou de tout nœud de données réexécute uniquement la ressource associée. Toutefois, en définissant le champ `failureAndRerunMode` sur `cascade` sur un objet de pipeline permet à une commande de réexécution sur un objet cible de se propager sur tous les consommateurs, dans les conditions suivantes :

- L'état des consommateurs de l'objet cible est `CASCADE_FAILED`.
- Les dépendances de l'objet cible n'ont pas de commandes de réexécution en attente.
- L'état des dépendances de l'objet cible n'est ni `FAILED`, ni `CASCADE_FAILED` ni `CANCELED`.

Si vous tentez de relancer un objet `CASCADE_FAILED` et que l'état de l'une de ses dépendances est `FAILED`, `CASCADE_FAILED` ou `CANCELED`, la réexécution échoue et renvoie l'objet à l'état `CASCADE_FAILED`. Pour réexécuter avec succès un objet en échec, vous devez retracer l'échec

jusqu'à la chaîne de dépendance pour localiser la source à l'origine de l'échec et réexécuter cet objet à la place. Lorsque vous émettez une commande de réexécution sur une ressource, vous pouvez également tenter de réexécuter tous les objets qui en dépendent.

Défaillance en cascade et remblayages

Si vous activez l'échec de cascade et que vous disposez d'un pipeline qui crée de nombreux renvois, des erreurs d'exécution de pipeline peuvent entraîner la création et la suppression de ressources en succession rapide sans effectuer un travail utile. AWS Data Pipeline tente de vous alerter de cette situation avec le message d'avertissement suivant lorsque vous enregistrez un 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. .` En effet, l'échec de cascade peut rapidement définir des activités en aval comme `CASCADE_FAILED` et arrêter les clusters EMR et les ressources EC2 inutiles. Nous vous recommandons de tester les pipelines avec de courtes plages de temps pour limiter les effets de cette situation.

Syntaxe du fichier de définition du pipeline

Les instructions contenues dans cette section concernent l'utilisation manuelle des fichiers de définition de pipeline à l'aide de l'interface de ligne de commande AWS Data Pipeline. Il s'agit d'une autre solution par rapport à la conception d'un pipeline de façon interactive à l'aide de la console AWS Data Pipeline.

Vous pouvez créer manuellement les fichiers de définition de pipeline (à l'aide de n'importe quel éditeur de texte qui prend en charge l'enregistrement des fichiers au format UTF-8) et soumettre les fichiers via l'interface de ligne de commande AWS Data Pipeline.

AWS Data Pipeline prend également en charge une grande variété d'expressions et de fonctions complexes dans les définitions de pipeline. Pour plus d'informations, veuillez consulter Expressions et fonctions de pipeline.

Structure de fichier

La première étape dans la création de la définition de pipeline consiste à composer des objets de définition de pipeline dans un fichier de définition de pipeline. L'exemple suivant illustre la structure générale d'un fichier de définition de pipeline. Ce fichier définit deux objets, qui sont délimités par « { » et « }' », et séparés par une virgule.

Dans l'exemple suivant, le premier objet définit deux paires nom-valeur, appelées champs. Le deuxième objet définit trois champs.

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

Lors de la création d'un fichier de définition de pipeline, vous devez sélectionner les types d'objets de pipeline dont vous aurez besoin, les ajouter au fichier de définition de pipeline, puis ajouter les champs appropriés. Pour en savoir plus sur les objets de pipeline, consultez Référence d'objet de pipeline.

Par exemple, vous pouvez créer un objet de définition de pipeline pour un nœud de données d'entrée et un autre pour le nœud de données de sortie. Créez ensuite un autre objet de définition de pipeline pour une activité, par exemple en traitant les données d'entrée à l'aide d'Amazon EMR.

Champs de pipeline

Une fois que vous connaissez les types d'objet à inclure dans votre fichier de définition de pipeline, ajoutez des champs à la définition de chaque objet de pipeline. Les noms de champs sont entre guillemets, et séparés des valeurs de champs par un espace, deux points et un espace, comme illustré dans l'exemple suivant.

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

La valeur de champ peut être une chaîne de texte, une référence à un autre objet, un appel de fonction, une expression ou une liste triée de l'un des types précédents. Pour plus d'informations sur les types de données qui peuvent être utilisés pour les valeurs de champs, consultez Types de données simples. Pour plus d'informations sur les fonctions que vous pouvez utiliser pour évaluer les valeurs de champs, consultez Evaluation d'expression.

Les champs sont limités à 2 048 caractères. Les objets peuvent être de 20 Ko, ce qui signifie que vous ne pouvez pas ajouter plusieurs larges champs à un objet.

Chaque objet de pipeline doit contenir les champs suivants : `id` et `type`, comme indiqué dans l'exemple suivant. D'autres champs peuvent également être requis en fonction du type d'objet. Sélectionnez une valeur `id` qui est significative pour vous et unique dans la définition de pipeline. La valeur `type` spécifie le type de l'objet. Spécifiez l'un des types d'objet de définition de pipeline pris en charge, qui sont énumérés dans la rubrique Référence d'objet de pipeline.

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

Pour plus d'informations sur les champs obligatoires et facultatifs de chaque objet, consultez la documentation de l'objet.

Pour inclure des champs d'un objet dans un autre objet, utilisez le champ `parent` avec une référence à l'objet. Par exemple, l'objet « B » comprend ses champs, « B1 » et « B2 », ainsi que les champs d'objet « A », « A1 » et « A2 ».

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

Vous pouvez définir des champs courants dans un objet avec l'ID « Par défaut ». Ces champs sont automatiquement inclus dans chaque objet du fichier de définition de pipeline qui ne définit pas explicitement son champ `parent` pour référencer un autre objet.

```
{
  "id" : "Default",
  "onFail" : {"ref" : "FailureNotification"},
```

```
  "maximumRetries" : "3",
  "workerGroup" : "myWorkerGroup"
}
```

Champs définis par l'utilisateur

Vous pouvez créer des champs définis par l'utilisateur ou des champs personnalisés sur vos composants de pipeline et faites-y référence avec des expressions. L'exemple suivant montre un champ personnalisé nommé `myCustomField` et `my_customFieldReference` ajouté à un DataNode objet 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"}
  },
```

Un champ défini par l'utilisateur doit comporter un nom préfixé avec le mot « my » en minuscules, suivi d'une majuscule ou d'un caractère de soulignement. De plus, un champ défini par l'utilisateur peut être une valeur chaîne comme l'exemple `myCustomField` précédent, ou une référence à un autre composant de pipeline comme l'exemple `my_customFieldReference` précédent.

> **Note**
>
> Sur les champs définis par l'utilisateur, AWS Data Pipeline ne vérifie que les références valides à d'autres composants de pipeline, et non toutes les valeurs de chaîne de champ personnalisé que vous ajoutez.

Utilisation de l'API

> **Note**
>
> Si vous n'écrivez pas de programmes qui interagissent avec AWS Data Pipeline, vous n'avez besoin d'installer aucun des kits SDK AWS. Vous pouvez créer et exécuter des

pipelines à l'aide de la console ou de l'interface de ligne de commande. Pour de plus amples informations, veuillez consulterConfiguration pour AWS Data Pipeline

La manière la plus simple d'écrire des applications qui interagissent avecAWS Data Pipelineou d'implémenter une tâche Runner personnalisée consiste à utiliser l'un des kits SDK AWS. Les kits SDK AWS fournissent une fonctionnalité qui simplifie l'appel des API de service web à partir de votre environnement de programmation préféré. Pour plus d'informations, consultez Installation du kit SDK AWS.

Installation du kit SDK AWS

Les kits SDK AWS fournissent des fonctions qui enveloppent l'API et prennent soin d'un grand nombre des détails de connexion, tels que le calcul des signatures, la gestion des nouvelles tentatives de demande et la gestion des erreurs. Les kits SDK contiennent également des exemples de code, des didacticiels et d'autres ressources pour vous aider à commencer à écrire des applications qui appellent AWS. L'appel des fonctions de wrapper dans un kit SDK peut considérablement simplifier le processus d'écriture d'une application AWS. Pour plus d'informations sur la façon de télécharger et d'utiliser les kits SDK AWS, consultez Exemples de codes et bibliothèques.

La prise en charge d'AWS Data Pipeline est disponible dans les kits SDK pour les plates-formes suivantes :

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

Envoi d'une demande HTTP à AWS Data Pipeline

Pour obtenir une description complète des objets de programmation dans AWS Data Pipeline, consultez le manuel Référence d'API AWS Data Pipeline.

Si vous n'utilisez pas l'un des kits SDK AWS, vous pouvez effectuer des opérations AWS Data Pipeline via HTTP à l'aide de la méthode POST. La méthode POST exige que vous indiquiez l'opération dans l'en-tête de la demande et que vous fournissiez les données de l'opération au format JSON dans le corps de la demande.

Contenu de l'en-tête HTTP

AWS Data Pipeline nécessite les informations suivantes dans l'en-tête d'une demande HTTP :

- `host` Le point de terminaison AWS Data Pipeline.

  Pour plus d'informations sur les points de terminaison, consultez Régions et points de terminaison.
- `x-amz-date` Vous devez fournir l'horodatage dans l'en-tête HTTP Date ou dans l'en-tête AWS x-amz-date (certaines bibliothèques client HTTP ne vous permettent pas de définir l'en-tête Date). Lorsqu'un en-tête x-amz-date est présent, le système ignore l'en-tête Date lors de l'authentification de la demande.

  La date doit être spécifiée dans l'un des formats suivants, comme indiqué dans le RFC HTTP/1.1 :
  - Sun, 06 Nov 1994 08:49:37 GMT (RFC 822, mis à jour par RFC 1123)
  - Sunday, 06-Nov-94 08:49:37 GMT (RFC 850, rendu obsolète par RFC 1036)
  - Sun Nov 6 08:49:37 1994 (format ANSI C asctime())
- `Authorization` Ensemble de paramètres d'autorisation qu'AWS utilise pour garantir la validité et l'authenticité de la demande. Pour plus d'informations sur la construction de cet en-tête, consultez Processus de signature Signature Version 4.
- `x-amz-target` Service de destination de la demande et de l'opération sur les données, au format suivant : `<<serviceName>>_<<API version>>.<<operationName>>`

  Par exemple, `DataPipeline_20121129.ActivatePipeline`
- `content-type` Spécifie JSON et la version. Par exemple, `Content-Type: application/x-amz-json-1.0`

Voici un exemple d'en-tête de demande HTTP d'activation d'un pipeline :

```
POST / HTTP/1.1
host: https://datapipeline.us-east-1.amazonaws.com
x-amz-date: Mon, 12 Nov 2012 17:49:52 GMT
```

```
x-amz-target: DataPipeline_20121129.ActivatePipeline
Authorization: AuthParams
Content-Type: application/x-amz-json-1.1
Content-Length: 39
Connection: Keep-Alive
```

Contenu du corps HTTP

Le corps d'une requête HTTP contient les données de l'opération spécifiée dans l'en-tête de la requête HTTP. Les données doivent être mises en forme conformément au schéma de données JSON pour chaque API AWS Data Pipeline. Le schéma de données JSON d'AWS Data Pipeline définit les types de données et les paramètres (tels que les opérateurs de comparaison et les constantes d'énumération) qui sont disponibles pour chaque opération.

Mise en forme du corps d'une demande HTTP

Utilisez le format de données JSON pour transmettre simultanément les valeurs de données et la structure de données. Pour imbriquer des éléments dans d'autres, utilisez la notation d'accolade. L'exemple suivant montre une demande de placement d'une définition de pipeline composée de trois objets et des emplacements correspondants.

```
{
 "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"}

      ]
    }
  ]
}
```

Gestion de la réponse HTTP

Voici quelques en-têtes importants de la réponse HTTP et la façon dont vous devez les gérer dans votre application :

- HTTP/1.1 : cet en-tête est suivi d'un code d'état. La valeur de code 200 indique une opération réussie. Toute autre valeur indique une erreur.
- x-amzn-RequestId : cet en-tête contient un ID de demande que vous pouvez utiliser pour résoudre une demande avec AWS Data Pipeline. K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG est un exemple d'ID de demande.
- x-amz-crc32 : AWS Data Pipeline calcule le total de contrôle CRC32 de la charge utile HTTP et renvoie ce total de contrôle dans un en-tête x-amz-crc32. Nous vous recommandons de calculer votre propre total de contrôle CRC32 côté client et de le comparer avec l'en-tête x-amz-crc32 ; si les totaux de contrôle ne correspondent pas, cela peut indiquer que les données ont été corrompues lors du transit. Si cela se produit, vous devez relancer votre demande.

Les utilisateurs de kits SDK AWS n'ont pas besoin d'effectuer manuellement cette vérification, car les kits SDK calculent le total de contrôle de chaque réponse à partir d'Amazon DynamoDB et font automatiquement une nouvelle tentative si un décalage est détecté.

Exemple de demande et de réponse JSON AWS Data Pipeline

L'exemple suivant montre une demande de création d'un pipeline. Il montre ensuite la réponse d'AWS Data Pipeline, comprenant notamment l'identifiant du pipeline venant d'être créé.

Demande HTTP POST

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

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

Réponse d'AWS Data Pipeline

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

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

Sécurité dans AWS Data Pipeline

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le cloud AWS. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des AWS programmes de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Data Pipeline, veuillez consulter Services AWS concernés par le programme de conformité.
- Sécurité dans le cloud : votre responsabilité est déterminée par le service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de AWS Data Pipeline. Les rubriques suivantes expliquent comment configurer AWS Data Pipeline pour répondre à vos objectifs de sécurité et de conformité. Vous pouvez également apprendre à utiliser d'autres services AWS capables de vous aider à surveiller et à sécuriser vos ressources AWS Data Pipeline.

Rubriques

Protection des données dans AWS Data Pipeline

Le modèle de responsabilité partagée AWS s'applique à la protection des données dans AWS Data Pipeline. Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure globale sur laquelle l'ensemble du AWS Cloud s'exécute. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Ce contenu comprend les tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour en savoir plus sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWSet RGPD (Règlement général sur la protection des données) sur le AWSBlog de sécurité.

À des fins de protection des données, nous vous recommandons de protéger les informations d'identification Compte AWS et de configurer les comptes utilisateur individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez les certificats SSL/TLS pour communiquer avec les ressources AWS. Nous vous recommandons le certificats TLS 1.2 ou une version ultérieure.
- Configurez une API (Interface de programmation) et le journal de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés FIPS (Federal Information Processing Standard) 140-2 lorsque vous accédez à AWS via une CLI (Interface de ligne de commande) ou une API (Interface de programmation), utilisez un point de terminaison FIPS (Federal Information Processing Standard). Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).
- AWS Data Pipelineprend en charge les ressources IMDSv2 pour Amazon EMR et Amazon EC2. Pour utiliser IMDSv2 avec Amazon EMR, utilisez les versions 5.23.1, 5.27.1 ou 5.32 ou ultérieures

ou la version 6.2 ou ultérieure. Pour plus d'informations, consultez Configurer les demandes de service de métadonnées pour les instances Amazon EC2 et Utiliser IMDSv2.

Il est fortement recommandé de ne jamais indiquer d'informations confidentielles ou sensibles, telles que des adresses électroniques de vos clients, dans des balises ou des champs de texte de forme libre comme un champ Name (Nom). Cela est également valable lorsque vous utilisez AWS Data Pipeline ou d'autres Services AWS à l'aide de la console, de l'API, d'AWS CLI ou des kits SDK AWS. Toutes les données que vous saisissez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Identity and Access Management (Gestion des identités et des accès) pour AWS Data Pipeline

Vos informations d'identification de sécurité vous identifient auprès des services dans AWS et vous accordent l'autorisation d'utiliser les ressources AWS, telles que vos pipelines. Vous pouvez utiliser les fonctions d'AWS Data Pipelineet d'AWS Identity and Access Management(IAM) pour permettreAWS Data Pipeline aux autres utilisateurs d'accéder à vosAWS Data Pipeline ressources sans partager vos informations d'identification de sécurité.

Les organisations peuvent partager l'accès aux pipelines afin de permettre aux personnes d'une organisation de développer et gérer ceux-ci de manière collaborative. Cependant, par exemple, il peut s'avérer nécessaire d'effectuer les actions suivantes :

- Contrôlez quels utilisateurs peuvent accéder à des pipelines spécifiques
- Protéger un pipeline de production afin d'éviter qu'il soit modifié par erreur
- Autoriser un auditeur à avoir un accès en lecture seule aux pipelines, mais l'empêcher d'effectuer des modifications

AWS Data Pipelineest intégré àAWS Identity and Access Management (IAM), qui offre un large éventail de fonctionnalités :

- Créez des utilisateurs et des groupes dans votreCompte AWS.
- Partagez facilement vosAWS ressources entre les utilisateurs de votreCompte AWS.

- Attribuer des informations d'identification de sécurité uniques à chaque utilisateur.
- Contrôler l'accès de chaque utilisateur aux services et ressources.
- Obtenez une facture unique pour tous les utilisateurs de votreCompte AWS.

Grâce à l'utilisation d'avecAWS Data Pipeline, vous pouvez contrôler si les utilisateurs de votre organisation peuvent exécuter une tâche à l'aide d'actions d'API particulières et s'ils peuvent utiliser les ressources AWS spécifiques. Vous pouvez utiliser des politiques IAM basées sur des balises de pipeline et des groupes de travail pour partager vos pipelines avec d'autres utilisateurs et contrôler leur niveau d'accès.

Table des matières

Stratégies IAM pour AWS Data Pipeline

Les entités IAM ne sont pas autorisées, par défaut, à créer ou modifier les ressources AWS. Pour autoriser les entités IAM à créer ou à modifier des ressources et à exécuter des tâches, vous devez créer les stratégies IAM qui accordent aux entités IAM l'autorisation d'utiliser les actions d'API et ressources spécifiques dont elles ont besoin, puis d'attacher ces stratégies aux entités IAM qui requièrent ces autorisations.

Quand vous attachez une politique à un utilisateur ou à un groupe d'utilisateurs, elle accorde ou refuse aux utilisateurs l'autorisation d'exécuter les tâches spécifiées sur les ressources spécifiées. Pour plus d'informations générales sur les stratégies IAM, consultez Autorisations et stratégies dans le guide de l'utilisateur. Pour plus d'informations sur la gestion et la création de stratégies IAM personnalisées, consultez Gestion des stratégies IAM.

Table des matières

Syntaxe d'une stratégie

Une politique IAM est un document JSON qui se compose d'une ou de plusieurs déclarations. Chaque déclaration est structurée comme suit :

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

Les éléments suivants constituent une déclaration de stratégie :

- Effect : effect peut avoir la valeur `Allow` ou `Deny`. Comme, par défaut, les entités IAM n'ont la permission d'utiliser les ressources et les actions d'API, toutes les demandes sont refusées. Une autorisation explicite remplace l'autorisation par défaut. Un refus explicite remplace toute autorisation.
- Action : action désigne l'action d'API spécifique pour laquelle vous accordez ou refusez l'autorisation. Pour obtenir la liste des actions pourAWS Data Pipeline, consultez la section Actions dans la référence de l'AWS Data PipelineAPI.
- Resource : la ressource affectée par l'action. La seule valeur valide est `"*"`.
- Condition : les conditions sont facultatives. Elles permettent de contrôler à quel moment votre stratégie sera effective.

  AWS Data Pipeline implémente les clés de contexte à l'échelle d'AWS (consultez Clés de condition disponibles), ainsi que les clés spécifiques au service suivantes.

  - `datapipeline:PipelineCreator`— Pour accorder l'accès à l'utilisateur qui a créé le pipeline. Pour voir un exemple, consultez Accorder l'autorisation d'accès complet au propriétaire du pipeline.
  - `datapipeline:Tag`— Pour accorder l'accès en fonction du balisage du pipeline. Pour plus d'informations, veuillez consulter Contrôle de l'accès aux pipelines à l'aide de balises.

- `datapipeline:workerGroup`— Pour accorder l'accès en fonction du nom du groupe de travail. Pour plus d'informations, veuillez consulter Contrôle de l'accès aux pipelines à l'aide de groupes de travail.

Contrôle de l'accès aux pipelines à l'aide de balises

Vous pouvez créer des stratégies IAM qui font référence aux balises de votre pipeline. Cela vous permet d'utiliser les balises de pipeline pour effectuer les actions suivantes :

- Accorder l'accès en lecture seule à un pipeline
- Accorder l'accès en lecture/écriture à un pipeline
- Bloquer l'accès à un pipeline

Par exemple, supposons qu'un gestionnaire possède deux environnements de pipeline (un de production et un de développement), ainsi qu'un groupe IAM pour chaque environnement. Pour les pipelines de l'environnement de production, le gestionnaire accorde un accès en lecture/écriture aux utilisateurs du groupe IAM de production, mais accorde un accès en lecture seule aux utilisateurs du groupe IAM de développeur. Pour les pipelines de l'environnement de développement, le gestionnaire accorde un accès en lecture/écriture aux groupes IAM de production et de développement.

Pour réaliser ce scénario, le responsable étiquette les pipelines de production avec la balise « environment=production » et associe la politique suivante au groupe IAM des développeurs. La première instruction accorde un accès en lecture seule à tous les pipelines. La seconde instruction accorde un accès en lecture/écriture aux pipelines qui n'ont pas la balise « environment=production ».

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

```
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "datapipeline:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {"datapipeline:Tag/environment": "production"}
      }
    }
  ]
}
```

En outre, le responsable associe la politique suivante au groupe IAM de production. Cette instruction accorde l'accès complet à tous les pipelines.

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

Pour voir d'autres exemples, consultez Accorder aux utilisateurs un accès en lecture seule en fonction d'une balise et Accorder aux utilisateurs l'accès complet en fonction d'une balise.

Contrôle de l'accès aux pipelines à l'aide de groupes de travail

Vous pouvez créer des politiques IAM qui attribuent des noms de groupes de travailleurs de référence.

Par exemple, supposons qu'un gestionnaire possède deux environnements de pipeline (un de production et un de développement), ainsi qu'un groupe IAM pour chaque environnement. Le gestionnaire possède trois serveurs de base de données sur lesquels les exécuteurs de tâches sont respectivement configurés pour les environnements de production, de pré-production et de développement. Le responsable souhaite s'assurer que les utilisateurs du groupe IAM de production peuvent créer des pipelines qui redirigent les tâches vers les ressources de production, et que les

utilisateurs du groupe IAM de développement peuvent créer des pipelines qui redirigent les tâches vers les ressources de pré-production et de développement.

Pour réaliser ce scénario, le gestionnaire installe l'exécuteur de tâches sur les ressources de production avec des informations d'identification de production et affecte au paramètre `workerGroup` la valeur « prodresource ». En outre, le gestionnaire installe l'exécuteur de tâches sur les ressources de développement avec des informations d'identification de développement et affecte au paramètre `workerGroup` les valeurs « pre-production » et « development ». Le responsable attache la politique suivante au groupe IAM des développeurs afin de bloquer l'accès aux ressources « prodresource ». La première instruction accorde un accès en lecture seule à tous les pipelines. La seconde instruction accorde un accès en lecture/écriture aux pipelines lorsque le nom du groupe de travail a le préfixe « dev » ou « pre-prod ».

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

En outre, le responsable attache la politique suivante au groupe IAM de production pour accorder l'accès aux ressources « prodresource ». La première instruction accorde un accès en lecture seule

à tous les pipelines. La seconde instruction accorde un accès en lecture/écriture lorsque le nom du groupe de travail a le préfixe « prod ».

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

Exemples de stratégies pour AWS Data Pipeline

Les exemples suivants montrent comment accorder aux utilisateurs un accès complet ou limité aux pipelines.

Table des matières

Exemple 1 : Accorder aux utilisateurs un accès en lecture seule en fonction d'une balise

La stratégie suivante permet aux utilisateurs d'utiliser les actions d'API AWS Data Pipeline en lecture seule, mais uniquement avec les pipelines qui ont la balise "environment=production".

L'action d' ListPipelines API ne prend pas en charge l'autorisation basée sur les balises.

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

Exemple 2 : Accorder aux utilisateurs un accès complet en fonction d'une balise

La politique suivante permet aux utilisateurs d'utiliser toutes les actions de l'AWS Data PipelineAPI, à l'exception ListPipelines, mais uniquement, des pipelines portant la balise « environment=test ».

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

```
        "datapipeline:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "datapipeline:Tag/environment": "test"
        }
      }
    }
  ]
}
```

Exemple 3 : Accorder un accès complet au propriétaire du pipeline

La stratégie suivante permet aux utilisateurs d'utiliser toutes les actions d'API AWS Data Pipeline, mais uniquement avec leurs propres pipelines.

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

Exemple 4 : Accorder aux utilisateurs l'accès à la console AWS Data Pipeline

La stratégie suivante permet aux utilisateurs de créer et de gérer un pipeline à l'aide de la console AWS Data Pipeline.

Cette stratégie inclut l'action pour les autorisations `PassRole` pour les ressources spécifiques liées au `role`ARN dont AWS Data Pipeline a besoin. Pour plus d'informations sur l'`PassRole`autorisation basée sur l'identité (IAM), consultez le billet de blog Octroi de l'autorisation de lancer des instances EC2 avec des rôles IAM (PassRoleautorisation).

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

```
}
```

Rôles IAM pour AWS Data Pipeline

AWS Data PipelineutiliseAWS Identity and Access Management des rôles. Les politiques d'autorisation associées aux rôles IAM déterminent les actionsAWS Data Pipeline que vos applications peuvent effectuer et lesAWS ressources auxquelles elles peuvent accéder. Pour de plus amples informations, veuillez consulter Rôles IAM dans le Guide de l'utilisateur IAM.

AWS Data Pipelinenécessite deux rôles IAM :

- Le rôle de pipeline contrôleAWS Data Pipeline l'accès à vos ressources AWS. Dans les définitions d'objets de pipeline, le`role` champ spécifie ce rôle.
- Le rôle d'instance EC2 contrôle l'accès des applications exécutées sur des instances EC2, y compris les instances EC2 des clusters Amazon EMR, auxAWS ressources. Dans les définitions d'objets de pipeline, le`resourceRole` champ spécifie ce rôle.

> **Important**
>
> Si vous avez créé un pipeline avant le 3 octobre 2022 à l'aide de laAWS Data Pipeline console avec des rôles par défaut,AWS Data Pipeline`DataPipelineDefaultRole` créez le pour vous et associez la politique`AWSDataPipelineRole` gérée au rôle. Depuis le 3 octobre 2022, la politique`AWSDataPipelineRole` gérée est obsolète et le rôle de pipeline doit être spécifié pour un pipeline lors de l'utilisation de la console.
> Nous vous recommandons de passer en revue les pipelines existants et de`DataPipelineDefaultRole` déterminer s'ils sont associés au pipeline et s'il`AWSDataPipelineRole` sont rattachés à ce rôle. Si tel est le cas, examinez l'accès autorisé par cette politique pour vous assurer qu'il est adapté à vos exigences de sécurité. Ajoutez, mettez à jour ou remplacez les politiques et les déclarations de stratégie associées à ce rôle si nécessaire. Vous pouvez également mettre à jour un pipeline pour utiliser un rôle que vous créez avec des politiques d'autorisation différentes.

Exemples de politiques d'autorisation pour lesAWS Data Pipeline rôles

Chaque rôle est associé à une ou plusieurs politiques d'autorisation qui déterminent lesAWS ressources auxquelles le rôle peut accéder et les actions qu'il peut effectuer. Cette rubrique fournit

un exemple de politique d'autorisations pour le rôle de pipeline. Il fournit également le contenu du`AmazonEC2RoleforDataPipelineRole`, qui est la politique gérée pour le rôle d'instance EC2 par défaut,`DataPipelineDefaultResourceRole`.

Exemple de stratégie d'autorisations de rôle de rôle de rôle

L'exemple de politique qui suit est conçu pour autoriser les fonctions essentielles quiAWS Data Pipeline nécessitent l'exécution d'un pipeline avec des ressources Amazon EC2 et Amazon EMR. Il fournit également des autorisations d'accès à d'autresAWS ressources, telles qu'Amazon Simple Storage Service et Amazon Simple Notification Service, requises par de nombreux pipelines. Si les objets définis dans un pipeline ne nécessitent pas les ressources d'unAWS service, nous vous recommandons vivement de supprimer les autorisations d'accès à ce service. Par exemple, si votre pipeline ne définit pas d'actionDynamoDB DataNode ou n'utilise pas cetteSnsAlarm action, nous vous recommandons de supprimer les instructions d'autorisation associées à ces actions.

- Remplacez *111122223333* par votre ID de compte AWS.
- *NameOfDataPipelineRole*Remplacez-le par le nom du rôle de pipeline (le rôle auquel cette politique est attachée).
- *NameOfDataPipelineResourceRole*Remplacez-le par le nom du rôle d'instance EC2.
- *us-west-1*Remplacez-le par la région appropriée à votre application.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListRolePolicies",
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/NameOfDataPipelineRole",
                "arn:aws:iam::111122223333 :role/NameOfDataPipelineResourceRole"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CancelSpotInstanceRequests",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteTags",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcEndpointServices",
                "ec2:DescribeVpcs",
                "ec2:DetachNetworkInterface",
                "ec2:ModifyImageAttribute",
                "ec2:ModifyInstanceAttribute",
                "ec2:RequestSpotInstances",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "ec2:DescribeVolumeStatus",
                "ec2:DescribeVolumes",
                "elasticmapreduce:TerminateJobFlows",
                "elasticmapreduce:ListSteps",
```

```
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:RunJobFlow",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:AddTags",
                "elasticmapreduce:RemoveTags",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ModifyInstanceGroups",
                "elasticmapreduce:GetCluster",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:AddJobFlowSteps",
                "elasticmapreduce:ListInstances",
                "iam:ListInstanceProfiles",
                "redshift:DescribeClusters"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:GetTopicAttributes",
                "sns:Publish"
            ],
            "Resource": [
                "arn:aws:sns:us-west-1:111122223333:MyFirstSNSTopic",
                "arn:aws:sns:us-west-1:111122223333:MySecondSNSTopic",
                "arn:aws:sns:us-west-1:111122223333:AnotherSNSTopic"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListMultipartUploads"
            ],
            "Resource": [
              "arn:aws:s3:::MyStagingS3Bucket",
              "arn:aws:s3:::MyLogsS3Bucket",
              "arn:aws:s3:::MyInputS3Bucket",
              "arn:aws:s3:::MyOutputS3Bucket",
              "arn:aws:s3:::AnotherRequiredS3Buckets"
            ]
        },
```

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

Politique gérée par défaut pour le rôle d'instance EC2

Le contenu de`AmazonEC2RoleforDataPipelineRole` est indiqué ci-dessous. Il s'agit de la politique gérée associée au rôle de ressource par défaut pourAWS Data Pipeline,`DataPipelineDefaultResourceRole`. Lorsque vous définissez un rôle de ressource pour votre pipeline, nous vous recommandons de commencer par cette politique d'autorisations, puis de supprimer les autorisations pour les actions deAWS service qui ne sont pas requises.

La version 3 de la politique est affichée, qui est la version la plus récente au moment de la rédaction de cet article. Consultez la version la plus récente de la stratégie à l'aide de la console IAM.

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

Création de rôles IAM pourAWS Data Pipeline et modification des autorisations de rôle

Utilisez les procédures suivantes pour créer des rôles pourAWS Data Pipeline utiliser la console IAM. Le processus se compose de deux étapes. Tout d'abord, vous devez créer une stratégie d'autorisation à attacher au rôle. Vous devez créer le rôle et d'attacher la stratégie. Après avoir créé

un rôle, vous pouvez modifier les autorisations du rôle en attachant et en détachant des politiques d'autorisation.

> **Note**
>
> Lorsque vous créez des rôles pourAWS Data Pipeline utiliser la console comme décrit ci-dessous, IAM crée et associe les politiques de confiance appropriées requises par le rôle.

Pour créer une politique d'autorisations à utiliser avec un rôle pourAWS Data Pipeline

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, sélectionnez Politiques, puis Créer une politique.
3. Sélectionnez l'onglet JSON.
4. Si vous créez un rôle de pipeline, copiez et collez le contenu de l'exemple de politique dans celui-ciExemple de stratégie d'autorisations de rôle de rôle de rôle, en le modifiant en fonction de vos exigences de sécurité. Sinon, si vous créez un rôle d'instance EC2 personnalisé, procédez de même pour l'exemple présenté dansPolitique gérée par défaut pour le rôle d'instance EC2.
5. Choisissez Review policy (Examiner une politique).
6. Entrez un nom pour la politique, par exemple,`MyDataPipelineRolePolicy` et une description facultative, puis choisissez Créer une politique.
7. Notez le nom de la stratégie. Vous en avez besoin lors de la création de votre rôle.

Pour créer un rôle IAM pour AWS Data Pipeline

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Rôles, puis Créer un rôle.
3. Sous Choisir un cas d'utilisation, choisissez Data Pipeline.
4. Sous Sélectionnez votre cas d'utilisation, effectuez l'une des actions suivantes :
   - ChoisissezData Pipeline de créer un rôle de pipeline.
   - ChoisissezEC2 Role for Data Pipeline de créer un rôle de ressource.
5. Sélectionnez Next: Permissions (Étape suivante : autorisations).
6. Si la politique par défaut pourAWS Data Pipeline est répertoriée, procédez comme suit pour créer le rôle, puis modifiez-le conformément aux instructions de la procédure suivante.

Sinon, entrez le nom de la politique que vous avez créée dans la procédure ci-dessus, puis sélectionnez-la dans la liste.

7. Choisissez Suivant : Tags, entrez les balises à ajouter au rôle, puis choisissez Suivant : Révision.
8. Entrez un nom pour le rôle, par exemple,`MyDataPipelineRole` et une description facultative, puis choisissez Créer un rôle.

Pour associer ou détacher une politique d'autorisations pour un rôle IAM pourAWS Data Pipeline

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Rôles
3. Dans la zone de recherche, commencez à saisir le nom du rôle que vous souhaitez modifier, par exemple, DataPipelineDefaultRoleou MyDataPipelineRole, puis choisissez le nom du rôle dans la liste.
4. Dans l'onglet Autorisations, procédez comme suit :
   - Pour dissocier une politique d'autorisations, sous Politiques d'autorisations, cliquez sur le bouton Supprimer situé à l'extrême droite de l'entrée de la politique. Choisissez Détacher lorsque vous êtes invité à confirmer.
   - Pour joindre une politique que vous avez créée précédemment, choisissez Joindre des politiques. Dans la zone de recherche, commencez à saisir le nom de la politique que vous souhaitez modifier, sélectionnez-la dans la liste, puis choisissez Joindre une politique.

Modification des rôles pour un pipeline existant

Si vous souhaitez attribuer un rôle de pipeline ou un rôle de ressource différent à un pipeline, vous pouvez utiliser l'éditeur d'architecture de laAWS Data Pipeline console.

Pour modifier les rôles attribués à un pipeline à l'aide de la console

1. Ouvrez laAWS Data Pipeline console à l'adresse https://console.aws.amazon.com/datapipeline/.
2. Sélectionnez le pipeline dans la liste, puis choisissez Actions, puis Modifier.
3. Dans le volet droit de l'éditeur d'architecture, choisissez Autres.
4. Dans les listes Role de ressource et Rôles, choisissez les rôlesAWS Data Pipeline que vous souhaitez attribuer, puis cliquez sur Enregistrer.

Journalisation et surveillance dans AWS Data Pipeline

AWS Data Pipelineest intégré àAWS CloudTrail, service qui enregistre les actions effectuées par un utilisateur, un rôle ou unAWS service dansAWS Data Pipeline. CloudTrail capture les appels d'API vers enAWS Data Pipeline tant qu'événements. Les appels capturés incluent des appels de la console AWS Data Pipeline et les appels de code vers les opérations d'API AWS Data Pipeline. Si vous créez un journal d'activité, vous pouvez activer la livraison continue d' CloudTrail événements à un compartiment Amazon S3, y compris des événements pourAWS Data Pipeline. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la CloudTrail console dans Event history (Historique des événements). À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée àAWS Data Pipeline, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur de la demande, la date de la demande, ainsi que d'autres informations.

Pour en savoir plus CloudTrail, consultez le Guide deAWS CloudTrail l'utilisateur.

AWS Data PipelineInformations dans CloudTrail

CloudTrail est activé dans votreAWS compte lors de la création de ce dernier. Lorsqu'une activité a lieu dansAWS Data Pipeline, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements deAWS service dans Event history (Historique des événements). Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour de plus amples informations, veuillez consulter Affichage des événements avec l'historique des CloudTrail événements.

Pour enregistrer en continu les événements dans votre compte AWS, y compris les événements d'AWS Data Pipeline, créez un journal d'activité. Un journal CloudTrail de suivi permet de livrer des fichiers journaux vers un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions AWS. Le journal d'activité consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autresAWS services pour analyser plus en profondeur les données d'événement collectées dans les CloudTrail journaux et agir sur celles-ci. Pour en savoir plus, consultez les ressources suivantes :

- Présentation de la création d'un journal d'activité
- CloudTrail Services et intégrations pris en charge
- Configuration des Notifications de Amazon SNS pour CloudTrail

- Réception de fichiers CloudTrail journaux de plusieurs régions et Réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes lesAWS Data Pipeline actions sont enregistrées CloudTrail et documentées dans le chapitre Actions de référence de l'API AWS Data Pipeline. Par exemple, les appels vers l'CreatePipelineaction génèrent les entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été effectuée avec les informations d'identification de rôle racine ou IAM.
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail .

Présentation des entrées des fichiers journaux AWS Data Pipeline

Un journal d'activité est une configuration qui permet d'envoyer des événements sous forme de fichiers journaux à un compartiment Simple Storage Service (Amazon S3) que vous spécifiez. CloudTrail les fichiers journaux peuvent contenir une ou plusieurs entrées de journal. Un événement représente une demande individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail Les fichiers journaux ne constituent pas une trace de pile ordonnée des appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre l'`CreatePipeline`opération :

```
    {
"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
  ]
}
```

Réponse aux incidents dans AWS Data Pipeline

La réponse aux incidents pour AWS Data Pipeline est de la responsabilité d'AWS. AWS dispose d'une politique et d'un programme officiels et documentés qui régissent la réponse aux incidents.

Les problèmes AWS opérationnels ayant des répercussions majeures sont publiés dans AWS Service Health Dashboard. Les problèmes opérationnels sont également publiés dans les différents comptes via le tableau de bord d'état personnel.

Validation de la conformité pour AWS Data Pipeline

AWS Data Pipeline n'entre pas dans le champ d'application des programmes de conformité AWS. Pour obtenir la liste des services AWS dans le cadre de programmes de conformité spécifiques, consultez Services AWS concernés par le programme de conformité. Pour obtenir des informations générales, consultez la page Programmes de conformité AWS.

Résilience dans AWS Data Pipeline

L'infrastructure mondiale d'AWS repose sur les Régions AWS et les zones de disponibilité AWS. Les Régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour en savoir plus sur AWS les régions et les zones de disponibilité, consultez AWS Infrastructure mondiale.

Sécurité de l'infrastructure dans AWS Data Pipeline

En tant que service géré,AWS Data Pipelineest protégé par leAWSprocédures de sécurité du réseau mondial qui sont décrites dans leAmazon Web Services : Présentation des procédures de sécuritélivre blanc.

Vous utilisez les appels d'API publiés AWS pour accéder à AWS Data Pipeline via le réseau. Les clients doivent supporter le protocole TLS (Sécurité de la couche transport) 1.0 ou une version ultérieure. Nous recommandons TLS 1.2 ou version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Configuration et analyse des vulnérabilités dans AWS Data Pipeline

La configuration et les contrôles informatiques sont une responsabilité partagée entre AWS et vous, notre client. Pour de plus amples informations, veuillez consulter Modèle de responsabilité partagée AWS.

Didacticiels

Les didacticiels suivants vous step-by-step guident tout au long du processus de création et d'utilisation de pipelines avecAWS Data Pipeline.

Didacticiels

- Traitez les données à l'aide d'Amazon EMR avec Hadoop Streaming
- Copier des données CSV entre des compartiments Amazon S3 à l'aide de AWS Data Pipeline
- Exportez des données MySQL vers Amazon S3 à l'aide de AWS Data Pipeline
- Copier des données vers Amazon Redshift à l'aide de AWS Data Pipeline

Traitez les données à l'aide d'Amazon EMR avec Hadoop Streaming

Vous pouvez l'utiliser AWS Data Pipeline pour gérer vos clusters Amazon EMR. AWS Data PipelineVous pouvez ainsi spécifier les conditions préalables qui doivent être remplies avant le lancement du cluster (par exemple, s'assurer que les données du jour ont été chargées sur Amazon S3), un calendrier pour exécuter le cluster de manière répétée et la configuration du cluster à utiliser. Le didacticiel suivant vous guide tout au long du lancement d'un simple cluster.

Dans ce didacticiel, vous allez créer un pipeline pour un cluster Amazon EMR simple afin d'exécuter une tâche Hadoop Streaming préexistante fournie par Amazon EMR et d'envoyer une notification Amazon SNS une fois la tâche terminée avec succès. Vous utilisez la ressource de cluster Amazon EMR fournie par AWS Data Pipeline pour cette tâche. L'exemple d'application est appelé WordCount et peut également être exécuté manuellement à partir de la console Amazon EMR. Notez que les clusters créés par vous AWS Data Pipeline en votre nom sont affichés dans la console Amazon EMR et sont facturés sur votre compte AWS.

Objets de pipeline

Le pipeline utilise les objets suivants :

EmrActivity

Définit le travail à effectuer dans le pipeline (exécuter une tâche Hadoop Streaming préexistante fournie par Amazon EMR).

EmrCluster

Ressource qu'AWS Data Pipeline utilise pour effectuer cette activité.

Un cluster est un ensemble d'instances Amazon EC2. AWS Data Pipelinelance le cluster puis y met fin une fois la tâche terminée.

Planificateur

Date et heure de début, et durée de l'activité. Si vous le souhaitez, vous pouvez indiquer la date et l'heure de fin.

SnsAlarm

Envoie une notification Amazon SNS à la rubrique que vous avez spécifiée une fois la tâche terminée avec succès.

Table des matières

Avant de commencer

Assurez-vous d'avoir complété les étapes suivantes :

- Effectuez les tâches définies dans Configuration pour AWS Data Pipeline.
- (Facultatif) Configurez un VPC pour le cluster et un groupe de sécurité pour le VPC.
- Créez une rubrique pour l'envoi de notification par e-mail et notez la rubrique Amazon Resource Name (ARN). Pour plus d'informations, consultez Création d'une rubrique dans le Guide de mise en route avec Amazon Simple Notification Service.

Lancement d'un cluster à l'aide de la ligne de commande

Si vous exécutez régulièrement un cluster Amazon EMR pour analyser des journaux Web ou analyser des données scientifiques, vous pouvez l'utiliser AWS Data Pipeline pour gérer vos clusters Amazon EMR. AvecAWS Data Pipeline, vous pouvez spécifier les conditions préalables qui doivent être remplies avant le lancement du cluster (par exemple, vous assurer que les données d'aujourd'hui ont été chargées sur Amazon S3.) Ce didacticiel explique comment lancer un cluster qui peut servir de modèle pour un pipeline simple basé sur Amazon EMR ou faire partie d'un pipeline plus complexe.

Prérequis

Avant de pouvoir utiliser l'interface de ligne de commande, vous devez exécuter les tâches suivantes :

1. Installez et configurez une interface de ligne de commande (CLI). Pour plus d'informations, veuillez consulter Accès à AWS Data Pipeline.
2. Assurez-vous que les rôles IAM sont nommés DataPipelineDefaultRoleet DataPipelineDefaultResourceRoleexistent. La AWS Data Pipeline console crée automatiquement ces rôles pour vous. Si vous n'avez pas utilisé la AWS Data Pipeline console au moins une fois, vous devez créer ces rôles manuellement. Pour plus d'informations, veuillez consulter Rôles IAM pour AWS Data Pipeline.

Tâches

- Création du fichier de définition du pipeline
- Chargement et activation de la définition de pipeline
- Surveillance des exécutions du pipeline

Création du fichier de définition du pipeline

Le code suivant est le fichier de définition du pipeline pour un cluster Amazon EMR simple qui exécute une tâche de streaming Hadoop existante fournie par Amazon EMR. Cet exemple d'application s'appelle WordCount et vous pouvez également l'exécuter à l'aide de la console Amazon EMR.

Copiez le code dans un fichier texte et enregistrez-le sous `MyEmrPipelineDefinition.json`. Vous devez remplacer l'emplacement du compartiment Amazon S3 par le nom d'un compartiment Amazon S3 dont vous êtes le propriétaire. Vous devez également remplacer les dates de début et de fin. Pour que votre pipeline se lance immédiatement, définissez le champ `startDateTime` sur une date passée et `endDateTime` sur une date future. AWS Data Pipeline démarre ensuite le lancement des clusters « en retard » pour tenter de régler ce qu'il considère comme des éléments en attente. Ce renvoi signifie que vous n'avez pas à attendre une heure pour qu'AWS Data Pipeline lance son premier cluster.

```
{
  "objects": [
    {
```

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

Le pipeline a trois objets :

- `Hourly`, qui représente la planification du travail. Vous pouvez définir une planification comme l'un des champs d'une activité. Dans ce cas, l'activité s'exécute conformément à cette planification ou, dans le cas présent, toutes les heures.
- `MyCluster`, qui représente l'ensemble des instances Amazon EC2 utilisées pour exécuter le cluster. Vous pouvez spécifier la taille et le nombre d'instances EC2 à exécuter en tant que cluster. Si vous ne spécifiez pas le nombre d'instances, le cluster démarre avec deux instances, un nœud principal et un nœud de tâches. Vous pouvez spécifier un sous-réseau pour y lancer le cluster. Vous pouvez ajouter des configurations supplémentaires au cluster, telles que des actions d'amorçage pour charger des logiciels supplémentaires sur l'AMI fournie par Amazon EMR.

- `MyEmrActivity`, qui représente le calcul à traiter avec le cluster. Amazon EMR prend en charge plusieurs types de clusters, notamment le streaming, la mise en cascade et les clusters Scripted Hive. Le `runsOn` champ fait référence àMyCluster, en l'utilisant comme spécification des fondements du cluster.

Chargement et activation de la définition de pipeline

Vous devez charger la définition de votre pipeline et activer votre pipeline. Dans les exemples de commandes suivants, remplacez *`pipeline_name`* par une étiquette pour votre pipeline et *`pipeline_file par le chemin complet du fichier`* de définition du pipeline. `.json`

AWS CLI

Pour créer la définition de votre pipeline et activer votre pipeline, utilisez la commande create-pipeline suivante. Notez l'ID de votre pipeline, car vous utiliserez cette valeur avec la plupart des commandes CLI.

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

Pour charger la définition de votre pipeline, utilisez la put-pipeline-definitioncommande suivante.

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

Si votre pipeline est validé avec succès, le `validationErrors` champ est vide. Vous devez consulter tous les avertissements.

Pour activer votre pipeline, utilisez la commande activate-pipeline suivante.

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

Vous pouvez vérifier que votre pipeline apparaît dans la liste des pipelines à l'aide de la commande list-pipelines suivante.

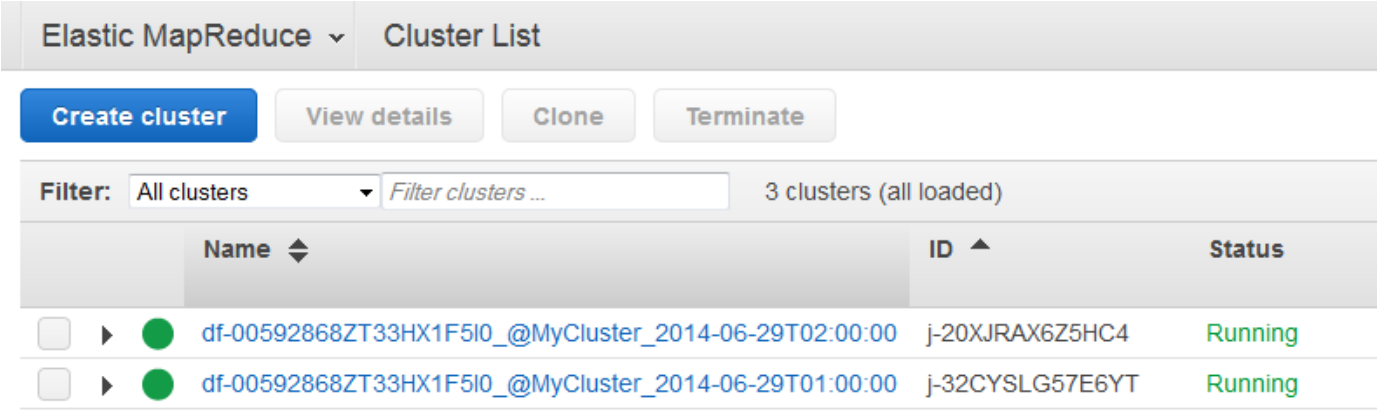
```
aws datapipeline list-pipelines
```

Surveillance des exécutions du pipeline

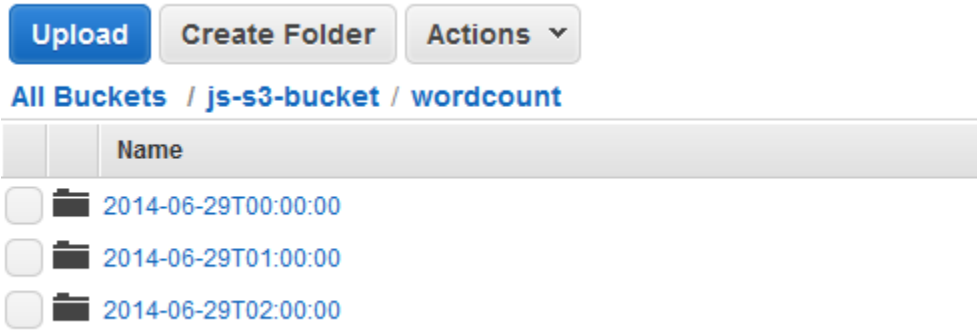
Vous pouvez consulter les clusters lancés à l'AWS Data Pipelineaide de la console Amazon EMR et vous pouvez consulter le dossier de sortie à l'aide de la console Amazon S3.

Pour vérifier la progression des clusters lancés par AWS Data Pipeline

1. Ouvrez la console Amazon EMR.
2. <launch-time>Les clusters qui ont été générés par AWS Data Pipeline ont un nom formaté comme suit : <pipeline-identifier>_@ *< > emr-cluster-name* _.



3. Une fois l'une des exécutions terminée, ouvrez la console Amazon S3 et vérifiez que le dossier de sortie horodaté existe et contient les résultats attendus du cluster.



Copier des données CSV entre des compartiments Amazon S3 à l'aide de AWS Data Pipeline

Une fois que vous avez lu Qu'est-ce que AWS Data Pipeline ? et décidé d'utiliser AWS Data Pipeline pour automatiser le déplacement et la transformation de vos données, il est temps de commencer la

création des pipelines de données. Pour vous aider à comprendre comment fonctionne AWS Data Pipeline, parcourons une tâche simple.

Ce didacticiel explique comment créer un pipeline de données pour copier des données d'un compartiment Amazon S3 vers un autre, puis envoyer une notification Amazon SNS une fois l'activité de copie terminée avec succès. Vous utilisez une instance EC2 gérée par AWS Data Pipeline pour l'activité de copie.

Objets de pipeline

Le pipeline utilise les objets suivants :

CopyActivity

Activité exécutée pour AWS Data Pipeline ce pipeline (copier les données CSV d'un compartiment Amazon S3 vers un autre).

> **Important**
>
> Il existe des limitations lorsque vous utilisez le format de fichier CSV avec `CopyActivity` et `S3DataNode`. Pour plus d'informations, veuillez consulter CopyActivity.

Planificateur

Date et heure de début, et récurrence de l'activité. Si vous le souhaitez, vous pouvez indiquer la date et l'heure de fin.

Ec2Resource

Ressource (instance EC2) qu'AWS Data Pipeline utilise pour exécuter l'activité.

S3 DataNode

Les nœuds d'entrée et de sortie (compartiments Amazon S3) pour ce pipeline.

SnsAlarm

L'action AWS Data Pipeline doit être entreprise lorsque les conditions spécifiées sont remplies (envoyer des notifications Amazon SNS à une rubrique une fois la tâche terminée avec succès).

Table des matières

- Copie des données CSV à l'aide de la ligne de commande

Avant de commencer

Assurez-vous d'avoir complété les étapes suivantes :

- Effectuez les tâches définies dans Configuration pour AWS Data Pipeline.
- (Facultatif) Configurez un VPC pour l'instance et un groupe de sécurité pour le VPC.
- Créez un bucket Amazon S3 en tant que source de données.

  Pour plus d'informations, consultez Création d'un compartiment dans le Guide de l'utilisateur d'Amazon Simple Storage Service.
- Chargez vos données dans votre compartiment Amazon S3.

  Pour plus d'informations, consultez Ajouter un objet à un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.
- Création d'un autre bucket Amazon S3 en tant que cible de données
- Créez une rubrique pour l'envoi de notification par e-mail et notez la rubrique Amazon Resource Name (ARN). Pour plus d'informations, consultez Création d'une rubrique dans le Guide de mise en route avec Amazon Simple Notification Service.
- (Facultatif) Ce didacticiel utilise les stratégies de rôle IAM par défaut créées par AWS Data Pipeline. Si vous préférez créer et configurer votre propre politique de rôle IAM et vos propres relations de confiance, suivez les instructions décrites dansRôles IAM pour AWS Data Pipeline.

Copie des données CSV à l'aide de la ligne de commande

Vous pouvez créer et utiliser des pipelines pour copier des données d'un compartiment Amazon S3 vers un autre.

Prérequis

Avant de commencer, exécutez les étapes suivantes :

1. Installez et configurez une interface de ligne de commande (CLI). Pour plus d'informations, veuillez consulter Accès à AWS Data Pipeline.
2. Assurez-vous que les rôles IAM sont nommés DataPipelineDefaultRoleet DataPipelineDefaultResourceRoleexistent. La AWS Data Pipeline console crée automatiquement

ces rôles pour vous. Si vous n'avez pas utilisé la AWS Data Pipeline console au moins une fois, vous devez créer ces rôles manuellement. Pour plus d'informations, veuillez consulter Rôles IAM pour AWS Data Pipeline.

Tâches

- Définition d'un pipeline au format JSON
- Chargement et activation de la définition de pipeline

Définition d'un pipeline au format JSON

Cet exemple de scénario montre comment utiliser les définitions de pipeline JSON et l'AWS Data Pipelineinterface de ligne de commande pour planifier la copie de données entre deux compartiments Amazon S3 à un intervalle de temps spécifique. Voici le fichier JSON intégral de définition de pipeline, suivi d'une explication de chacune de ses sections.

> **Note**
>
> Nous vous recommandons d'utiliser un éditeur de texte qui peut vous aider à vérifier la syntaxe des fichiers au format JSON et de nommer le fichier avec l'extension .json.

Dans cet exemple, pour plus de clarté, nous ignorons les champs facultatifs et n'affichons que les champs obligatoires. Voici le fichier JSON complet de l'exemple :

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

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

Planificateur

Le pipeline définit une planification avec une date de début et une date de fin, ainsi qu'une période pour déterminer la fréquence à laquelle l'activité du pipeline s'exécute.

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

Nœuds de données Amazon S3

Ensuite, le composant du DataNode pipeline S3 d'entrée définit un emplacement pour les fichiers d'entrée ; dans ce cas, un emplacement de compartiment Amazon S3. Le DataNode composant S3 d'entrée est défini par les champs suivants :

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

Id

Nom défini par l'utilisateur de l'emplacement d'entrée (libellé fourni à titre de référence uniquement).

Type

Le type de composant du pipeline, qui est « S3 DataNode » pour correspondre à l'emplacement où se trouvent les données, dans un compartiment Amazon S3.

Planificateur

Une référence au composant de planification que nous avons créé dans les lignes précédentes du fichier JSON intitulé « MySchedule ».

Chemin

Chemin d'accès aux données associées au nœud de données. La syntaxe d'un nœud de données est déterminée par son type. Par exemple, la syntaxe d'un chemin Amazon S3 suit une syntaxe différente qui convient à une table de base de données.

Ensuite, le DataNode composant S3 de sortie définit l'emplacement de destination de sortie pour les données. Il suit le même format que le DataNode composant S3 d'entrée, à l'exception du nom du composant et d'un chemin différent pour indiquer le fichier cible.

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

Ressource

Il s'agit d'une définition de la ressource de calcul qui exécute l'opération de copie. Dans cet exemple, AWS Data Pipeline doit automatiquement créer une instance EC2 pour effectuer la tâche de copie et mettre fin à la ressource une fois la tâche terminée. Les champs définis ici contrôlent la création et le fonctionnement de l'instance EC2 qui effectue le travail. Le composant EC2Resource est défini par les champs suivants :

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

Id

Nom défini par l'utilisateur pour la planification du pipeline (libellé fourni à titre de référence uniquement).

Type

Type de ressource de calcul pour effectuer le travail ; dans ce cas, une instance EC2. D'autres types de ressources sont disponibles, par exemple un EmrCluster type.

Planificateur

Planification sur laquelle créer la ressource de calcul.

instanceType

Taille de l'instance EC2 à créer. Assurez-vous que vous définissez pour l'instance EC2 la taille appropriée qui correspond le mieux à la charge du travail que vous souhaitez effectuer avec AWS Data Pipeline. Dans ce cas, nous avons défini une instance EC2 m1.medium. Pour plus d'informations sur les différents types d'instances et sur le moment de les utiliser, consultez la rubrique Types d'instances Amazon EC2 à l'adresse http://aws.amazon.com/ec2/instance-types/.

Rôle

Le rôle IAM du compte qui accède aux ressources, par exemple en accédant à un bucket Amazon S3 pour récupérer des données.

resourceRole

Rôle IAM du compte qui crée des ressources, comme la création et la configuration d'une instance EC2 en votre nom. Le rôle et le rôle ResourceRole peuvent être identiques, mais ils fournissent séparément une plus grande granularité dans votre configuration de sécurité.

Activité

La dernière section du fichier JSON correspond à la définition de l'activité représentant le travail à effectuer. Cet exemple permet `CopyActivity` de copier les données d'un fichier CSV d'un bucket http://aws.amazon.com/ec2/instance-types/ vers un autre. Le composant `CopyActivity` est défini par les champs suivants :

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

```
    "ref": "MySchedule"
  }
}
```

Id

Nom défini par l'utilisateur pour l'activité (libellé fourni à titre de référence uniquement).

Type

Le type d'activité à effectuer, tel queMyCopyActivity.

runsOn

Ressource de calcul qui effectue le travail que cette activité définit. Dans cet exemple, nous fournissons une référence à l'instance EC2 définie précédemment. L'utilisation du champ `runsOn` entraîne la création automatique de l'instance EC2 par AWS Data Pipeline. Le champ `runsOn` indique que la ressource existe dans l'infrastructure AWS, tandis que la valeur `workerGroup` signifie que vous voulez utiliser vos propres ressources locales pour effectuer le travail.

Entrée

Emplacement des données à copier.

Sortie

Emplacement cible des données.

Planificateur

Planification d'exécution de cette activité.

Chargement et activation de la définition de pipeline

Vous devez charger la définition de votre pipeline et activer votre pipeline. Dans les exemples de commandes suivants, remplacez *`pipeline_name`* par une étiquette pour votre pipeline et *`pipeline_file par le chemin complet du fichier`* de définition du pipeline. `.json`

AWS CLI

Pour créer votre définition de pipeline et activer votre pipeline, utilisez la commande create-pipeline suivante. Notez l'ID de votre pipeline, car vous utiliserez cette valeur avec la plupart des commandes CLI.

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

Pour charger la définition de votre pipeline, utilisez la put-pipeline-definitioncommande suivante.

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

Si votre pipeline est validé avec succès, le `validationErrors` champ est vide. Vous devez consulter tous les avertissements.

Pour activer votre pipeline, utilisez la commande activate-pipeline suivante.

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

Vous pouvez vérifier que votre pipeline apparaît dans la liste des pipelines à l'aide de la commande list-pipelines suivante.

```
aws datapipeline list-pipelines
```

Exportez des données MySQL vers Amazon S3 à l'aide de AWS Data Pipeline

Ce didacticiel explique comment créer un pipeline de données pour copier des données (lignes) d'une table de la base de données MySQL vers un fichier CSV (valeurs séparées par des virgules) dans un compartiment Amazon S3, puis envoyer une notification Amazon SNS une fois l'activité de copie terminée avec succès. Vous allez utiliser une instance EC2 fournie par AWS Data Pipeline pour cette activité de copie.

Objets de pipeline

Le pipeline utilise les objets suivants :

- CopyActivity
- Ec2Resource

- MySqlDataNode
- S3 DataNode
- SnsAlarm

Table des matières

Avant de commencer

Assurez-vous d'avoir complété les étapes suivantes :

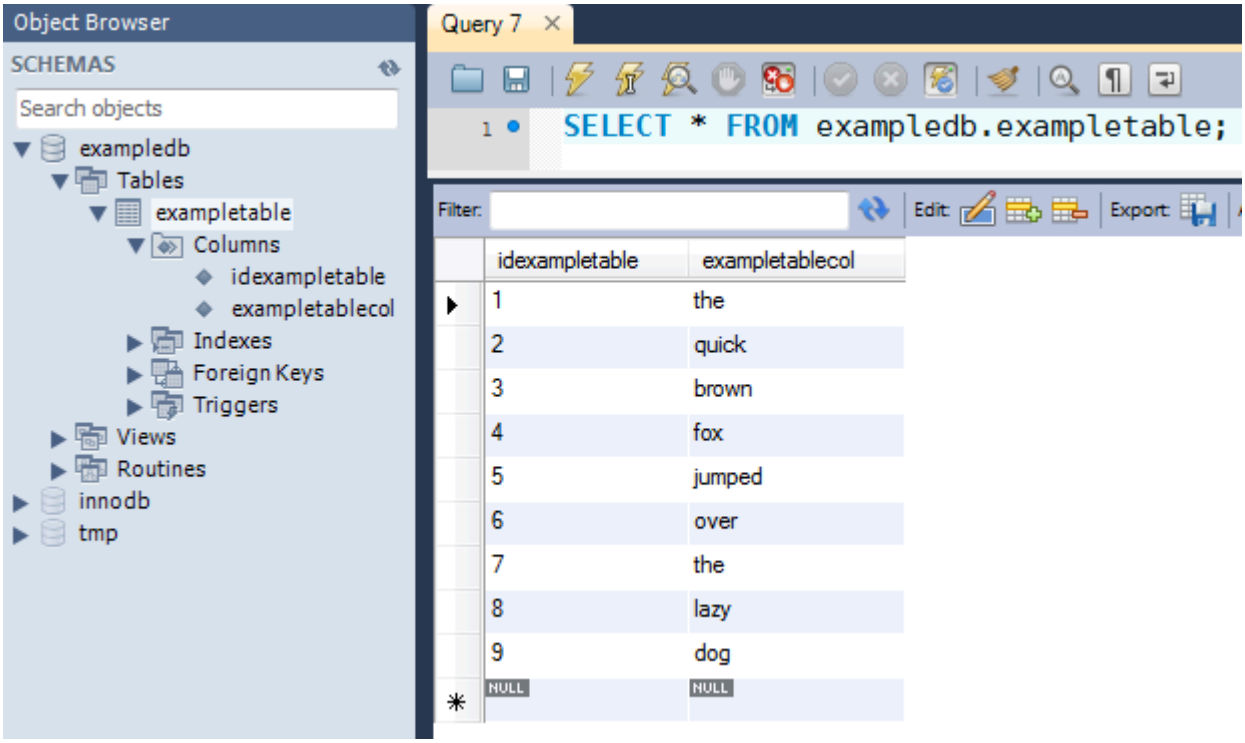
- Effectuez les tâches définies dans Configuration pour AWS Data Pipeline.
- (Facultatif) Configurez un VPC pour l'instance et un groupe de sécurité pour le VPC.
- Créez un bucket Amazon S3 en tant que sortie de données.

  Pour plus d'informations, consultez le guide de l'utilisateur de Create a bucket in Amazon Simple Storage Service.
- Créez et lancez une instance de base de données MySQL en tant que source de données.

  Pour plus d'informations, consultez Lancer une instance de base de données dans le guide de démarrage Amazon RDS. Après avoir créé une instance Amazon RDS, consultez la section Créer une table dans la documentation MySQL.

  > **Note**
  >
  > Notez le nom d'utilisateur et le mot de passe que vous avez utilisés pour créer l'instance MySQL. Après avoir lancé votre instance de base de données MySQL, notez le point de terminaison de l'instance. Vous aurez besoin de ces informations ultérieurement.
- Connectez-vous à votre instance de base de données MySQL, créez une table, puis ajoutez des valeurs de données de test à cette nouvelle table.

  À titre d'illustration, nous avons créé ce didacticiel en utilisant une table MySQL avec la configuration et les exemples de données suivants. La capture d'écran suivante provient de MySQL Workbench 5.2 CE :



Pour de plus amples informations, veuillez consulter Création d'une table dans la documentation MySQL et la page produit MySQL Workbench.

- Créez une rubrique pour l'envoi de notification par e-mail et notez la rubrique Amazon Resource Name (ARN). Pour plus d'informations, consultez la section Création d'une rubrique dans le Guide de démarrage d'Amazon Simple Notification Service.
- (Facultatif) Ce didacticiel utilise les stratégies de rôle IAM par défaut créées par AWS Data Pipeline. Si vous préférez créer et configurer votre politique de rôle IAM et vos relations de confiance, suivez les instructions décrites dansRôles IAM pour AWS Data Pipeline.

Copie de données MySQL à l'aide de la ligne de commande

Vous pouvez créer un pipeline pour copier les données d'une table MySQL vers un fichier dans un compartiment Amazon S3.

Prérequis

Avant de commencer, exécutez les étapes suivantes :

1. Installez et configurez une interface de ligne de commande (CLI). Pour plus d'informations, veuillez consulter Accès à AWS Data Pipeline.

2. Assurez-vous que les rôles IAM sont nommés DataPipelineDefaultRoleet DataPipelineDefaultResourceRoleexistent. La AWS Data Pipeline console crée automatiquement ces rôles pour vous. Si vous n'avez pas utilisé la AWS Data Pipeline console au moins une fois, vous devez créer ces rôles manuellement. Pour plus d'informations, veuillez consulter Rôles IAM pour AWS Data Pipeline.
3. Configurez un compartiment Amazon S3 et une instance Amazon RDS. Pour plus d'informations, veuillez consulter Avant de commencer.

Tâches

- Définition d'un pipeline au format JSON
- Chargement et activation de la définition de pipeline

Définition d'un pipeline au format JSON

Cet exemple de scénario montre comment utiliser les définitions de pipeline JSON et l'AWS Data Pipelineinterface de ligne de commande pour copier les données (lignes) d'une table d'une base de données MySQL vers un fichier CSV (valeurs séparées par des virgules) dans un compartiment Amazon S3 à un intervalle de temps spécifié.

Voici le fichier JSON intégral de définition de pipeline, suivi d'une explication de chacune de ses sections.

> **Note**
>
> Nous vous recommandons d'utiliser un éditeur de texte qui peut vous aider à vérifier la syntaxe des fichiers au format JSON et de nommer le fichier avec l'extension .json.

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

```
      "id": "CopyActivityId112",
      "input": {
        "ref": "MySqlDataNodeId115"
      },
      "schedule": {
        "ref": "ScheduleId113"
      },
      "name": "My Copy",
      "runsOn": {
        "ref": "Ec2ResourceId116"
      },
      "onSuccess": {
        "ref": "ActionId1"
      },
      "onFail": {
        "ref": "SnsAlarmId117"
      },
      "output": {
        "ref": "S3DataNodeId114"
      },
      "type": "CopyActivity"
    },
    {
      "id": "S3DataNodeId114",
      "schedule": {
        "ref": "ScheduleId113"
      },
      "filePath": "s3://example-bucket/rds-output/output.csv",
      "name": "My S3 Data",
      "type": "S3DataNode"
    },
    {
      "id": "MySqlDataNodeId115",
      "username": "my-username",
      "schedule": {
        "ref": "ScheduleId113"
      },
      "name": "My RDS Data",
      "*password": "my-password",
      "table": "table-name",
      "connectionString": "jdbc:mysql://your-sql-instance-name.id.region-
name.rds.amazonaws.com:3306/database-name",
      "selectQuery": "select * from #{table}",
      "type": "SqlDataNode"
```

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

Nœud de données MySQL

Le composant du MySqlDataNode pipeline d'entrée définit un emplacement pour les données d'entrée ; dans ce cas, il s'agit d'une instance Amazon RDS. Le MySqlDataNode composant d'entrée est défini par les champs suivants :

```
{
  "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

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

Nom d'utilisateur

Nom d'utilisateur du compte de base de données disposant des autorisations suffisantes pour extraire des données de la table de base de données. Remplacez *my-username* par le nom de votre utilisateur.

Planificateur

Référence au composant de planification que nous avons créé dans les lignes précédentes du fichier JSON.

Nom

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

*Password

Mot de passe du compte de base de données précédé du préfixe astérisque pour indiquer qu'AWS Data Pipeline doit chiffrer la valeur de mot de passe. Remplacez *my-password* par

le mot de passe correspondant à votre utilisateur. Le champ de mot de passe est précédé du caractère spécial astérisque. Pour plus d'informations, veuillez consulter Caractères spéciaux.

Tableau

Nom de la table de base de données qui contient les données à copier. Remplacez *table-name* par le nom de votre table de base de données.

connectionChaîne

Chaîne de connexion JDBC permettant à l'CopyActivityobjet de se connecter à la base de données.

selectQuery

Requête SQL SELECT valide qui spécifie les données à copier à partir de la table de base de données. Notez que `#{table}` est une expression qui réutilise le nom de table fourni par la variable « table » dans les lignes précédentes du fichier JSON.

Type

Le SqlDataNode type, qui est une instance Amazon RDS utilisant MySQL dans cet exemple.

> **Note**
>
> Le type MySqlDataNode est obsolète. Bien que vous puissiez encore l'utiliserMySqlDataNode, nous vous recommandons d'utiliserSqlDataNode.

Nœud de données Amazon S3

Ensuite, le composant du pipeline S3Output définit un emplacement pour le fichier de sortie ; dans ce cas, un fichier CSV dans un emplacement de compartiment Amazon S3. Le DataNode composant S3 de sortie est défini par les champs suivants :

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

```
},
```

Id

ID défini par l'utilisateur (libellé fourni à titre de référence uniquement).

Planificateur

Référence au composant de planification que nous avons créé dans les lignes précédentes du fichier JSON.

filePath

Chemin d'accès aux données associées au nœud de données, qui est un fichier de sortie CSV dans cet exemple.

Nom

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

Type

Le type d'objet du pipeline, S3 DataNode pour correspondre à l'emplacement où se trouvent les données, dans un compartiment Amazon S3.

Ressource

Il s'agit d'une définition de la ressource de calcul qui exécute l'opération de copie. Dans cet exemple, AWS Data Pipeline doit automatiquement créer une instance EC2 pour effectuer la tâche de copie et mettre fin à la ressource une fois la tâche terminée. Les champs définis ici contrôlent la création et le fonctionnement de l'instance EC2 qui effectue le travail. Le composant EC2Resource est défini par les champs suivants :

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

Id

ID défini par l'utilisateur (libellé fourni à titre de référence uniquement).

Planificateur

Planification sur laquelle créer la ressource de calcul.

Nom

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

Rôle

Le rôle IAM du compte qui accède aux ressources, par exemple en accédant à un bucket Amazon S3 pour récupérer des données.

Type

Type de ressource de calcul pour effectuer le travail ; dans ce cas, une instance EC2. D'autres types de ressources sont disponibles, par exemple un EmrCluster type.

resourceRole

Rôle IAM du compte qui crée des ressources, comme la création et la configuration d'une instance EC2 en votre nom. Le rôle et le rôle ResourceRole peuvent être identiques, mais ils fournissent séparément une plus grande granularité dans votre configuration de sécurité.

Activité

La dernière section du fichier JSON correspond à la définition de l'activité représentant le travail à effectuer. Dans ce cas, nous utilisons un CopyActivity composant pour copier les données d'un fichier d'un compartiment Amazon S3 vers un autre fichier. Le composant CopyActivity est défini par les champs suivants :

```
{
  "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 défini par l'utilisateur (libellé fourni à titre de référence uniquement)

Entrée

Emplacement des données MySQL à copier

Planificateur

Planification d'exécution de cette activité

Nom

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement)

runsOn

Ressource de calcul qui effectue le travail que cette activité définit. Dans cet exemple, nous fournissons une référence à l'instance EC2 définie précédemment. L'utilisation du champ `runsOn` entraîne la création automatique de l'instance EC2 par AWS Data Pipeline. Le champ `runsOn` indique que la ressource existe dans l'infrastructure AWS, tandis que la valeur workerGroup signifie que vous voulez utiliser vos propres ressources locales pour effectuer le travail.

onSuccess

Notification SnsAlarm à envoyer si l'activité se termine correctement

onFail

Notification SnsAlarm à envoyer si l'activité échoue

Sortie

L'emplacement Amazon S3 du fichier de sortie CSV

Type

Type d'activité à effectuer.

Chargement et activation de la définition de pipeline

Vous devez charger la définition de votre pipeline et activer votre pipeline. Dans les exemples de commandes suivants, remplacez *pipeline_name* par une étiquette pour votre pipeline et *pipeline_file par le chemin complet du fichier* de définition du pipeline. `.json`

AWS CLI

Pour créer votre définition de pipeline et activer votre pipeline, utilisez la commande create-pipeline suivante. Notez l'ID de votre pipeline, car vous utiliserez cette valeur avec la plupart des commandes CLI.

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

Pour charger la définition de votre pipeline, utilisez la put-pipeline-definitioncommande suivante.

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

Si votre pipeline est validé avec succès, le `validationErrors` champ est vide. Vous devez consulter tous les avertissements.

Pour activer votre pipeline, utilisez la commande activate-pipeline suivante.

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

Vous pouvez vérifier que votre pipeline apparaît dans la liste des pipelines à l'aide de la commande list-pipelines suivante.

```
aws datapipeline list-pipelines
```

Copier des données vers Amazon Redshift à l'aide de AWS Data Pipeline

Ce didacticiel explique comment créer un pipeline qui déplace régulièrement des données d'Amazon S3 vers Amazon Redshift à l'aide du modèle Copy to Redshift de la AWS Data Pipeline console ou d'un fichier de définition de pipeline avec l'AWS Data Pipelineinterface de ligne de commande.

Amazon S3 est un service Web qui vous permet de stocker des données dans le cloud. Pour en savoir plus, consultez Guide de l'utilisateur Amazon Simple Storage Service.

Amazon Redshift est un service d'entrepôt de données dans le cloud. Pour plus d'informations, consultez le guide de gestion Amazon Redshift.

Le didacticiel nécessite plusieurs conditions préalables. Après avoir terminé les étapes suivantes, vous pouvez poursuivre le didacticiel à l'aide de la console ou de l'interface de ligne de commande.

Table des matières

Avant de commencer : configurer les options COPY et charger des données

Avant de copier des données dans Amazon RedshiftAWS Data Pipeline, assurez-vous de :

- Chargez des données depuis Amazon S3.
- Configurez l'`COPY`activité dans Amazon Redshift.

Une fois que ces options fonctionnent correctement et que vous avez réussi à charger des données, transférez ces options à AWS Data Pipeline pour y effectuer la copie.

Pour connaître `COPY` les options, consultez la section COPY du manuel Amazon Redshift Database Developer Guide.

Pour savoir comment charger des données depuis Amazon S3, consultez la section Chargement de données depuis Amazon S3 dans le manuel Amazon Redshift Database Developer Guide.

Par exemple, la commande SQL suivante dans Amazon Redshift crée une nouvelle table nommée LISTING et copie des exemples de données à partir d'un compartiment accessible au public dans Amazon S3.

Remplacez le <iam-role-arn> et la région par vos propres valeurs.

Pour en savoir plus sur cet exemple, consultez Charger des exemples de données depuis Amazon S3 dans le guide de démarrage 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';
```

Configuration du pipeline, création d'un groupe de sécurité et création d'un cluster Amazon Redshift

Pour se préparer pour le didacticiel

1. Effectuez les tâches définies dans Configuration pour AWS Data Pipeline.
2. Créez un groupe de sécurité.
    a. Ouvrez la console Amazon EC2.
    b. Dans le volet de navigation, cliquez sur Security Groups.
    c. Cliquez sur Create Security Group.
    d. Attribuez un nom et une description au groupe de sécurité.
    e. [EC2-Classic] Sélectionnez No VPC pour VPC.
    f. [EC2-VPC] Sélectionnez l'ID de votre VPC pour VPC.
    g. Cliquez sur Create.

3. [EC2-Classic] Créez un groupe de sécurité de cluster Amazon Redshift et spécifiez le groupe de sécurité Amazon EC2.

   a. Ouvrez la console Amazon Redshift.

   b. Dans le volet de navigation, cliquez sur Security Groups.

   c. Cliquez sur Create Cluster Security Group.

   d. Dans la boîte de dialogue Create Cluster Security Group, indiquez un nom et une description pour le groupe de sécurité du cluster.

   e. Cliquez sur le nom du nouveau groupe de sécurité de cluster.

   f. Cliquez sur Add Connection Type.

   g. Dans la boîte de dialogue Add Connection Type, sélectionnez Groupe de sécurité EC2 à partir de Type de connexion, sélectionnez le groupe de sécurité que vous avez créé dans Nom du groupe de sécurité EC2, puis cliquez sur Authorize.

4. [EC2-VPC] Créez un groupe de sécurité de cluster Amazon Redshift et spécifiez le groupe de sécurité VPC.

   a. Ouvrez la console Amazon EC2.

   b. Dans le volet de navigation, cliquez sur Security Groups.

   c. Cliquez sur Create Security Group.

   d. Dans la boîte de dialogue Créer un groupe de sécurité, spécifiez le nom et la description du groupe de sécurité, puis sélectionnez l'ID de votre VPC pour VPC.

   e. Cliquez sur Add Rule. Spécifiez le type de protocole et la plage de ports, puis commencez à taper l'ID du groupe de sécurité dans Source. Sélectionnez le groupe de sécurité que vous avez créé à la deuxième étape.

   f. Cliquez sur Create.

5. Les étapes suivantes résument la procédure à suivre.

   Si vous possédez déjà un cluster Amazon Redshift, notez l'ID du cluster.

   Pour créer un nouveau cluster et charger des exemples de données, suivez les étapes décrites dans la section Premiers pas avec Amazon Redshift. Pour plus d'informations sur la création de clusters, consultez la section Création d'un cluster dans le Guide de gestion Amazon Redshift.

   a. Ouvrez la console Amazon Redshift.

   b. Cliquez sur Launch Cluster.

c. Fournissez les détails obligatoires de votre cluster, puis cliquez sur Continuer.

d. Fournissez le nœud de configuration, puis cliquez sur Continuer.

e. Sur la page des informations de configuration supplémentaires, sélectionnez le groupe de sécurité de cluster que vous avez créé, puis cliquez sur Continuer.

f. Vérifiez les spécifications de votre cluster, puis cliquez sur Launch Cluster.

Copier des données vers Amazon Redshift à l'aide de la ligne de commande

Ce didacticiel explique comment copier des données depuis Amazon S3 vers Amazon Redshift. Vous allez créer une nouvelle table dans Amazon Redshift, puis vous l'utiliserez AWS Data Pipeline pour transférer des données vers cette table à partir d'un compartiment Amazon S3 public, qui contient des exemples de données d'entrée au format CSV. Les journaux sont enregistrés dans un compartiment Amazon S3 dont vous êtes le propriétaire.

Amazon S3 est un service Web qui vous permet de stocker des données dans le cloud. Pour en savoir plus, consultez Guide de l'utilisateur Amazon Simple Storage Service. Amazon Redshift est un service d'entrepôt de données dans le cloud. Pour plus d'informations, consultez le guide de gestion Amazon Redshift.

Prérequis

Avant de commencer, exécutez les étapes suivantes :

1. Installez et configurez une interface de ligne de commande (CLI). Pour plus d'informations, veuillez consulter Accès à AWS Data Pipeline.
2. Assurez-vous que les rôles IAM sont nommés DataPipelineDefaultRoleet DataPipelineDefaultResourceRoleexistent. La AWS Data Pipeline console crée automatiquement ces rôles pour vous. Si vous n'avez pas utilisé la AWS Data Pipeline console au moins une fois, vous devez créer ces rôles manuellement. Pour plus d'informations, veuillez consulter Rôles IAM pour AWS Data Pipeline.
3. Configurez la `COPY` commande dans Amazon Redshift, car ces mêmes options devront fonctionner lorsque vous effectuerez la copie dans AWS Data Pipeline Amazon Redshift. Pour plus d'informations, consultez Avant de commencer : configurer les options COPY et charger des données.

4. Configurez une base de données Amazon Redshift. Pour plus d'informations, veuillez consulter Configuration du pipeline, création d'un groupe de sécurité et création d'un cluster Amazon Redshift.

Tâches

- Définition d'un pipeline au format JSON
- Chargement et activation de la définition de pipeline

Définition d'un pipeline au format JSON

Cet exemple de scénario montre comment copier des données depuis un compartiment Amazon S3 vers Amazon Redshift.

Voici le fichier JSON intégral de définition de pipeline, suivi d'une explication de chacune de ses sections. Nous vous recommandons d'utiliser un éditeur de texte qui peut vous aider à vérifier la syntaxe des fichiers au format JSON et de nommer le fichier avec l'extension `.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"
      }
    }
  ]
}
```

Pour plus d'informations sur ces objets, consultez la documentation suivante.

Objets

- Nœuds de données
- Ressource
- Activité

Nœuds de données

L'exemple utilise un nœud de données d'entrée, un nœud de données de sortie et une base de données.

Nœud de données d'entrée

Le composant du `S3DataNode` pipeline d'entrée définit l'emplacement des données d'entrée dans Amazon S3 et le format des données d'entrée. Pour plus d'informations, veuillez consulter S3 DataNode.

Le composant d'entrée est défini par les champs suivants :

```
{
  "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 défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`schedule`

Référence au composant planification.

`filePath`

Chemin d'accès aux données associées au nœud de données (fichier d'entrée CSV dans l'exemple).

`name`

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`dataFormat`

Référence au format des données de l'activité à traiter.

Nœud de données de sortie

Le composant du `RedshiftDataNode` pipeline de sortie définit un emplacement pour les données de sortie ; dans ce cas, une table dans une base de données Amazon Redshift. Pour

plus d'informations, veuillez consulter RedshiftDataNode. Le composant de sortie est défini par les champs suivants :

```
{
  "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 défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`schedule`

Référence au composant planification.

`tableName`

Nom de la table Amazon Redshift.

`name`

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`createTableSql`

Expression SQL permettant de créer la table dans la base de données.

`database`

Une référence à la base de données Amazon Redshift.

Database (Base de données)

Le composant `RedshiftDatabase` est défini par les champs ci-après. Pour plus d'informations, veuillez consulter RedshiftDatabase.

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

`id`

ID défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`databaseName`

Nom de la base de données logique.

`username`

Nom d'utilisateur pour la connexion à la base de données.

`name`

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`password`

Mot de passe pour la connexion à la base de données.

`clusterId`

ID du cluster Redshift.

Ressource

Il s'agit d'une définition de la ressource de calcul qui exécute l'opération de copie. Dans cet exemple, AWS Data Pipeline doit automatiquement créer une instance EC2 pour effectuer la tâche de copie et mettre fin à l'instance une fois la tâche terminée. Les champs définis ici contrôlent la création et le fonctionnement de l'instance qui effectue le travail. Pour plus d'informations, veuillez consulter Ec2Resource.

Le composant `Ec2Resource` est défini par les champs suivants :

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

`id`

ID défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`schedule`

Planification sur laquelle créer la ressource de calcul.

`securityGroups`

Groupe de sécurité à utiliser pour les instances du pool de ressources.

`name`

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`role`

Le rôle IAM du compte qui accède aux ressources, par exemple en accédant à un bucket Amazon S3 pour récupérer des données.

`logUri`

Le chemin de destination Amazon S3 pour sauvegarder les journaux de Task Runner depuis le`Ec2Resource`.

`resourceRole`

Rôle IAM du compte qui crée des ressources, comme la création et la configuration d'une instance EC2 en votre nom. Le rôle et le rôle ResourceRole peuvent être identiques, mais ils fournissent séparément une plus grande granularité dans votre configuration de sécurité.

Activité

La dernière section du fichier JSON correspond à la définition de l'activité représentant le travail à effectuer. Dans ce cas, nous utilisons un `RedshiftCopyActivity` composant pour copier les données d'Amazon S3 vers Amazon Redshift. Pour plus d'informations, veuillez consulter RedshiftCopyActivity.

Le composant `RedshiftCopyActivity` est défini par les champs suivants :

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

`id`

ID défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`input`

Une référence au fichier source Amazon S3.

`schedule`

Planification d'exécution de cette activité.

`insertMode`

Type d'insertion (`KEEP_EXISTING`, `OVERWRITE_EXISTING` ou `TRUNCATE`).

`name`

Nom défini par l'utilisateur (libellé fourni à titre de référence uniquement).

`runsOn`

Ressource de calcul qui effectue le travail que cette activité définit.

`output`

Une référence à la table de destination Amazon Redshift.

Chargement et activation de la définition de pipeline

Vous devez charger la définition de votre pipeline et activer votre pipeline. Dans les exemples de commandes suivants, remplacez *`pipeline_name`* par une étiquette pour votre pipeline et *`pipeline_file par le chemin complet du fichier`* de définition du pipeline. `.json`

AWS CLI

Pour créer la définition de votre pipeline et activer votre pipeline, utilisez la commande create-pipeline suivante. Notez l'ID de votre pipeline, car vous utiliserez cette valeur avec la plupart des commandes CLI.

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

Pour charger la définition de votre pipeline, utilisez la put-pipeline-definitioncommande suivante.

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

Si votre pipeline est validé avec succès, le `validationErrors` champ est vide. Vous devez consulter tous les avertissements.

Pour activer votre pipeline, utilisez la commande activate-pipeline suivante.

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

Vous pouvez vérifier que votre pipeline apparaît dans la liste des pipelines à l'aide de la commande list-pipelines suivante.

```
aws datapipeline list-pipelines
```

Expressions et fonctions de pipeline

Cette section explique la syntaxe d'utilisation des expressions et des fonctions dans les pipelines, y compris les types de données associés.

Types de données simples

Les types de données suivants peuvent être définis comme valeurs de champ.

Types

- DateTime
- Numérique
- Références d'objet
- Period
- Chaîne

DateTime

AWS Data Pipeline prend en charge la date et l'heure exprimées au format « AAAA-MM-JJTHH:MM:SS » (UTC/GMT uniquement). L'exemple suivant définit le champ `startDateTime` d'un objet `Schedule` avec la valeur `1/15/2012, 11:59 p.m.`, dans le fuseau horaire UTC/GMT.

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

Numérique

AWS Data Pipeline prend en charge les nombres entiers et les valeurs à virgule flottante.

Références d'objet

Objet dans la définition du pipeline. Il peut s'agir de l'objet actuel, du nom d'un objet défini ailleurs dans le pipeline ou d'un objet qui répertorie l'objet actuel dans un champ, référencé par le mot clé `node`. Pour plus d'informations sur `node`, veuillez consulter Référencement des champs et des objets. Pour plus d'informations sur les types d'objet des pipelines, consultez Référence d'objet de pipeline.

Period

Indique la fréquence à laquelle un événement planifié doit s'exécuter. Elle est exprimée au format « N [`years`|`months`|`weeks`|`days`|`hours`|`minutes`] », où N est une valeur entière positive.

La période minimale est de 15 minutes et la durée maximale de 3 ans.

L'exemple suivant définit le champ `period` de l'objet `Schedule` sur 3 heures. Cette action crée une planification qui s'exécute toutes les trois heures.

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

Chaîne

Valeurs de chaîne standard. Les chaînes doivent être entourées de guillemets ("). Vous pouvez utiliser la barre oblique inverse (\) pour introduire une séquence d'échappement devant les caractères d'une chaîne. Les chaînes multilignes ne sont pas prises en charge.

Les exemples suivants montrent des exemples de valeurs de chaîne valides pour le champ `id`.

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

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

Les chaînes peuvent également contenir des expressions qui correspondent à des valeurs de chaîne. Celles-ci sont insérées dans la chaîne et délimitées comme suit : « #{ » et « } ». L'exemple suivant utilise une expression pour insérer le nom de l'objet courant dans un chemin.

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

Pour plus d'informations sur l'utilisation des expressions, consultez Référencement des champs et des objets et Evaluation d'expression.

Expressions

Les expressions vous permettent de partager une valeur entre objets associés. Les expressions sont traitées par le service web AWS Data Pipeline lors de l'exécution, en s'assurant que toutes les expressions sont remplacées par la valeur de l'expression.

Les expressions sont délimitées par : « #{ » et « } ». Vous pouvez utiliser une expression dans n'importe quel objet de définition de pipeline où une chaîne est légale. Si un emplacement est une référence ou est de type ID, NAME, TYPE, SPHERE, sa valeur n'est pas évaluée et il est utilisé tel quel.

L'expression suivante appelle l'une des fonctions AWS Data Pipeline. Pour plus d'informations, consultez Evaluation d'expression.

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

Référencement des champs et des objets

Les expressions peuvent utiliser les champs de l'objet actuel où l'expression existe, ou les champs d'un autre objet qui est lié par une référence.

Le format d'emplacement se compose de l'heure de création suivie par l'heure de création d'objet, par exemple : `@S3BackupLocation_2018-01-31T11:05:33`.

Vous pouvez également référencer l'ID d'emplacement exact spécifié dans la définition de pipeline, comme l'ID de l'emplacement de sauvegarde Amazon S3. Pour référencer l'ID d'emplacement, utilisez `#{parent.@id}`.

Dans l'exemple suivant, le champ `filePath` fait référence au champ `id` du même objet pour former un nom de fichier. La valeur de `filePath` correspond à « `s3://mybucket/ExampleDataNode.csv` ».

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

Pour utiliser un champ qui existe sur un autre objet lié par une référence, utilisez le mot clé `node`. Ce mot clé n'est disponible qu'avec les objets d'alarme (alarm) et de condition préalable (precondition).

Dans l'exemple précédent, une expression d'un objet `SnsAlarm` peut faire référence à la plage de dates et à la plage d'heures d'un objet `Schedule`, car `S3DataNode` fait référence aux deux.

En particulier, le champ `message` d'un `FailureNotify` peut utiliser les champs liés à l'exécution `@scheduledStartTime` et `@scheduledEndTime` d'`ExampleSchedule`, car le champ `onFail` d'`ExampleDataNode` fait référence à `FailureNotify` et que son champ `schedule` fait référence à `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**
>
> Vous pouvez créer des pipelines ayant des dépendances, telles que les tâches de votre pipeline qui dépendent du travail d'autres systèmes ou tâches. Si votre pipeline nécessite certaines ressources, ajoutez ces dépendances au pipeline à l'aide de conditions préalables que vous associez à des nœuds de données et à des tâches. Cette étape rend vos pipelines plus faciles à déboguer et plus résistants. De plus, conservez vos dépendances au sein d'un seul pipeline chaque fois que possible, car la résolution des problèmes de pipeline est difficile.

Expressions imbriquées

AWS Data Pipeline vous permet d'imbriquer des valeurs pour créer des expressions plus complexes. Par exemple, pour effectuer un calcul de temps (soustraire 30 minutes de `scheduledStartTime`) et mettre en forme le résultat à utiliser dans une définition de pipeline, vous pouvez utiliser l'expression suivante dans une activité :

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

et à l'aide du préfixe node si l'expression fait une partie d'un SnsAlarm ou d'un Precondition :

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

Listes

Les expressions peuvent être évaluées sur les listes et les fonctions sur les listes. Par exemple, supposons que la liste soit définie comme suit : `"myList":["one","two"]`. Si cette liste est utilisée dans l'expression`#{'this is ' + myList}`, il évaluera les`["this is one", "this is two"]`. Si vous avez deux listes, Data Pipeline les aplatit lors de leur évaluation. Par exemple, si `myList1` est défini comme `[1,2]` et `myList2` comme `[3,4]`, l'expression `[#{myList1}, #{myList2}]` est analysée comme `[1,2,3,4]`.

Expression de nœud

AWS Data Pipeline utilise l'expression `#{node.*}` dans `SnsAlarm` ou `PreCondition` pour créer une référence arrière à l'objet parent d'un composant du pipeline. Comme `SnsAlarm` et `PreCondition` sont référencés depuis une activité ou une ressource sans référence arrière à leur encontre, `node` offre le moyen de faire référence au référent. Par exemple, la définition de pipeline suivante illustre comment une notification d'échec peut utiliser `node` pour effectuer une référence à son parent, dans ce cas `ShellCommandActivity`, et inclure les heures de début et de fin planifiées du parent dans le message `SnsAlarm`. La référence scheduledStartTime sur ShellCommandActivity ne nécessite pas le préfixe `node`, car scheduledStartTime fait référence à lui-même.

> **Note**
>
> Les champs précédés par le signe AT (@) indiquent que ces champs sont des champs liés à l'exécution.

```
{
  "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 prend en charge les références transitives pour les champs définis par l'utilisateur, mais pas pour les champs liés à l'exécution. Une référence transitive est une référence entre deux composants d'un pipeline qui dépend d'un autre composant de pipeline comme intermédiaire. L'exemple suivant montre une référence à un champ transitif défini par l'utilisateur et une référence à un champ lié à l'exécution non transitif, les deux étant valides. Pour plus d'informations, consultez Champs définis par l'utilisateur.

```
{
  "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"
}
```

Evaluation d'expression

AWS Data Pipeline fournit un ensemble de fonctions que vous pouvez utiliser pour calculer la valeur d'un champ. L'exemple suivant utilise la fonction `makeDate` pour définir le champ `startDateTime` d'un objet `Schedule` avec la valeur `"2011-05-24T0:00:00"` (GMT/UTC).

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

Fonctions mathématiques

Les fonctions suivantes peuvent être utilisées avec des valeurs numériques.

Fonction	Description
+	Addition. Exemple : `#{1 + 2}` Résultat: `3`
-	Soustraction. Exemple : `#{1 - 2}` Résultat: `-1`
*	Multiplication. Exemple : `#{1 * 2}` Résultat: `2`
/	Division. Si vous divisez deux nombres entiers, le résultat est tronqué. Exemple : `#{1 / 2}`, résultat :`0` Exemple : `#{1.0 / 2}`, résultat :`.5`
^	Exposant. Exemple : `#{2 ^ 2}` Résultat: `4.0`

Fonctions de chaîne

Les fonctions suivantes peuvent être utilisées avec des valeurs chaîne (string).

Fonction	Description
+	Concaténation. Les valeurs autres que les valeurs chaîne sont converties d'abord en chaînes. Exemple : `#{"hel" + "lo"}` Résultat: `"hello"`

Fonctions de date et d'heure

Les fonctions suivantes peuvent être utilisées avec des valeurs date/heure (DateTime). A titre d'exemple, `myDateTime` a pour valeur `May 24, 2011 @ 5:10 pm GMT`.

> **Note**
>
> Le format date/heure d'AWS Data Pipeline est le temps Joda (Joda Time), qui remplace les classes Java de date et d'heure. Pour plus d'informations, consultez Temps Joda - Classe DateTimeFormat.

Fonction	Description
`int day(DateTime myDateTime)`	Obtient le jour de la valeur DateTime sous forme de nombre entier. Exemple : `#{day(myDateTime)}` Résultat: `24`

Fonction	Description
`int dayOfYear(DateTime myDateTime)`	Obtient le jour de l'année de la valeur DateTime sous forme de nombre entier. Exemple : `#{dayOfYear(myDateTime)}` Résultat: `144`
`DateTime firstOfMonth(DateTime myDateTime)`	Crée un objet DateTime pour le début du mois selon la valeur DateTime spécifiée. Exemple : `#{firstOfMonth(myDateTime)}` Résultat: `"2011-05-01T17:10:00z"`
`String format(DateTime myDateTime,String format)`	Crée un objet chaîne (string) qui est le résultat de la conversion de la valeur DateTime spécifiée à l'aide de la chaîne de format spécifiée. Exemple : `#{format(myDateTime,'YYYY-MM-dd HH:mm:ss z')}` Résultat: `"2011-05-24T17:10:00 UTC"`

Fonction	Description
`int hour(DateTime myDateTime)`	Obtient l'heure de la valeur DateTime sous forme de nombre entier. Exemple : `#{hour(myDateTime)}` Résultat: 17
`DateTime makeDate(int year,int month,int day)`	Crée un objet DateTime (UTC), avec l'année, le mois et le jour spécifiés, à minuit. Exemple : `#{makeDate(2011,5,24)}` Résultat: `"2011-05-24T0:00:00z"`
`DateTime makeDateTime(int year,int month,int day,int hour,int minute)`	Crée un objet DateTime (UTC), avec l'année, le mois, le jour, l'heure et la minute spécifiés. Exemple : `#{makeDateTime(2011,5,24,14,21)}` Résultat: `"2011-05-24T14:21:00z"`

Fonction	Description
`DateTime midnight(DateTime myDateTime)`	Crée un objet DateTime pour le minuit en cours, par rapport à la valeur DateTime spécifiée. Par exemple, lorsque `MyDateTime` est `2011-05-25T17:10:00z`, le résultat est le suivant. Exemple : `#{midnight(myDateTime)}` Résultat: `"2011-05-25T0:00:00z"`
`DateTime minusDays(DateTime myDateTime,int daysToSub)`	Crée un objet DateTime qui est le résultat de la soustraction du nombre de jours spécifiés de la valeur DateTime spécifiée. Exemple : `#{minusDays(myDateTime,1)}` Résultat: `"2011-05-23T17:10:00z"`
`DateTime minusHours(DateTime myDateTime,int hoursToSub)`	Crée un objet DateTime qui est le résultat de la soustraction du nombre d'heures spécifiées de la valeur DateTime spécifiée. Exemple : `#{minusHours(myDateTime,1)}` Résultat: `"2011-05-24T16:10:00z"`

Fonction	Description
`DateTime minusMinutes(DateTime myDateTime,int minutesToSub)`	Crée un objet DateTime qui est le résultat de la soustraction du nombre de minutes spécifiées de la valeur DateTime spécifiée. Exemple : `#{minusMinutes(myDateTime,1)}` Résultat: `"2011-05-24T17:09:00z"`
`DateTime minusMonths(DateTime myDateTime,int monthsToSub)`	Crée un objet DateTime qui est le résultat de la soustraction du nombre de mois spécifiés de la valeur DateTime spécifiée. Exemple : `#{minusMonths(myDateTime,1)}` Résultat: `"2011-04-24T17:10:00z"`
`DateTime minusWeeks(DateTime myDateTime,int weeksToSub)`	Crée un objet DateTime qui est le résultat de la soustraction du nombre de semaines spécifiées de la valeur DateTime spécifiée. Exemple : `#{minusWeeks(myDateTime,1)}` Résultat: `"2011-05-17T17:10:00z"`

Fonction	Description
`DateTime minusYears(DateTime myDateTime,int yearsToSub)`	Crée un objet DateTime qui est le résultat de la soustraction du nombre d'années spécifiées de la valeur DateTime spécifiée. Exemple : `#{minusYears(myDateTime,1)}` Résultat: `"2010-05-24T17:10:00z"`
`int minute(DateTime myDateTime)`	Obtient la minute de la valeur DateTime sous forme de nombre entier. Exemple : `#{minute(myDateTime)}` Résultat: `10`
`int month(DateTime myDateTime)`	Obtient le mois de la valeur DateTime sous forme de nombre entier. Exemple : `#{month(myDateTime)}` Résultat: 5

Fonction	Description
`DateTime plusDays(DateTime myDateTime,int daysToAdd)`	Crée un objet DateTime qui est le résultat de l'ajout du nombre de jours spécifiés à la valeur DateTime spécifiée. Exemple : `#{plusDays(myDateTime,1)}` Résultat: `"2011-05-25T17:10:00z"`
`DateTime plusHours(DateTime myDateTime,int hoursToAdd)`	Crée un objet DateTime qui est le résultat de l'ajout du nombre d'heures spécifiées à la valeur DateTime spécifiée. Exemple : `#{plusHours(myDateTime,1)}` Résultat: `"2011-05-24T18:10:00z"`
`DateTime plusMinutes(DateTime myDateTime,int minutesToAdd)`	Crée un objet DateTime qui est le résultat de l'ajout du nombre de minutes spécifiées à la valeur DateTime spécifiée . Exemple : `#{plusMinutes(myDateTime,1)}` Résultat: `"2011-05-24 17:11:00z"`

Fonction	Description
`DateTime plusMonths(DateTime myDateTime,int monthsToAdd)`	Crée un objet DateTime qui est le résultat de l'ajout du nombre de mois spécifiés à la valeur DateTime spécifiée. Exemple : `#{plusMonths(myDateTime,1)}` Résultat: `"2011-06-24T17:10:00z"`
`DateTime plusWeeks(DateTime myDateTime,int weeksToAdd)`	Crée un objet DateTime qui est le résultat de l'ajout du nombre de semaines spécifiées à la valeur DateTime spécifiée. Exemple : `#{plusWeeks(myDateTime,1)}` Résultat: `"2011-05-31T17:10:00z"`
`DateTime plusYears(DateTime myDateTime,int yearsToAdd)`	Crée un objet DateTime qui est le résultat de l'ajout du nombre d'années spécifiées à la valeur DateTime spécifiée. Exemple : `#{plusYears(myDateTime,1)}` Résultat: `"2012-05-24T17:10:00z"`

Fonction	Description
`DateTime sunday(DateTime myDateTime)`	Crée un objet DateTime pour le dimanche précédent , par rapport à la valeur DateTime spécifiée. Si la valeur DateTime spécifiée est un dimanche (Sunday), le résultat est la valeur DateTime spécifiée. Exemple : `#{sunday(myDateTime)}` Résultat: `"2011-05-22 17:10:00 UTC"`
`int year(DateTime myDateTime)`	Obtient l'année de la valeur DateTime sous forme de nombre entier. Exemple : `#{year(myDateTime)}` Résultat: `2011`
`DateTime yesterday(DateTime myDateTime)`	Crée un objet DateTime pour le jour précédent, par rapport à la valeur DateTime spécifiée . Le résultat est le même que celui de minusDays(1). Exemple : `#{yesterday(myDateTime)}` Résultat: `"2011-05-23T17:10:00z"`

Caractères spéciaux

AWS Data Pipeline utilise certains caractères qui ont une signification particulière dans les définitions de pipeline, comme indiqué dans le tableau suivant.

Caractère spécial	Description	Exemples
@	Champ disponible à l'exécution. Ce caractère est un préfixe de nom de champ dans le cas d'un champ qui n'est disponible que lorsqu'un pipeline s'exécute.	@actualStartTime @failureReason @resourceStatus
#	Expression. Les expressions sont délimitées par « #{ » et « } », et le contenu des accolades est évalué par AWS Data Pipeline. Pour plus d'informations, consultez Expressions.	#{format(myDateTime,'AAAA-MM-jj hh:mm:ss')} s3://mybucket/#{id}.csv
*	Champ chiffré. Ce caractère est un préfixe de nom de champ qui permet d'indiquer qu'AWS Data Pipeline doit chiffrer le contenu du champ en transit entre la console ou l'interface de ligne de commande et le service AWS Data Pipeline.	*password

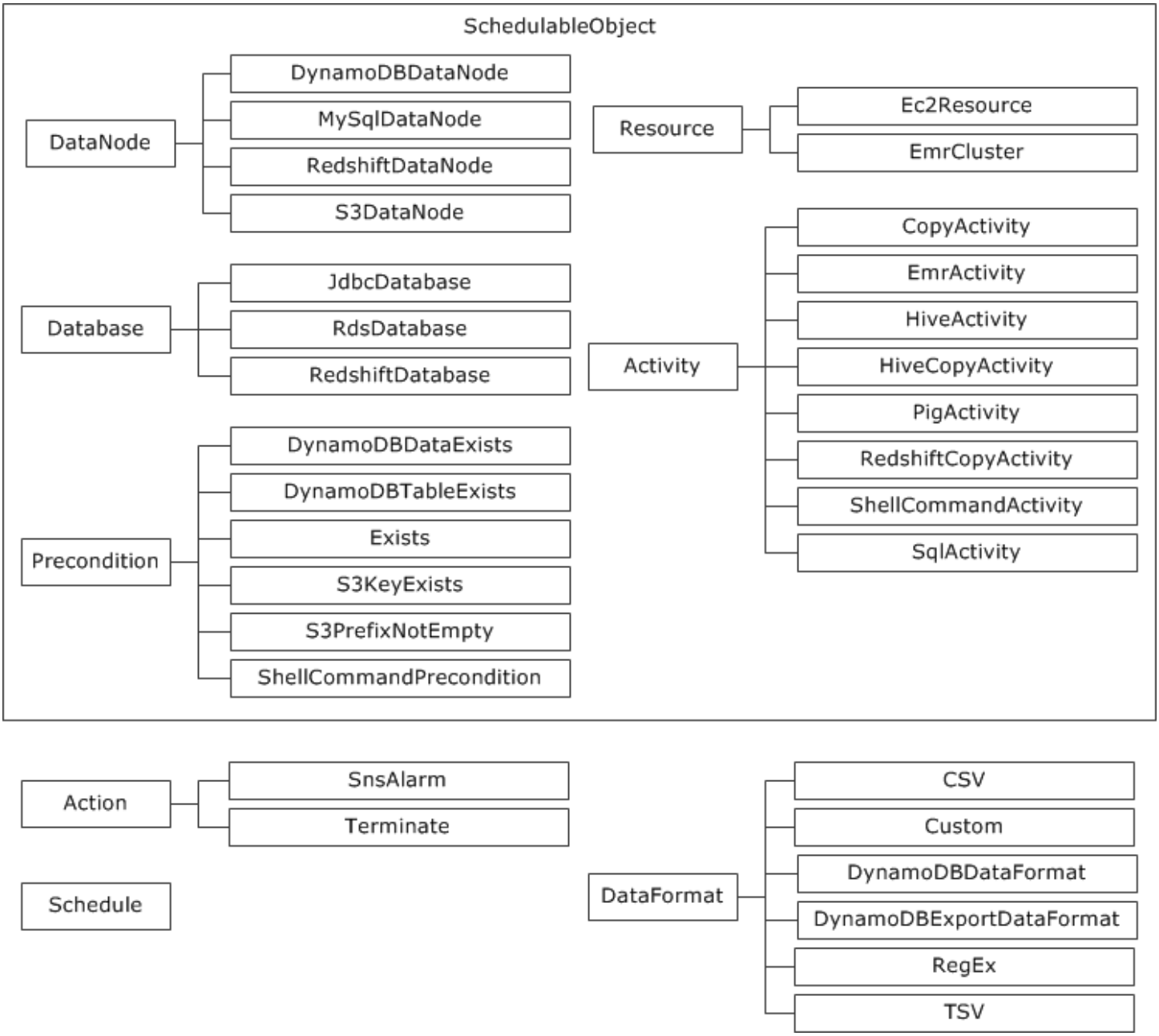
Référence d'objet de pipeline

Vous pouvez utiliser les objets et composants suivants dans la définition du pipeline.

Table des matières

> **Note**
>
> Pour un exemple d'application utilisant le SDK AWS Data Pipeline Java, voir Data Pipeline DynamoDB Export Java Sample on. GitHub

Le schéma suivant illustre la hiérarchie d'objets pour AWS Data Pipeline.



Nœuds de données

Les objets suivants représentent les objets de nœuds de données AWS Data Pipeline :

Objets

- DynamoDB DataNode
- MySqlDataNode
- RedshiftDataNode

- S3 DataNode
- SqlDataNode

DynamoDB DataNode

Définit un nœud de données à l'aide de DynamoDB, qui est spécifié en tant qu'entrée d'un objet or. `HiveActivity` `EMRActivity`

> **Note**
>
> L'objet `DynamoDBDataNode` ne prend pas en charge la condition préalable `Exists`.

Exemple

Voici un exemple de ce type d'objet. Cet objet référence deux autres objets que vous pourriez définir dans le même fichier de définition du pipeline. `CopyPeriod` est un objet `Schedule` et `Ready` est un objet de condition préalable.

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

Syntaxe

Champs obligatoires	Description	Type d'option
tableName	La table DynamoDB.	Chaîne

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle de planification. Les utilisateurs doivent spécifier une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifications (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour obtenir des exemples de configurations de planification facultatives, consultez la section Planification.	Objet de référence, par exemple, « schedule » : {"ref » : » myScheduleId «}

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si ce champ est défini, une activité à distance qui n'est pas exécutée dans l'intervalle de temps défini au départ peut être retentée.	Période

Champs facultatifs	Description	Type d'option
dataFormat	DataFormat pour les données décrites par ce nœud de données. Actuellement pris en charge pour HiveActivity et HiveCopyActivity.	Objet de référence , « DataFormat » : {"ref » « MyDynamoDB DataFormatId «}
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultanées d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}
readThroughputPerc ent	Définit la vitesse des opérations de lecture pour maintenir votre débit DynamoDB dans la plage allouée pour votre table. La valeur est un nombre double compris entre 0,1 et 1,0 (inclus).	Double

Champs facultatifs	Description	Type d'option
region	Code de la région dans laquelle la table DynamoDB existe. Par exemple, us-east-1. Ceci est utilisé HiveActivity lorsqu'il effectue une mise en scène pour les tables DynamoDB dans Hive.	Énumération
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourceId «}

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur runsOn et que workerGroup existe, workerGroup est ignoré.	Chaîne
writeThroughputPercent	Définit la vitesse des opérations d'écriture pour maintenir votre débit DynamoDB dans la plage allouée pour votre table. La valeur est un nombre double compris entre 0,1 et 1,0 (inclus).	Double

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

MySqlDataNode

Définit un nœud de données à l'aide de MySQL.

> **Note**
>
> Le type `MySqlDataNode` est obsolète. Nous vous recommandons d'utiliser à la place SqlDataNode.

Exemple

Voici un exemple de ce type d'objet. Cet objet référence deux autres objets que vous pourriez définir dans le même fichier de définition du pipeline. `CopyPeriod` est un objet `Schedule` et `Ready` est un objet de condition préalable.

```
{
  "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" }
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
table	Nom de la table dans la base de données MySQL.	Chaîne

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Les utilisate urs doivent spécifier une référence de planifica tion à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planifica tion de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets	Objet de référence , par exemple « schedule » : {"ref » : » mySchedul eld «}

Champs d'invocation de l'objet	Description	Type d'option
	héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifica tions (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
createTableSql	Expression SQL create table qui crée la table.	Chaîne
database	Nom de la base de données.	Objet de référence, par exemple « base de données » : {"ref » : » myDatabaseId «}
dependsOn	Spécifie la dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » :

Champs facultatifs	Description	Type d'option
		{"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
insertQuery	Instruction SQL pour insérer des données dans la table.	Chaîne
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}
reportProgressTime out	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourc eId «}

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération
schemaName	Nom du schéma contenant la table.	Chaîne
selectQuery	Instruction SQL pour récupérer les données de la table.	Chaîne
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur runsOn et que workerGroup existe, workerGro up est ignoré.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- S3 DataNode

RedshiftDataNode

Définit un nœud de données à l'aide d'Amazon Redshift. `RedshiftDataNode`représente les propriétés des données d'une base de données, telle qu'une table de données, utilisée par votre pipeline.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "id" : "MyRedshiftDataNode",
```

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

Syntaxe

Champs obligatoires	Description	Type d'option
database	Base de données dans laquelle réside la table.	Objet de référence , par exemple « database » : {"ref » : » myRedshif tDatabase Id "}
tableName	Nom de la table Amazon Redshift. La table est créée si elle n'existe pas déjà et que vous l'avez fournie createTableSql.	Chaîne

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Les utilisate urs doivent spécifier une référence de planifica tion à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planifica tion de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline	Objet de référence , par exemple « schedule » : {"ref » : » mySchedul eId «}

Champs d'invocation de l'objet	Description	Type d'option
	dispose d'une arborescence de planifications (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
createTableSql	Expression SQL permettant de créer la table dans la base de données. Nous vous recommandons de spécifier le schéma dans lequel la table doit être créée, par exemple : CREATE TABLE MySchema. MyTable (BestColumn varchar (25) clé primaire distkey, entier sortKey). numberOfWins AWS Data Pipelineexécute le script dans le createTableSql champ si la table, spécifiée par TableName, n'existe pas dans le schéma spécifié par le champ SchemaName. Par exemple, si vous spécifiez SchemaName comme MySchema mais que vous n'incluez	Chaîne

Champs facultatifs	Description	Type d'option
	pas MySchema dans le createTableSql champ, la table est créée dans le mauvais schéma (par défaut, elle sera créée dans PUBLIC). La raison en est qu'AWS Data Pipeline n'analyse pas vos instructions CREATE TABLE.	
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}
primaryKeys	Si vous ne spécifiez aucune valeur primaryKe ys pour la table de destination dans `RedShiftCopyActivity` , vous pouvez définir une liste de colonnes à l'aide de ce champ, qui agit alors en tant que mergeKey. Toutefois, si une clé primaire est définie dans une table Amazon Redshift, ce paramètre remplace la clé existante.	Chaîne

Champs facultatifs	Description	Type d'option
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourceId «}
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération

Champs facultatifs	Description	Type d'option
schemaName	Ce champ facultatif spécifie le nom du schéma de la table Amazon Redshift. S'il n'est pas spécifié, le nom du schéma est PUBLIC, qui est le schéma par défaut dans Amazon Redshift. Pour plus d'informations, consultez le manuel Amazon Redshift Database Developer Guide.	Chaîne
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur runsOn et que workerGroup existe, workerGroup est ignoré.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence, par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence, par exemple "cascadeFailedOn« : {" ref » : » myRunnableObject Id "}

Champs liés à l'exécution	Description	Type d'option
emrStepLog	Journaux d'étapes EMR disponibles uniquement sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution.	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime

Champs liés à l'exécution	Description	Type d'option
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

S3 DataNode

Définit un nœud de données à l'aide d'Amazon S3. Par défaut, le S3 DataNode utilise le chiffrement côté serveur. Si vous souhaitez désactiver cette option, définissez s3 EncryptionType sur NONE.

> **Note**
>
> Lorsque vous utilisez un `S3DataNode` comme entrée de `CopyActivity`, seuls les formats de données CSV et TSV sont pris en charge.

Exemple

Voici un exemple de ce type d'objet. Cet objet référence un autre objet que vous pourriez définir dans le même fichier de définition du pipeline. `CopyPeriod` est un objet `Schedule`.

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

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Les utilisate urs doivent spécifier une référence de planifica tion à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planifica	Objet de référence , par exemple « schedule » : {"ref » : » mySchedul eld «}

Champs d'invocation de l'objet	Description	Type d'option
	tion de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifica tions (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
compression	Type de compression des données décrit par le S3DataNode. « none » n'est pas une compressi on et « gzip » est compressé avec l'algorithme gzip. Ce champ n'est pris en charge que pour une utilisation avec Amazon Redshift et lorsque vous utilisez S3 DataNode avec. CopyActivity	Énumération
dataFormat	DataFormat pour les données décrites par ce S3DataNode.	Objet de référence , par exemple « dataFormat » :

Champs facultatifs	Description	Type d'option
		{"ref » : » myDataFormat Id "}
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
directoryPath	Chemin du répertoire Amazon S3 sous forme d'URI : s3://my-bucket/my-key-for-directory. Vous devez fournir une valeur filePath ou directoryPath.	Chaîne
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
filePath	Le chemin d'accès à l'objet dans Amazon S3 sous forme d'URI, par exemple : s3://my-bucket/my-key-for-file. Vous devez fournir une valeur filePath ou directoryPath. Ces valeurs représentent un dossier et un nom de fichier. Utilisez la valeur directoryPath pour accueillir plusieurs fichiers dans un répertoire.	Chaîne
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période

Champs facultatifs	Description	Type d'option
manifestFilePath	Le chemin Amazon S3 vers un fichier manifeste au format pris en charge par Amazon Redshift. AWS Data Pipelineutilise le fichier manifeste pour copier les fichiers Amazon S3 spécifiés dans la table. Ce champ n'est valide que lorsqu'un RedShiftCopyActivity fait référence au S3DataNode.	Chaîne
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}

Champs facultatifs	Description	Type d'option
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}
reportProgressTime out	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourc eId «}
s3 EncryptionType	Remplace le type de chiffrement Amazon S3. Les valeurs possibles sont SERVER_SI DE_ENCRYPTION ou NONE. Le chiffrement côté serveur est activé par défaut.	Énumération

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur runsOn et que workerGroup existe, workerGroup est ignoré.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence, par exemple « ActiveInstances » :

Champs liés à l'exécution	Description	Type d'option
		{"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnableObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

consultez aussi

- MySqlDataNode

SqlDataNode

Définit un nœud de données à l'aide de SQL.

Exemple

Voici un exemple de ce type d'objet. Cet objet référence deux autres objets que vous pourriez définir dans le même fichier de définition du pipeline. `CopyPeriod` est un objet `Schedule` et `Ready` est un objet de condition préalable.

```
{
  "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" }
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
table	Nom de la table dans la base de données SQL.	Chaîne

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle de planification. Les utilisateurs doivent spécifier une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifications (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	Objet de référence, par exemple « schedule » : {"ref » : » myScheduleId «}

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
createTableSql	Expression SQL create table qui crée la table.	Chaîne
database	Nom de la base de données.	Objet de référence, par exemple « base de données » : {"ref » : » myDatabas eId «}
dependsOn	Spécifie la dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
insertQuery	Instruction SQL pour insérer des données dans la table.	Chaîne
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont	Entier

Champs facultatifs	Description	Type d'option
	pas comptabilisées dans le nombre d'instances actives.	
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}

Champs facultatifs	Description	Type d'option
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourceId «}
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération

Champs facultatifs	Description	Type d'option
schemaName	Nom du schéma contenant la table.	Chaîne
selectQuery	Instruction SQL pour récupérer les données de la table.	Chaîne
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur runsOn et que workerGroup existe, workerGroup est ignoré.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence, par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence, par exemple "cascadeFailedOn« : {" ref » : » myRunnableObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquement sur les tentatives d'activité EMR	Chaîne

Champs liés à l'exécution	Description	Type d'option
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime

Champs liés à l'exécution	Description	Type d'option
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

consultez aussi

- S3 DataNode

Activités

Les objets suivants sont les objets d'activité AWS Data Pipeline :

Objets

- CopyActivity
- EmrActivity
- HadoopActivity
- HiveActivity
- HiveCopyActivity
- PigActivity
- RedshiftCopyActivity
- ShellCommandActivity
- SqlActivity

CopyActivity

Copie les données d'un emplacement à un autre. `CopyActivity`prend en charge S3 DataNode et SqlDataNode en entrée et en sortie et l'opération de copie est normalement effectuée record-by-record. `CopyActivity`Fournit toutefois une copie haute performance d'Amazon S3 vers Amazon S3 lorsque toutes les conditions suivantes sont remplies :

- L'entrée et la sortie sont S3 DataNodes
- Le champ `dataFormat` est le même pour l'entrée et pour la sortie.

Si vous fournissez des fichiers de données compressés en tant qu'entrées et ne l'indiquez pas à l'aide du champ `compression` des nœuds de données S3, `CopyActivity` risque d'échouer. Dans ce cas, `CopyActivity` ne détecte pas correctement la fin du caractère d'enregistrement et l'opération échoue. En outre, `CopyActivity` prend en charge la copie d'un répertoire vers un autre répertoire et la copie d'un fichier dans un répertoire, mais la record-by-record copie se produit lors de la copie d'un répertoire dans un fichier. Enfin, `CopyActivity` ne prend pas en charge la copie de fichiers Amazon S3 en plusieurs parties.

`CopyActivity` présente des limites spécifiques à sa prise en charge CSV. Lorsque vous utilisez un S3 DataNode comme entrée pour`CopyActivity`, vous ne pouvez utiliser qu'une variante Unix/Linux

du format de fichier de données CSV pour les champs d'entrée et de sortie Amazon S3. La variante Unix/Linux nécessite les éléments suivants :

- Le séparateur doit être la virgule (« , »).
- Les enregistrements ne sont pas entre guillemets.
- Le caractère d'échappement par défaut est la valeur ASCII 92 (barre oblique inverse).
- La fin de l'identifiant d'enregistrement est la valeur ASCII 10 (ou « \n »).

Les systèmes Windows utilisent généralement une séquence de end-of-record caractères différente : retour en chariot et alimentation en ligne en même temps (valeur ASCII 13 et valeur ASCII 10). Vous devez gérer cette différence à l'aide d'un mécanisme supplémentaire, tel qu'un script de pré-copie de script permettant de modifier les données d'entrée, afin de vous assurer que `CopyActivity` puisse correctement détecter la fin d'un enregistrement ; dans le cas contraire, `CopyActivity` échoue de manière répétée.

Lorsque vous utilisez `CopyActivity` pour exporter à partir d'un objet PostgreSQL RDS vers un format de données TSV, le caractère NULL par défaut est \n.

Exemple

Voici un exemple de ce type d'objet. Cet objet référence trois autres objets que vous pourriez définir dans le même fichier de définition du pipeline. `CopyPeriod` est un objet `Schedule`. `InputData` et `OutputData` sont des objets de nœud de données.

```
{
  "id" : "S3ToS3Copy",
  "type" : "CopyActivity",
  "schedule" : { "ref" : "CopyPeriod" },
  "input" : { "ref" : "InputData" },
  "output" : { "ref" : "OutputData" },
  "runsOn" : { "ref" : "MyEc2Resource" }
}
```

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle de planification. Les utilisateurs doivent spécifier une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifications (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	Objet de référence, par exemple « schedule » : {"ref » : » myScheduleId «}

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourceId «}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur runsOn et que workerGroup existe, workerGroup est ignoré.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération

Champs facultatifs	Description	Type d'option
input	Source de données d'entrée.	Objet de référence , par exemple « input » : {"ref » : » myDataNode Id "}
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
output	Source de données de sortie.	Objet de référence , par exemple « output » : {"ref » : » myDataNode Id "}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}
reportProgressTime out	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime

Champs liés à l'exécution	Description	Type d'option
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromI nstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthSta tusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime

Champs liés à l'exécution	Description	Type d'option
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnableObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne

Champs système	Description	Type d'option
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

consultez aussi

- ShellCommandActivity
- EmrActivity
- Exportez des données MySQL vers Amazon S3 à l'aide de AWS Data Pipeline

EmrActivity

Exécute un cluster EMR.

AWS Data Pipelineutilise un format d'étape différent de celui d'Amazon EMR ; par exemple, AWS Data Pipeline utilise des arguments séparés par des virgules après le nom du fichier JAR dans le champ de l'étape. `EmrActivity` L'exemple suivant montre une étape formatée pour Amazon EMR, suivie AWS Data Pipeline de son équivalent :

```
s3://example-bucket/MyWork.jar arg1 arg2 arg3
```

```
"s3://example-bucket/MyWork.jar,arg1,arg2,arg3"
```

Exemples

Voici un exemple de ce type d'objet. Cet exemple utilise d'anciennes versions d'Amazon EMR. Vérifiez l'exactitude de cet exemple avec la version du cluster Amazon EMR que vous utilisez.

Cet objet référence trois autres objets que vous pourriez définir dans le même fichier de définition du pipeline. `MyEmrCluster` est un objet `EmrCluster`. `MyS3Input` et `MyS3Output` sont des objets `S3DataNode`.

> **Note**
>
> Dans cet exemple, vous pouvez remplacer le champ `step` par votre chaîne de clusters souhaitée, qui peut être, entre autres, un script Pig, un cluster Hadoop Streaming ou votre propre fichier JAR personnalisé avec ses paramètres.

Hadoop 2.x (AMI 3.x)

```
{
  "id" : "MyEmrActivity",
  "type" : "EmrActivity",
  "runsOn" : { "ref" : "MyEmrCluster" },
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : ["s3://mybucket/myPath/myStep.jar,firstArg,secondArg,-files,s3://mybucket/
myPath/myFile.py,-input,s3://myinputbucket/path,-output,s3://myoutputbucket/path,-
mapper,myFile.py,-reducer,reducerName","s3://mybucket/myPath/myotherStep.jar,..."],
  "postStepCommand" : "scp localFiles remoteFiles",
  "input" : { "ref" : "MyS3Input" },
  "output" : { "ref" : "MyS3Output" }
}
```

> **Note**
>
> Pour transmettre des arguments à une application dans une étape, vous devez spécifier la région dans le chemin du script, comme indiqué dans l'exemple suivant. Il est également possible que vous deviez faire précéder les arguments que vous transmettez d'une séquence d'échappement. Par exemple, si vous utilisez `script-runner.jar` pour exécuter un script shell et que vous souhaitez transmettre des arguments au script, vous devez faire précéder les virgules qui les séparent d'une séquence d'échappement. L'extrait d'étape suivant montre comment procéder :
>
> ```
> "step" : "s3://eu-west-1.elasticmapreduce/libs/script-runner/script-
> runner.jar,s3://datapipeline/echo.sh,a\\\\,b\\\\,c"
> ```
>
> Cette étape utilise `script-runner.jar` pour exécuter le script shell `echo.sh` et transmet `a`, `b` et `c` comme un seul argument au script. Comme le premier caractère d'échappement est supprimé de l'argument obtenu, il se peut que vous ayez à nouveau besoin de le faire précéder d'une séquence d'échappement. Par exemple, si vous avez `File\.gz` comme

argument dans JSON, vous pouvez le faire précéder d'une séquence d'échappement avec `File\\\\.gz`. Cependant, comme la première séquence d'échappement est ignorée, vous devez utiliser `File\\\\\\\\.gz` .

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle de planification. Spécifiez une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Vous pouvez répondre à cette exigence en définissant explicitement une planification sur l'objet, par exemple, en spécifiant `"schedule": {"ref": "DefaultSchedule"}` . Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifications (planifications au sein de la planification maître), vous pouvez créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	Objet de référence , par exemple, « schedule » : {"ref » : » mySchedulEld «}

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	Le cluster Amazon EMR sur lequel cette tâche sera exécutée.	Objet de référence , par exemple, « RunSon » : {"ref » : » myEmrCluster Id "}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur `runsOn` et que `workerGroup` existe, `workerGroup` est ignoré.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence , par exemple, « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération

Champs facultatifs	Description	Type d'option
input	Emplacement des données d'entrée.	Objet de référence , par exemple, « input » : {"ref » : » myDataNode Id "}
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple, « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple, "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple, « onSuccess » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
output	Emplacement des données de sortie.	Objet de référence , par exemple, « output » : {"ref » : » myDataNode Id "}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple, « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI Amazon S3, tel que 's3 ://BucketName/ Prefix/ 'pour le téléchargement des journaux pour le pipeline.	Chaîne
postStepCommand	Scripts shell à exécuter une fois toutes les étapes terminées. Pour spécifier plusieurs scripts, jusqu'à 255, ajoutez plusieurs champs `postStepCommand` .	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple, « précondition » : {"ref » : » myPrecond itionId «}
preStepCommand	Scripts shell à exécuter avant l'exécution de toute étape. Pour spécifier plusieurs scripts, jusqu'à 255, ajoutez plusieurs champs `preStepCommand` .	Chaîne
reportProgressTime out	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période

Champs facultatifs	Description	Type d'option
resizeClusterBeforeCourir	Redimensionnez le cluster avant d'effectuer cette activité afin de l'adapter aux tables DynamoDB spécifiées en entrée ou en sortie. **Note** Si votre `EmrActivity` utilise un `DynamoDBDataNode` comme nœud de données en entrée ou sortie et que vous avez défini `resizeClusterBeforeRunning` sur `TRUE`, AWS Data Pipeline commence à utiliser les types d'instance `m3.xlarge`. Vos choix de type d'instance sont alors remplacés par `m3.xlarge`, ce qui peut accroître vos coûts mensuels.	Booléen
resizeClusterMaxInstances	Limite du nombre maximal d'instances qui peuvent être demandées par l'algorithme de redimensionnement.	Entier
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Les valeurs sont : `cron`, `ondemand` et `timeseries` (cron, à la demande et séries chronologiques). La planification `timeseries` signifie que les instances sont programmées à la fin de chaque intervalle. La planification `cron` signifie que les instances sont programmées au début de chaque intervalle. Une planification `ondemand` vous permet d'exécuter un pipeline une fois par activation. Vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification `ondemand`, elle doit être spécifiée dans l'objet par défaut et être le seul `scheduleType` spécifié pour les objets du pipeline. Pour utiliser des pipelines `ondemand`, vous devez appeler l'opération `ActivatePipeline` pour chaque exécution suivante.	Énumération
step	Une ou plusieurs étapes que le cluster doit exécuter. Pour spécifier plusieurs étapes, jusqu'à 255, ajoutez plusieurs champs step. Utilisez des arguments séparés par des virgules saisis après le nom de fichier JAR ; par exemple, `s3://example-bucket/MyWork.jar,arg1,arg2,arg3` .	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple, "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Les journaux d'étapes Amazon EMR sont disponibles uniquement pour les tentatives d'activité EMR	Chaîne
errorId	`errorId` si l'objet a échoué.	Chaîne
errorMessage	`errorMessage` si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple, « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- ShellCommandActivity
- CopyActivity
- EmrCluster

HadoopActivity

Exécute une MapReduce tâche sur un cluster. Le cluster peut être un cluster EMR géré par AWS Data Pipeline ou une autre ressource si vous en utilisez. TaskRunner HadoopActivity À utiliser lorsque vous souhaitez exécuter un travail en parallèle. Cela vous permet d'utiliser les ressources de planification du framework YARN ou du négociateur de MapReduce ressources dans Hadoop 1. Si vous souhaitez exécuter le travail de manière séquentielle à l'aide de l'action Amazon EMR Step, vous pouvez toujours utiliser. EmrActivity

Exemples

HadoopActivity à l'aide d'un cluster EMR géré par AWS Data Pipeline

L' HadoopActivity objet suivant utilise une EmrCluster ressource pour exécuter un programme :

```
 {
   "name": "MyHadoopActivity",
   "schedule": {"ref": "ResourcePeriod"},
   "runsOn": {"ref": “MyEmrCluster”},
   "type": "HadoopActivity",
   "preActivityTaskConfig":{"ref":"preTaskScriptConfig”},
   "jarUri": "/home/hadoop/contrib/streaming/hadoop-streaming.jar",
   "argument": [
     "-files",
     “s3://elasticmapreduce/samples/wordcount/wordSplitter.py“,
     "-mapper",
     "wordSplitter.py",
     "-reducer",
     "aggregate",
     "-input",
     "s3://elasticmapreduce/samples/wordcount/input/",
     "-output",
     “s3://test-bucket/MyHadoopActivity/#{@pipelineId}/
#{format(@scheduledStartTime,'YYYY-MM-dd')}"
   ],
   "maximumRetries": "0",
   "postActivityTaskConfig":{"ref":"postTaskScriptConfig”},
   "hadoopQueue" : “high”
 }
```

Voici le correspondant *`MyEmrCluster`*, qui configure les files d'attente FairScheduler et dans YARN pour les AMI basées sur Hadoop 2 :

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
   "hadoopSchedulerType" : "PARALLEL_FAIR_SCHEDULING",
  “amiVersion” : “3.7.0”,
  "bootstrapAction" : ["s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
\,high\,default,-z,yarn.scheduler.capacity.root.high.capacity=50,-
```

```
z,yarn.scheduler.capacity.root.low.capacity=10,-
z,yarn.scheduler.capacity.root.default.capacity=30”]
}
```

Voici ce que EmrCluster vous utilisez pour configurer FairScheduler dans 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"
          }
```

Les configurations suivantes CapacityScheduler pour EmrCluster les AMI basées sur Hadoop 2 :

```
{
      "id": "MyEmrCluster",
      "type": "EmrCluster",
      "hadoopSchedulerType": "PARALLEL_CAPACITY_SCHEDULING",
      "amiVersion": "3.7.0",
      "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
\\\\,high,-z,yarn.scheduler.capacity.root.high.capacity=40,-
z,yarn.scheduler.capacity.root.low.capacity=60"
    }
```

HadoopActivity en utilisant un cluster EMR existant

Dans cet exemple, vous utilisez workergroups et a TaskRunner pour exécuter un programme sur un cluster EMR existant. La définition de pipeline suivante permet HadoopActivity de :

- Exécutez un MapReduce programme uniquement sur *myWorkerGroup*des ressources. Pour de plus amples informations sur les groupes de travail, consultez Exécution de tâches sur des ressources existantes à l'aide de Task Runner.
- Exécuter une preActivityTask configuration et une postActivityTask configuration

```
{
```

```
  "objects": [
    {
      "argument": [
        "-files",
        "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
        "-mapper",
        "wordSplitter.py",
        "-reducer",
        "aggregate",
        "-input",
        "s3://elasticmapreduce/samples/wordcount/input/",
        "-output",
        "s3://test-bucket/MyHadoopActivity/#{@pipelineId}/
#{format(@scheduledStartTime,'YYYY-MM-dd')}"
      ],
      "id": "MyHadoopActivity",
      "jarUri": "/home/hadoop/contrib/streaming/hadoop-streaming.jar",
      "name": "MyHadoopActivity",
      "type": "HadoopActivity"
    },
    {
      "id": "SchedulePeriod",
      "startDateTime": "start_datetime",
      "name": "SchedulePeriod",
      "period": "1 day",
      "type": "Schedule",
      "endDateTime": "end_datetime"
    },
    {
      "id": "ShellScriptConfig",
      "scriptUri": "s3://test-bucket/scripts/preTaskScript.sh",
      "name": "preTaskScriptConfig",
      "scriptArgument": [
        "test",
        "argument"
      ],
      "type": "ShellScriptConfig"
    },
    {
      "id": "ShellScriptConfig",
      "scriptUri": "s3://test-bucket/scripts/postTaskScript.sh",
      "name": "postTaskScriptConfig",
      "scriptArgument": [
        "test",
```

```
        "argument"
      ],
      "type": "ShellScriptConfig"
    },
    {
      "id": "Default",
      "scheduleType": "cron",
      "schedule": {
        "ref": "SchedulePeriod"
      },
      "name": "Default",
      "pipelineLogUri": "s3://test-bucket/logs/2015-05-22T18:02:00.343Z642f3fe415",
      "maximumRetries": "0",
      "workerGroup": "myWorkerGroup",
      "preActivityTaskConfig": {
        "ref": "preTaskScriptConfig"
      },
      "postActivityTaskConfig": {
        "ref": "postTaskScriptConfig"
      }
    }
  ]
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
jarUri	Emplacement d'un fichier JAR dans Amazon S3 ou dans le système de fichiers local du cluster à exécuter HadoopActivity.	Chaîne

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Les utilisate urs doivent spécifier une référence de planifica tion à un autre objet pour définir l'ordre	Objet de référence , par exemple « schedule » :

Champs d'invocation de l'objet	Description	Type d'option
	d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifications (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	{"ref » : » mySchedulEld «}

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	Cluster EMR sur lequel la tâche s'exécute.	Objet de référence, par exemple « RunSon » : {"ref » : » myEmrCluster Id "}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur	Chaîne

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
	runsOn et que workerGroup existe, workerGroup est ignoré.	

Champs facultatifs	Description	Type d'option
argument	Arguments à passer au fichier JAR.	Chaîne
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
hadoopQueue	Nom de la file d'attente du planificateur Hadoop dans laquelle l'activité est envoyée.	Chaîne
input	Emplacement des données d'entrée.	Objet de référence , par exemple « input » : {"ref » : » myDataNode Id "}

Champs facultatifs	Description	Type d'option
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
mainClass	La classe principale du fichier JAR avec lequel vous exécutez HadoopActivity.	Chaîne
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
output	Emplacement des données de sortie.	Objet de référence , par exemple « output » : {"ref » : » myDataNode Id "}

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
postActivityTaskConfig	Script de configuration de post-activité à exécuter. Se compose d'un URI du script shell dans Amazon S3 et d'une liste d'arguments.	Objet de référence, par exemple "postActivityTaskConfig » : {"ref » : » myShellScript ConfigId «}
preActivityTaskConfig	Script de configuration de pré-activité à exécuter. Se compose d'un URI du script shell dans Amazon S3 et d'une liste d'arguments.	Objet de référence, par exemple "preActivityTaskConfig » : {"ref » : » myShellScript ConfigId «}
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecondition Id «}
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime

Champs liés à l'exécution	Description	Type d'option
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromI nstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthSta tusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime

Champs liés à l'exécution	Description	Type d'option
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnableObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne

Champs système	Description	Type d'option
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- ShellCommandActivity
- CopyActivity
- EmrCluster

HiveActivity

Exécute une requête Hive sur un cluster EMR. `HiveActivity`facilite la configuration d'une activité Amazon EMR et crée automatiquement des tables Hive en fonction des données d'entrée provenant d'Amazon S3 ou d'Amazon RDS. Vous devez uniquement spécifier la requête HiveQL à exécuter sur la source de données. AWS Data Pipeline crée automatiquement les tables Hive avec `${input1}`, `${input2}`, etc. en fonction des champs d'entrée de l'objet `HiveActivity`.

Pour les entrées Amazon S3, le `dataFormat` champ est utilisé pour créer les noms des colonnes Hive.

Pour les entrées MySQL (Amazon RDS), les noms de colonne de la requête SQL sont utilisés pour créer les noms de colonnes Hive.

> **Note**
>
> Cette activité utilise la sérialisation/désérialisation (Serde) CSV de Hive.

Exemple

Voici un exemple de ce type d'objet. Cet objet référence trois autres objets que vous définissez dans le même fichier de définition du pipeline. `MySchedule` est un objet `Schedule`. `MyS3Input` et `MyS3Output` sont des objets de nœud de données.

```
{
  "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" }
}
```

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Spécifiez une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Vous pouvez satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planifica tion de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifica tions (planifications au sein de la planification maître), vous pouvez créer un objet parent ayant une référence de planification. Pour plus	Objet de référence , par exemple « schedule » : {"ref » : » mySchedul eId «}

Champs d'invocation de l'objet	Description	Type d'option
	d'informations sur les exemples de configura tions de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Groupe obligatoire (l'un des groupes suivants est obligatoi re)	Description	Type d'option
hiveScript	Script Hive à exécuter.	Chaîne
scriptUri	Emplacement du script Hive à exécuter (par exemple, s3://scriptLocation).	Chaîne

Groupe obligatoire	Description	Type d'option
runsOn	Cluster EMR sur lequel `HiveActivity` s'exécute.	Objet de référence , par exemple « RunSon » : {"ref » : » myEmrClus ter Id "}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur `runsOn` et que `workerGroup` existe, `workerGroup` est ignoré.	Chaîne
input	Source de données d'entrée.	Objet de référence , tel que « input » :

Groupe obligatoire	Description	Type d'option
		{"ref » : » myDataNod e Id "}
output	Source de données de sortie.	Objet de référence , tel que « output » : {"ref » : » myDataNod e Id "}

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence, tel que « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
hadoopQueue	Nom de la file d'attente du programmeur Hadoop dans laquelle la tâche sera envoyée.	Chaîne
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période

Champs facultatifs	Description	Type d'option
maxActiveInstances	Nombre maximal d'instances actives simultanées d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence, tel que « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence, tel que "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence, tel que « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence, tel que « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne

Champs facultatifs	Description	Type d'option
postActivityTaskConfig	Script de configuration de post-activité à exécuter. Se compose d'un URI du script shell dans Amazon S3 et d'une liste d'arguments.	Objet de référence, tel que "postActivityTaskConfig » : {"ref » : » myShellScript ConfigId «}
preActivityTaskConfig	Script de configuration de pré-activité à exécuter. Se compose d'un URI du script shell dans Amazon S3 et d'une liste d'arguments.	Objet de référence, tel que "preActivityTaskConfig » : {"ref » : » myShellScript ConfigId «}
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence, tel que « précondition » : {"ref » : » myPreconditionId «}
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période

Champs facultatifs	Description	Type d'option
resizeClusterBeforeCourir	Redimensionnez le cluster avant d'effectuer cette activité pour prendre en charge les nœuds de données DynamoDB spécifiés en entrée ou en sortie. **Note** Si votre activité utilise un `DynamoDBDataNode` comme nœud de données en entrée ou sortie et que vous avez défini `resizeClusterBeforeRunning` sur `TRUE`, AWS Data Pipeline commence à utiliser les types d'instance `m3.xlarge`. Vos choix de type d'instance sont alors remplacés par `m3.xlarge`, ce qui peut accroître vos coûts mensuels.	Booléen
resizeClusterMaxInstances	Limite du nombre maximal d'instances qui peuvent être demandées par l'algorithme de redimensionnement.	Entier
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération

Champs facultatifs	Description	Type d'option
scriptVariable	Spécifie les variables de script qu'Amazon EMR doit transmettre à Hive lors de l'exécuti on d'un script. Les exemples de variables de script suivants transmettent, respectiv ement, une variable SAMPLE et une variable FILTER_DATE à Hive : `SAMPLE=s3://elasticmapreduce/samples/hive-ads` et `FILTER_DATE=#{form at(@scheduledStartTime,'YYYY-MM-dd')}%` . Ce champ accepte plusieurs valeurs et fonctionne avec les champs `script` et `scriptUri` . En outre, `scriptVariable` fonctionne que l'étape soit définie sur `true` ou `false`. Ce champ est particulièrement utile pour envoyer des valeurs dynamiques à Hive en utilisant des expressions et des fonctions AWS Data Pipeline.	Chaîne
stage	Détermine si le transit est activé avant ou après l'exécution du script. Ce champ n'étant pas autorisé avec Hive 11, utilisez un AMI Amazon EMR version 3.2.0 ou ultérieure.	Booléen

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , tel que « ActiveIns tances » : {"ref » : » myRunnableObject Id "}

Champs liés à l'exécution	Description	Type d'option
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , tel que "cascadeF ailedOn« : {" ref » : » myRunnableObject Id "}
emrStepLog	Les journaux d'étapes Amazon EMR ne sont disponibles que pour les tentatives d'activité EMR.	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromI nstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@ healthSta tusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTi me	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCom pletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour un objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour un objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence, tel que « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- ShellCommandActivity
- EmrActivity

HiveCopyActivity

Exécute une requête Hive sur un cluster EMR. `HiveCopyActivity`facilite la copie de données entre les tables DynamoDB. `HiveCopyActivity`accepte une instruction HiveQL pour filtrer les données d'entrée de DynamoDB au niveau des colonnes et des lignes.

Exemple

L'exemple suivant montre comment utiliser `HiveCopyActivity` et `DynamoDBExportDataFormat` pour copier les données d'un `DynamoDBDataNode` dans un autre, tout en filtrant les données, en fonction de la date et de l'heure.

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

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle de planification. Les utilisateurs doivent spécifier une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifications (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	Objet de référence, par exemple « schedule » : {"ref » : » myScheduleId «}

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	Spécifie le cluster sur lequel lancer l'exécution.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourceId «}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur `runsOn` et que `workerGroup` existe, `workerGroup` est ignoré.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
dependsOn	Spécifie la dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération

Champs facultatifs	Description	Type d'option
filterSql	Fragment d'instruction SQL Hive qui filtre un sous-ensemble de données DynamoDB ou Amazon S3 à copier. Le filtre doit contenir uniquement des prédicats et il ne doit pas commencer par une clause `WHERE`, car AWS Data Pipeline ajoute celle-ci automatiquement.	Chaîne
input	Source de données d'entrée. Ce champ doit correspondre à `S3DataNode` ou `DynamoDBDataNode`. Si vous utilisez `DynamoDBNode`, spécifiez un `DynamoDBExportDataFormat`.	Objet de référence, par exemple « input » : {"ref » : » myDataNode Id "}
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini sur`ondemand`.	Période
maxActiveInstances	Nombre maximal d'instances actives simultanées d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence, par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence, par exemple "onLateAction« : {" ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
output	Source de données de sortie. Si l'entrée est `S3DataNode` , la sortie doit être `DynamoDBDataNode` . Sinon, la valeur peut être `S3DataNode` ou `DynamoDBDataNode` . Si vous utilisez `DynamoDBNode` , spécifiez un `DynamoDBExportDataFormat` .	Objet de référence , par exemple « output » : {"ref » : » myDataNode Id "}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI Amazon S3, par exemple `'s3://BucketName/Key/'` , pour le téléchargement des journaux pour le pipeline.	Chaîne
postActivityTaskConfig	Script de configuration de post-activité à exécuter. Se compose d'un URI du script shell dans Amazon S3 et d'une liste d'arguments.	Objet de référence, par exemple "postActivityTaskConfig » : {"ref » : » myShellScript ConfigId «}
preActivityTaskConfig	Script de configuration de pré-activité à exécuter. Se compose d'un URI du script shell dans Amazon S3 et d'une liste d'arguments.	Objet de référence, par exemple "preActivityTaskConfig » : {"ref » : » myShellScript ConfigId «}

Champs facultatifs	Description	Type d'option
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}
reportProgressTime out	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
resizeClusterBefor eCourir	Redimensionnez le cluster avant d'effectu er cette activité pour prendre en charge les nœuds de données DynamoDB spécifiés en entrée ou en sortie. **Note** Si votre activité utilise un `DynamoDBD ataNode` comme nœud de données en entrée ou sortie et que vous avez défini `resizeClusterBefor eRunning` sur `TRUE`, AWS Data Pipeline commence à utiliser les types d'instance `m3.xlarge` . Vos choix de type d'instance sont alors remplacés par `m3.xlarge` , ce qui peut accroître vos coûts mensuels.	Booléen
resizeClusterMaxIn stances	Limite du nombre maximal d'instances qui peuvent être demandées par l'algorithme de redimensionnement.	Entier

Champs facultatifs	Description	Type d'option
retryDelay	Délai entre deux nouvelles tentatives.	Période
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}

Champs liés à l'exécution	Description	Type d'option
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Les journaux d'étapes Amazon EMR ne sont disponibles que pour les tentatives d'activité EMR.	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromI nstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@ healthSta tusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTi me	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCom pletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- ShellCommandActivity
- EmrActivity

PigActivity

PigActivity fournit un support natif pour les scripts Pig AWS Data Pipeline sans qu'il soit nécessaire d'utiliser `ShellCommandActivity` ou`EmrActivity`. En outre, PigActivity prend en charge le transfert des données. Lorsque le champ « stage » est défini sur true, AWS Data Pipeline prépare les données d'entrée en tant que schéma dans Pig, sans code supplémentaire de l'utilisateur.

Exemple

L'exemple de pipeline suivant montre comment utiliser `PigActivity`. L'exemple de pipeline effectue les étapes suivantes :

- MyPigActivity1 charge des données depuis Amazon S3 et exécute un script Pig qui sélectionne quelques colonnes de données et les télécharge sur Amazon S3.
- MyPigActivity2 charge la première sortie, sélectionne quelques colonnes et trois lignes de données, puis la télécharge sur Amazon S3 en tant que deuxième sortie.
- MyPigActivity3 charge les deuxièmes données de sortie, insère deux lignes de données et uniquement la colonne nommée « cinquième » sur Amazon RDS.
- MyPigActivity4 charge les données Amazon RDS, sélectionne la première ligne de données et les télécharge sur Amazon S3.

```
{
  "objects": [
    {
      "id": "MyInputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "directoryPath": "s3://example-bucket/pigTestInput",
      "name": "MyInputData1",
      "dataFormat": {
        "ref": "MyInputDataType1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyPigActivity4",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyOutputData3"
      },
      "pipelineLogUri": "s3://example-bucket/path/",
      "name": "MyPigActivity4",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "type": "PigActivity",
      "dependsOn": {
        "ref": "MyPigActivity3"
      },
      "output": {
        "ref": "MyOutputData4"
      },
      "script": "B = LIMIT ${input1} 1; ${output1} = FOREACH B GENERATE one;",
      "stage": "true"
    },
    {
      "id": "MyPigActivity3",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
```

```
      },
      "input": {
        "ref": "MyOutputData2"
      },
      "pipelineLogUri": "s3://example-bucket/path",
      "name": "MyPigActivity3",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "script": "B = LIMIT ${input1} 2; ${output1} = FOREACH B GENERATE Fifth;",
      "type": "PigActivity",
      "dependsOn": {
        "ref": "MyPigActivity2"
      },
      "output": {
        "ref": "MyOutputData3"
      },
      "stage": "true"
    },
    {
      "id": "MyOutputData2",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "name": "MyOutputData2",
      "directoryPath": "s3://example-bucket/PigActivityOutput2",
      "dataFormat": {
        "ref": "MyOutputDataType2"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyOutputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "name": "MyOutputData1",
      "directoryPath": "s3://example-bucket/PigActivityOutput1",
      "dataFormat": {
        "ref": "MyOutputDataType1"
      },
      "type": "S3DataNode"
    },
    {
```

```
      "id": "MyInputDataType1",
      "name": "MyInputDataType1",
      "column": [
        "First STRING",
        "Second STRING",
        "Third STRING",
        "Fourth STRING",
        "Fifth STRING",
        "Sixth STRING",
        "Seventh STRING",
        "Eighth STRING",
        "Ninth STRING",
        "Tenth STRING"
      ],
      "inputRegEx": "^(\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\\S+)",
      "type": "RegEx"
    },
    {
      "id": "MyEmrResource",
      "region": "us-east-1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "keyPair": "example-keypair",
      "masterInstanceType": "m1.small",
      "enableDebugging": "true",
      "name": "MyEmrResource",
      "actionOnTaskFailure": "continue",
      "type": "EmrCluster"
    },
    {
      "id": "MyOutputDataType4",
      "name": "MyOutputDataType4",
      "column": "one STRING",
      "type": "CSV"
    },
    {
      "id": "MyOutputData4",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "directoryPath": "s3://example-bucket/PigActivityOutput3",
      "name": "MyOutputData4",
```

```
      "dataFormat": {
        "ref": "MyOutputDataType4"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyOutputDataType1",
      "name": "MyOutputDataType1",
      "column": [
        "First STRING",
        "Second STRING",
        "Third STRING",
        "Fourth STRING",
        "Fifth STRING",
        "Sixth STRING",
        "Seventh STRING",
        "Eighth STRING"
      ],
      "columnSeparator": "*",
      "type": "Custom"
    },
    {
      "id": "MyOutputData3",
      "username": "___",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "insertQuery": "insert into #{table} (one) values (?)",
      "name": "MyOutputData3",
      "*password": "___",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "connectionString": "jdbc:mysql://example-database-instance:3306/example-
database",
      "selectQuery": "select * from #{table}",
      "table": "example-table-name",
      "type": "MySqlDataNode"
    },
    {
      "id": "MyOutputDataType2",
      "name": "MyOutputDataType2",
      "column": [
        "Third STRING",
```

```
        "Fourth STRING",
        "Fifth STRING",
        "Sixth STRING",
        "Seventh STRING",
        "Eighth STRING"
      ],
      "type": "TSV"
    },
    {
      "id": "MyPigActivity2",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyOutputData1"
      },
      "pipelineLogUri": "s3://example-bucket/path",
      "name": "MyPigActivity2",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "dependsOn": {
        "ref": "MyPigActivity1"
      },
      "type": "PigActivity",
      "script": "B = LIMIT ${input1} 3; ${output1} = FOREACH B GENERATE Third, Fourth,
Fifth, Sixth, Seventh, Eighth;",
      "output": {
        "ref": "MyOutputData2"
      },
      "stage": "true"
    },
    {
      "id": "MyEmrResourcePeriod",
      "startDateTime": "2013-05-20T00:00:00",
      "name": "MyEmrResourcePeriod",
      "period": "1 day",
      "type": "Schedule",
      "endDateTime": "2013-05-21T00:00:00"
    },
    {
      "id": "MyPigActivity1",
      "scheduleType": "CRON",
```

```
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyInputData1"
      },
      "pipelineLogUri": "s3://example-bucket/path",
      "scriptUri": "s3://example-bucket/script/pigTestScipt.q",
      "name": "MyPigActivity1",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "scriptVariable": [
        "column1=First",
        "column2=Second",
        "three=3"
      ],
      "type": "PigActivity",
      "output": {
        "ref": "MyOutputData1"
      },
      "stage": "true"
    }
  ]
}
```

Le contenu de `pigTestScript.q` est le suivant.

```
B = LIMIT ${input1} $three; ${output1} = FOREACH B GENERATE $column1, $column2, Third,
 Fourth, Fifth, Sixth, Seventh, Eighth;
```

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle de planification. Les utilisateurs doivent spécifier une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Les	Objet de référence, par exemple, « schedule » : {"ref » : » mySchedulEld «}

Champs d'invocation de l'objet	Description	Type d'option
	utilisateurs peuvent satisfaire à cette exigence en définissant explicitement un calendrier sur l'objet, par exemple en spécifiant « schedule » : {"ref » : "DefaultSchedule«}. Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une arborescence de planifications (planifications au sein de la planification maître), les utilisateurs peuvent créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
script	Script Pig à exécuter.	Chaîne
scriptUri	Emplacement du script Pig à exécuter (par exemple, s3://scriptLocation).	Chaîne

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	Cluster EMR sur lequel cela s' PigActivity exécute.	Objet de référence , par exemple, « RunSon » : {"ref » : » myEmrCluster Id "}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur `runsOn` et que `workerGroup` existe, `workerGroup` est ignoré.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
dependsOn	Spécifie la dépendance sur un autre objet exécutable.	Objet de référence , par exemple, « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération

Champs facultatifs	Description	Type d'option
input	Source de données d'entrée.	Objet de référence , par exemple, « input » : {"ref » : » myDataNode Id "}
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple, « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple, "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple, « onSuccess » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
output	Source de données de sortie.	Objet de référence , par exemple, « output » : {"ref » : » myDataNode Id "}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple, « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI Amazon S3 (tel que 's3 ://BucketName/ Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
postActivityTaskCo nfig	Script de configuration de post-activité à exécuter. Il s'agit d'un URI du script shell dans Amazon S33 et d'une liste d'arguments.	Objet de référence , par exemple, "postActivityTaskC onfig » : {"ref » : » myShellScript ConfigId «}
preActivityTaskConfig	Script de configuration de pré-activité à exécuter. Se compose d'un URI du script shell dans Amazon S3 et d'une liste d'arguments.	Objet de référence , par exemple, "preActivityTaskCo nfig » : {"ref » : » myShellScript ConfigId «}
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple, « précondition » : {"ref » : » myPrecond itionId «}

Champs facultatifs	Description	Type d'option
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
resizeClusterBeforeCourir	Redimensionnez le cluster avant d'effectuer cette activité pour prendre en charge les nœuds de données DynamoDB spécifiés en entrée ou en sortie. **Note** Si votre activité utilise un `DynamoDBDataNode` comme nœud de données en entrée ou sortie et que vous avez défini `resizeClusterBeforeRunning` sur `TRUE`, AWS Data Pipeline commence à utiliser les types d'instance `m3.xlarge` . Vos choix de type d'instance sont alors remplacés par `m3.xlarge` , ce qui peut accroître vos coûts mensuels.	Booléen
resizeClusterMaxInstances	Limite du nombre maximal d'instances qui peuvent être demandées par l'algorithme de redimensionnement.	Entier
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Dans la planification de type séries chronologiques, les instances sont planifiées à la fin de chaque intervalle et dans la planification de type cron, les instances sont planifiées au début de chaque intervalle. Une planification à la demande vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul scheduleType pour les objets du pipeline. Pour utiliser des pipelines à la demande, il suffit d'appeler l' ActivatePipeline opération pour chaque exécution suivante. Les valeurs sont : cron, ondemand et timeseries (cron, à la demande et séries chronologiques).	Énumération
scriptVariable	Arguments à transmettre au script Pig. Vous pouvez utiliser scriptVariable avec script ou scriptUri.	Chaîne
stage	Détermine si la gestion intermédiaire est activée et permet à votre script Pig d'avoir accès aux tables de données mises en lots, telles que ${INPUT1} et ${OUTPUT1}.	Booléen

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple, « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple, "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Les journaux d'étapes Amazon EMR ne sont disponibles que pour les tentatives d'activité EMR.	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple, « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- ShellCommandActivity
- EmrActivity

RedshiftCopyActivity

Copie les données depuis DynamoDB ou Amazon S3 vers Amazon Redshift. Vous pouvez charger les données dans une nouvelle table ou les fusionner facilement dans une table existante.

Voici une présentation d'un cas d'utilisation dans lequel vous pouvez utiliser `RedshiftCopyActivity` :

1. Commencez par utiliser AWS Data Pipeline pour stocker vos données dans Amazon S3.

2. `RedshiftCopyActivity`À utiliser pour déplacer les données d'Amazon RDS et Amazon EMR vers Amazon Redshift.

   Cela vous permet de charger vos données dans Amazon Redshift où vous pouvez les analyser.
3. SqlActivityÀ utiliser pour exécuter des requêtes SQL sur les données que vous avez chargées dans Amazon Redshift.

En outre, `RedshiftCopyActivity` prend en charge un fichier manifeste et vous permet donc d'utiliser un `S3DataNode`. Pour plus d'informations, consultez S3 DataNode.

Exemple

Voici un exemple de ce type d'objet.

Pour prendre en charge les formats de conversion, cet exemple utilise les paramètres de conversion spéciaux EMPTYASNULL et IGNOREBLANKLINES dans `commandOptions`. Pour plus d'informations, consultez la section Paramètres de conversion des données dans le manuel Amazon Redshift Database Developer Guide.

```
{
  "id" : "S3ToRedshiftCopyActivity",
  "type" : "RedshiftCopyActivity",
  "input" : { "ref": "MyS3DataNode" },
  "output" : { "ref": "MyRedshiftDataNode" },
  "insertMode" : "KEEP_EXISTING",
  "schedule" : { "ref": "Hour" },
  "runsOn" : { "ref": "MyEc2Resource" },
  "commandOptions": ["EMPTYASNULL", "IGNOREBLANKLINES"]
}
```

L'exemple de définition de pipeline suivant illustre une activité qui utilise le mode d'insertion `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"
      }
    }
  ]
}
```

APPENDL'opération ajoute des éléments à une table, quelles que soient les clés primaires ou les clés de tri. Par exemple, si vous avez le tableau suivant, vous pouvez ajouter un enregistrement avec les mêmes valeurs d'ID et d'utilisateur.

```
ID(PK)     USER
1          aaa
2          bbb
```

Vous pouvez ajouter un enregistrement avec les mêmes valeurs d'ID et d'utilisateur.

```
ID(PK)     USER
1          aaa
2          bbb
1          aaa
```

> **Note**
>
> Si une opération APPEND est interrompue et retentée, le pipeline de réexécution résultant ajoute potentiellement depuis le début. Comme cela peut entraîner de nouvelles duplications, soyez conscient de ce comportement, en particulier si vous avez une logique qui comptabilise le nombre de lignes.

Pour obtenir un didacticiel, consultez Copier des données vers Amazon Redshift à l'aide de AWS Data Pipeline.

Syntaxe

Champs obligatoires	Description	Type d'option
insertMode	Détermine comment AWS Data Pipeline traite les données pré-existantes de la table cible qui chevauchent des lignes de données à charger. Les valeurs valides sont : KEEP_EXISTING , OVERWRITE_EXISTING , TRUNCATE et APPEND. KEEP_EXISTING ajoute de nouvelles lignes à la table, en conservant toutes les lignes existantes non modifiées.	Énumération

Champs obligatoires	Description	Type d'option
	KEEP_EXISTING et OVERWRITE_EXISTING utilise les clés primaire, de tri et de distribution pour identifier les lignes entrantes à associer aux lignes existantes. Consultez la section Mise à jour et insertion de nouvelles données dans le manuel Amazon Redshift Database Developer Guide. TRUNCATE supprime toutes les données de la table de destination avant d'écrire les nouvelles données. APPEND ajoute tous les enregistrements à la fin de la table Redshift. APPEND ne nécessite aucune clé primaire, de distribution ou de tri, par conséquent, des doublons potentiels peuvent être ajoutés.	

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Spécifiez une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Dans la plupart des cas, nous vous recommand ons de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Vous pouvez, par exemple, définir explicite ment une planification sur l'objet en spécifian	Objet de référence, tel que : "schedule":{"ref": "mySchedu leId"}

Champs d'invocation de l'objet	Description	Type d'option
	t `"schedule": {"ref": "DefaultSchedule"}` . Si la planification maître de votre pipeline contient des planifications imbriquées, créez un objet parent ayant une référence de planification. Pour obtenir des exemples de configurations de planification facultatives, consultez la section Planification.	

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourceId «}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur `runsOn` et que `workerGroup` existe, workerGroup est ignoré.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
commandOptions	Prend des paramètres à transmettre au nœud de données Amazon Redshift pendant l'`COPY`opération. Pour plus d'informations sur les paramètres, consultez COPY dans le manuel Amazon Redshift Database Developer Guide. Lorsqu'elle charge la table, la commande `COPY` tente implicitement de convertir les chaînes dans le type de données de la colonne cible. En plus des conversions de données par défaut qui s'exécutent de façon automatique, si vous rencontrez des erreurs ou si vous avez d'autres besoins de conversion, vous pouvez spécifier des paramètres de conversion supplémen taires. Pour plus d'informations, consultez la section Paramètres de conversion des données dans le manuel Amazon Redshift Database Developer Guide. Si un format de données est associé au nœud de données d'entrée ou de sortie, les paramètres fournis sont ignorés. Dans la mesure où l'opération de copie utilise d'abord `COPY` pour insérer des données dans une table intermédiaire, puis utilise une commande `INSERT` pour copier les données de	Chaîne

Champs facultatifs	Description	Type d'option
	la table intermédiaire dans la table de destination, certains paramètres de la commande COPY ne s'appliquent pas, comme la fonction de la commande COPY qui lui permet d'activer la compression automatique de la table. Si une compression est nécessaire, ajoutez les détails d'encodage de colonne à l'instruction CREATE TABLE. De plus, dans certains cas, lorsqu'il doit décharger des données du cluster Amazon Redshift et créer des fichiers dans Amazon S3, il s'appuie sur RedshiftCopyActivity UNLOAD l'opération d'Amazon Redshift. Pour améliorer les performances pendant la copie et le déchargement, spécifiez le paramètre PARALLEL OFF à partir de la commande UNLOAD. Pour plus d'informations sur les paramètres, consultez UNLOAD dans le manuel Amazon Redshift Database Developer Guide.	
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence : "dependsOn": {"ref":"myActivityId"}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
input	Nœud de données d'entrée. La source de données peut être Amazon S3, DynamoDB ou Amazon Redshift.	Objet de référence : "input":{ "ref":"myDataNodeId"}

Champs facultatifs	Description	Type d'option
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence : `"onFail": {"ref":"m yActionId"}`
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence : `"onLateAction": {"ref":"myAc tionId"}`
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence : `"onSuccess": {"ref":"myActio nId"}`
output	Nœud de données de sortie. L'emplacement de sortie peut être Amazon S3 ou Amazon Redshift.	Objet de référence : `"output": {"ref":"m yDataNodeId"}`

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence : `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence : `"precondition": {"ref":"myPreconditionId"}`
file d'attente	Correspond au `query_group` paramètre d'Amazon Redshift, qui vous permet d'attribuer et de prioriser les activités simultanées en fonction de leur placement dans les files d'attente. Amazon Redshift limite le nombre de connexions simultanées à 15. Pour plus d'informations, consultez la section Affectation de requêtes à des files d'attente dans le manuel Amazon RDS Database Developer Guide.	Chaîne
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période

Champs facultatifs	Description	Type d'option
retryDelay	Délai entre deux nouvelles tentatives.	Période
scheduleType	Permet de spécifier si la planification s'applique aux objets de votre pipeline. Les valeurs sont : `cron`, `ondemand` et `timeseries` (cron, à la demande et séries chronologiques). La planification `timeseries` signifie que les instances sont programmées à la fin de chaque intervalle. La planification `Cron` signifie que les instances sont programmées au début de chaque intervalle. Une planification `ondemand` vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Pour utiliser des pipelines `ondemand`, vous devez appeler l'opération `ActivatePipeline` pour chaque exécution suivante. Si vous utilisez une planification `ondemand`, vous devez la spécifier dans l'objet par défaut et faire en sorte qu'elle soit le seul `scheduleType` spécifié pour les objets du pipeline.	Énumération

Champs facultatifs	Description	Type d'option
transformSql	Expression `SQL SELECT` utilisée pour transformer les données d'entrée. Exécutez l'expression `transformSql` sur la table nommée `staging`. Lorsque vous copiez des données depuis DynamoDB ou Amazon S3AWS Data Pipeline, vous créez une table appelée « staging » et y chargez initialement les données. Les données de cette table sont utilisées pour mettre à jour la table cible. Le schéma de sortie de `transformSql` doit correspondre au schéma de la table cible finale. Si vous spécifiez l'option `transformSql` , une seconde table intermédiaire est créée à partir de l'instruction SQL spécifiée. Les données de cette seconde table intermédiaire sont ensuite mises à jour dans la table cible finale.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence : `"activeInstances": {"ref":"myRunnableObjectId"}`

Champs liés à l'exécution	Description	Type d'option
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence : `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	Journaux d'étapes EMR disponibles uniquement sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@ healthSta tusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTi me	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCom pletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence : `"waitingOn": {"ref":"myRunnableObjectId"}`

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	Sphère d'un objet. Indique sa situation dans le cycle de vie. Par exemple, les objets de composant produisent des objets d'instance qui exécutent des objets « tentatives ».	Chaîne

ShellCommandActivity

Exécute une commande ou un script. Vous pouvez utiliser `ShellCommandActivity` pour exécuter les tâches planifiées de type séries chronologiques ou de type cron.

Lorsque le `stage` champ est défini sur true et utilisé avec un`S3DataNode`, `ShellCommandActivity` prend en charge le concept de données intermédiaires, ce qui signifie que vous pouvez déplacer des données d'Amazon S3 vers un emplacement d'étape, tel qu'Amazon EC2 ou votre environnement local, travailler sur les données à l'aide de scripts et les `ShellCommandActivity` replacer vers Amazon S3.

Dans ce cas, lorsque votre commande shell est connectée à un `S3DataNode` en entrée, vos scripts shell opèrent directement sur les données avec `${INPUT1_STAGING_DIR}`, `${INPUT2_STAGING_DIR}` et d'autres champs, en faisant référence aux champs `ShellCommandActivity` en entrée.

De même, le résultat de la commande shell peut être transféré dans un répertoire de sortie pour être automatiquement transféré vers Amazon S3, référencé par `${OUTPUT1_STAGING_DIR}${OUTPUT2_STAGING_DIR}`, etc.

Ces expressions peuvent être transmises comme arguments de ligne de commande à la commande shell pour que vous les utilisiez dans la logique de transformation des données.

`ShellCommandActivity` renvoie les chaînes et codes d'erreur Linux. Si une activité `ShellCommandActivity` se traduit par une erreur, la valeur `error` retournée est différente de zéro.

Exemple

Voici un exemple de ce type d'objet.

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

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle `schedule`. Pour définir l'ordre d'exécution des dépendances de cet objet, spécifiez une référence `schedule` à un autre objet. Pour satisfaire cette exigence, définissez explicitement un `schedule` sur l'objet, par exemple, en spécifiant `"schedule": {"ref": "DefaultSchedule"}` . Dans la plupart des cas, il est préférable de placer la référence `schedule` sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Si le pipeline se compose d'une arborescence de planifications (planifications au sein de la planification maître), créez un objet parent ayant une référence de planification. Pour répartir la charge, AWS Data Pipeline crée des objets physiques légèrement plus tôt	Objet de référence , par exemple « schedule » : {"ref » : » mySchedulEld «}

Champs d'invocation de l'objet	Description	Type d'option
	que prévu, mais les exécute conformément à la planification définie. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
command	Commande à exécuter. Utilisez la valeur $ pour référencer les paramètres de positionnement et `scriptArgument` pour spécifier les paramètres de la commande. Cette valeur et les paramètres associés doivent fonctionner dans l'environnement à partir duquel vous lancez l'exécuteur de tâches.	Chaîne
scriptUri	Chemin d'accès par URI Amazon S3 d'un fichier à télécharger et à exécuter en tant que commande shell. Spécifiez un seul `scriptUri`, ou champ `command`. Étant donné que le champ `scriptUri` ne peut pas utiliser de paramètres, utilisez plutôt `command`.	Chaîne

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	La ressource de calcul permettant d'exécuter l'activité ou la commande, par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence, par exemple « RunSon » : {"ref » : » myResourceId «}
workerGroup	Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur `runsOn` et que `workerGroup` existe, `workerGroup` est ignoré.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans la période de départ définie peut être retentée.	Période
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence, par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération

Champs facultatifs	Description	Type d'option
input	Emplacement des données d'entrée.	Objet de référence , par exemple « input » : {"ref » : » myDataNode Id "}
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
output	Emplacement des données de sortie.	Objet de référence , par exemple « output » : {"ref » : » myDataNode Id "}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI Amazon S3, par exemple `'s3://BucketName/Key/'` pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPreconditionId «}
reportProgressTimeout	Délai pour les appels successifs adressés à `reportProgress` par les activités à distance. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et font l'objet d'une nouvelle tentative.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs facultatifs	Description	Type d'option
scheduleType	Permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Les valeurs sont : `cron`, `ondemand` et `timeseries` . Si la planification est définie sur `timeseries` , les instances sont programmées à la fin de chaque intervalle. Si la planification est définie sur `Cron`, les instances sont programmées au début de chaque intervalle. Si la planification est définie sur `ondemand`, vous pouvez exécuter un pipeline une fois, par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification `ondemand`, spécifiez-la dans l'objet par défaut comme seul `scheduleType` pour les objets du pipeline. Pour utiliser des pipelines `ondemand`, vous devez appeler l'opération `ActivatePipeline` pour chaque exécution suivante.	Énumération

Champs facultatifs	Description	Type d'option
scriptArgument	Tableau de chaînes au format JSON à transmettre à la commande spécifiée par le champ command. Par exemple, si la valeur du champ command est `echo $1 $2`, spécifiez `scriptArgument` en tant que `"param1", "param2"`. En cas d'arguments et de paramètres multiples, transmettez le `scriptArgument` comme suit : `"scriptArgument":"arg1","scriptArgument":"param1","scriptArgument":"arg2","scriptArgument":"param2"`. Le `scriptArgument` ne peut être utilisé qu'avec `command` ; son utilisation avec `scriptUri` provoque une erreur.	Chaîne
stage	Détermine si la gestion intermédiaire est activée et permet à vos commandes shell d'avoir accès aux variables de données mises en lots, telles que `${INPUT1_STAGING_DIR}` et `${OUTPUT1_STAGING_DIR}`.	Booléen
stderr	Chemin qui reçoit les messages d'erreur système redirigés à partir de la commande. Si vous utilisez ce `runs0n` champ, il doit s'agir d'un chemin Amazon S3 en raison de la nature transitoire de la ressource exécutant votre activité. Toutefois, si vous spécifiez le champ `workerGroup`, un chemin de fichier local est autorisé.	Chaîne

Champs facultatifs	Description	Type d'option
stdout	Le chemin Amazon S3 qui reçoit la sortie redirigée de la commande. Si vous utilisez ce `runs0n` champ, il doit s'agir d'un chemin Amazon S3 en raison de la nature transitoire de la ressource exécutant votre activité. Toutefois, si vous spécifiez le champ `workerGroup` , un chemin de fichier local est autorisé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	`cancellationReason` si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances à l'origine de l'échec de l'objet.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}

Champs liés à l'exécution	Description	Type d'option
emrStepLog	Les journaux d'étapes Amazon EMR sont disponibles uniquement pour les tentatives d'activité Amazon EMR.	Chaîne
errorId	`errorId` si l'objet a échoué.	Chaîne
errorMessage	`errorMessage` si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Des journaux de tâches Hadoop sont disponibles en cas de tentative d'activités basées sur Amazon EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime

Champs liés à l'exécution	Description	Type d'option
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	Statut de l'objet.	Chaîne
@Version	Version AWS Data Pipeline utilisée pour créer l'objet.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	Emplacement d'un objet dans le cycle de vie. Les objets de composant entraînent des objets d'instance, qui exécutent des objets « tentative s ».	Chaîne

consultez aussi

- CopyActivity
- EmrActivity

SqlActivity

Exécute une requête SQL (script) sur une base de données.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "id" : "MySqlActivity",
  "type" : "SqlActivity",
  "database" : { "ref": "MyDatabaseID" },
  "script" : "SQLQuery" | "scriptUri" : s3://scriptBucket/query.sql,
  "schedule" : { "ref": "MyScheduleID" },
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
database	Base de données sur laquelle exécuter le script SQL fourni.	Objet de référence, par exemple « base de données » : {"ref » : » myDatabas eId «}

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Vous devez	Objet de référence , par exemple

Champs d'invocation de l'objet	Description	Type d'option
	spécifier une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Vous pouvez définir explicitement une planification sur l'objet, par exemple, en spécifiant `"schedule": {"ref": "DefaultSchedule"}` . Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Si le pipeline dispose d'une arborescence de planifications imbriquées dans la planification maître, créez un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	« schedule » : {"ref » : » mySchedulEld «}

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
script	Script SQL à exécuter. Vous devez spécifier script ou scriptUri. Lorsque le script est stocké dans Amazon S3, le script n'est pas évalué en tant qu'expression. Spécifier plusieurs valeurs pour scriptArgument est utile lorsque le script est stocké dans Amazon S3.	Chaîne

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
scriptUri	URI spécifiant l'emplacement d'un script SQL à exécuter dans l'activité.	Chaîne

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
runsOn	Ressource de calcul pour exécuter l'activité ou la commande. Par exemple, une instance Amazon EC2 ou un cluster Amazon EMR.	Objet de référence , par exemple « RunSon » : {"ref » : » myResourc eId «}
workerGroup	Groupe de travail. Utilisé pour les tâches d'acheminement. Si vous fournissez une valeur `runsOn` et que `workerGroup` existe, `workerGroup` est ignoré.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période

Champs facultatifs	Description	Type d'option
dependsOn	Spécifie une dépendance sur un autre objet exécutable.	Objet de référence , par exemple « DependsOn » : {"ref » : » myActivityId «}
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
input	Emplacement des données d'entrée.	Objet de référence , par exemple « input » : {"ref » : » myDataNode Id "}
lateAfterTimeout	Période depuis le début planifié du pipeline au sein de laquelle l'objet exécuté doit démarrer.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions qui doivent être déclenchées si un objet n'a pas encore été planifié ou n'est toujours pas terminé au cours de la période écoulée depuis le début prévu du pipeline, comme spécifié par « lateAfterTimeout ».	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
output	Emplacement des données de sortie. Cela n'est utile que pour le référencement depuis un script (par exemple`#{output.tablename}`) et pour créer la table de sortie en définissant « createTableSql » dans le nœud de données de sortie. La sortie de la requête SQL n'est pas écrite dans le nœud des données de sortie.	Objet de référence , par exemple « output » : {"ref » : » myDataNode Id "}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
pipelineLogUri	L'URI S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
precondition	Définit une condition préalable facultative. Un nœud de données n'est pas marqué « READY » tant que toutes les conditions préalables ne sont pas remplies.	Objet de référence , par exemple « précondition » : {"ref » : » myPrecond itionId «}

Champs facultatifs	Description	Type d'option
file d’attente	[Amazon Redshift uniquement] Correspond au paramètre query_group d'Amazon Redshift, qui vous permet d'attribuer et de hiérarchiser les activités simultanées en fonction de leur placement dans les files d'attente. Amazon Redshift limite le nombre de connexions simultanées à 15. Pour plus d'informations, consultez Attribution de requêtes aux files d'attente dans le manuel Amazon Redshift Developer Guide.	Chaîne
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Les valeurs sont : `cron`, `ondemand` et `timeseries` (cron, à la demande et séries chronologiques). Une planification `timeseries` signifie que les instances sont programmées à la fin de chaque intervalle. Une planification `cron` signifie que les instances sont programmées au début de chaque intervalle. Une planification `ondemand` vous permet d'exécuter un pipeline une fois par activation. Cela signifie que vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification `ondemand`, elle doit être spécifiée dans l'objet par défaut et être le seul `scheduleType` spécifié pour les objets du pipeline. Pour utiliser des pipelines `ondemand`, vous devez appeler l'opération `ActivatePipeline` pour chaque exécution suivante.	Énumération

Champs facultatifs	Description	Type d'option
scriptArgument	Liste de variables pour le script. Vous pouvez également placer directement des expressio ns dans le champ script. Spécifier plusieurs valeurs pour scriptArgument est utile lorsque le script est stocké dans Amazon S3. Exemple : # {format (@scheduledStartTime, « YY-MM-DD HH:MM:SS"}\n# {format (PlusPeriod (@, « 1 jour »)scheduledStartTime, « YY-MM-DD HH:MM:SS"}	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}

Champs liés à l'exécution	Description	Type d'option
emrStepLog	Journaux d'étapes EMR disponibles uniquement sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution.	DateTime
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime

Champs liés à l'exécution	Description	Type d'option
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

Ressources

Les objets suivants sont les objets de ressource AWS Data Pipeline :

Objets

- Ec2Resource
- EmrCluster
- HttpProxy

Ec2Resource

Instance Amazon EC2 qui exécute le travail défini par une activité de pipeline.

AWS Data Pipelineprend désormais en charge IMDSv2 pour l'instance Amazon EC2, qui utilise une méthode orientée session pour mieux gérer l'authentification lors de la récupération des informations de métadonnées à partir des instances. Une session commence et termine une série de demandes utilisées par le logiciel exécuté sur une instance Amazon EC2 pour accéder aux métadonnées et aux informations d'identification de l'instance Amazon EC2 stockées localement. Le logiciel démarre une session par une simple requête HTTP PUT adressée à IMDSv2. IMDSv2 renvoie un jeton secret au logiciel exécuté sur l'instance Amazon EC2, qui utilisera le jeton comme mot de passe pour demander à IMDSv2 des métadonnées et des informations d'identification.

> **Note**
>
> Pour utiliser IMDSv2 pour votre instance Amazon EC2, vous devez modifier les paramètres, car l'AMI par défaut n'est pas compatible avec IMDSv2. Vous pouvez spécifier une nouvelle version d'AMI que vous pouvez récupérer via le paramètre SSM suivant : `/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs`

Pour plus d'informations sur les instances Amazon EC2 par défaut AWS Data Pipeline créées si vous ne spécifiez aucune instance, consultez. Instances Amazon EC2 par défaut par région AWS

Exemples

EC2-Classic

> **Important**
>
> Seuls AWS les comptes créés avant le 4 décembre 2013 sont compatibles avec la plateforme EC2-Classic. Si vous possédez l'un de ces comptes, vous pouvez avoir la possibilité de créer

des objets EC2Resource pour un pipeline dans un réseau EC2-Classic plutôt qu'un VPC. Nous vous recommandons vivement de créer des ressources pour tous vos pipelines dans des VPC. En outre, si vous disposez de ressources existantes dans EC2-Classic, nous vous recommandons de les migrer vers un VPC.

L'exemple d'objet suivant lance une instance EC2 dans EC2-Classic, avec certains champs facultatifs définis.

```
{
  "id" : "MyEC2Resource",
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroups" : [
    "test-group",
    "default"
  ],
  "keyPair" : "my-key-pair"
}
```

EC2-VPC

L'exemple d'objet suivant objet lance une instance EC2 dans un VPC personnalisé, avec quelques champs facultatifs.

```
{
  "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"
```

```
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
resourceRole	Rôle IAM qui contrôle les ressources auxquelles l'instance Amazon EC2 peut accéder.	Chaîne
rôle	Rôle IAM AWS Data Pipeline utilisé pour créer l'instance EC2.	Chaîne

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécution d'un intervalle de planification. Pour définir l'ordre d'exécution des dépendances de cet objet, spécifiez une référence de planification à un autre objet. Vous pouvez effectuer cette opération de différentes manières : • Pour vous assurer que tous les objets dans le pipeline héritent de la planification, définissez explicitement une planification sur l'objet : `"schedule": {"ref": "DefaultSchedule"}` . Dans la plupart des cas, il s'avère utile de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. • Si le pipeline dispose d'une arborescence de planifications imbriquées dans la planification maître, vous pouvez créer un objet	Objet de référence, par exemple, `"schedule":{"ref": "mySchedule Id"}`

Champs d'invocation de l'objet	Description	Type d'option
	parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Champs facultatifs	Description	Type d'option
actionOnResourceDéfaillance	Action effectuée après une défaillance de ressource pour cette ressource. Les valeurs valides sont `"retryall"` et `"retrynone"` .	Chaîne
actionOnTaskDéfaillance	Action effectuée après l'échec d'une tâche pour cette ressource. Les valeurs valides sont `"continue"` ou `"terminate"` .	Chaîne
associatePublicIpAdresse	Indique si vous souhaitez attribuer une adresse IP publique à l'instance. Si l'instance se trouve dans Amazon EC2 ou Amazon VPC, la valeur par défaut est. `true` Sinon, la valeur par défaut est `false`.	Booléen
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans la période de départ définie peut être retentée.	Période
availabilityZone	Zone de disponibilité dans laquelle lancer l'instance Amazon EC2.	Chaîne

Champs facultatifs	Description	Type d'option
Désactiver IMDS V1	La valeur par défaut est false et active à la fois IMDSv1 et IMDSv2. Si vous le définissez sur true, il désactive IMDSv1 et ne fournit que IMDSv2.	Booléen
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
httpProxy	Hôte proxy que les clients utilisent pour se connecter aux services AWS.	Objet de référence , par exemple, `"httpProxy": {"ref":"myHttpProxyId"}`
imageId	ID de l'AMI à utiliser pour l'instance. Par défaut, AWS Data Pipeline utilise le type de virtualisation de l'AMI HVM. Les ID d'AMI spécifiques utilisés sont basés sur la région. Vous pouvez remplacer l'AMI par défaut en spécifiant l'AMI HVM de votre choix. Pour plus d'informations sur les types d'AMI, consultez les sections Types de virtualisation d'AMI Linux et Trouver une AMI Linux dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.	Chaîne
initTimeout	Délai d'attente pour le démarrage de la ressource.	Période
instanceCount	Obsolète.	Entier
instanceType	Type d'instance Amazon EC2 à démarrer.	Chaîne
keyPair	Nom de la paire de clés. Si vous lancez une instance Amazon EC2 sans spécifier de paire de clés, vous ne pouvez pas vous y connecter.	Chaîne

Champs facultatifs	Description	Type d'option
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maxActiveInstances	Nombre maximal d'instances actives simultané es d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
minInstanceCount	Obsolète.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple, `"onFail": {"ref":"m yActionId"}`
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou est toujours en cours d'exécuti on.	Objet de référence , par exemple, `"onLateAction": {"ref":"myAc tionId"}`
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple, `"onSuccess": {"ref":"myActio nId"}`

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple, `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	L'URI Amazon S3 (par exemple `'s3://BucketName/Key/'`) pour le téléchargement des journaux pour le pipeline.	Chaîne
region	Code de la région dans laquelle l'instance Amazon EC2 doit s'exécuter. Par défaut, l'instance s'exécute dans la même région que le pipeline. Vous pouvez exécuter l'instance dans la même région qu'un ensemble de données dépendantes.	Énumération
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et feront l'objet d'une nouvelle tentative.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période
runAsUser	L'utilisateur qui doit exécuter le TaskRunner.	Chaîne
runsOn	Ce champ n'est pas autorisé sur cet objet.	Objet de référence , par exemple, `"runsOn": {"ref":"myResourceId"}`

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin d'un intervalle, ou à la demande. Les valeurs sont les suivantes : • `timeseries` . Les instances sont planifiées à la fin de chaque intervalle. • `cron`. Les instances sont planifiées au début de chaque intervalle. • `ondemand`. Vous permet d'exécuter un pipeline une fois par activation. Vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification à la demande, elle doit être spécifiée dans l'objet par défaut et être le seul `scheduleType` pour les objets du pipeline. Pour utiliser des pipelines à la demande, vous devez appeler l'opérati on `ActivatePipeline` pour chaque exécution suivante.	Énumération
securityGroupIds	Les identifiants d'un ou de plusieurs groupes de sécurité Amazon EC2 à utiliser pour les instances du pool de ressources.	Chaîne
securityGroups	Un ou plusieurs groupes de sécurité Amazon EC2 à utiliser pour les instances du pool de ressources.	Chaîne
spotBidPrice	Montant maximum par heure pour votre instance Spot en dollars, qui est une valeur décimale comprise entre 0 et 20,00 (non incluse).	Chaîne

Champs facultatifs	Description	Type d'option
subnetId	ID du sous-réseau Amazon EC2 dans lequel démarrer l'instance.	Chaîne
terminateAfter	Nombre d'heures après lequel résilier la ressource.	Période
useOnDemandOnLastAttempt	Lors de la dernière tentative de demande d'une instance Spot, effectuez une demande pour les instances à la demande au lieu d'une instance Spot. Cela garantit que si toutes les tentatives précédentes ont échoué, la dernière tentative n'est pas interrompue.	Booléen
workerGroup	Ce champ n'est pas autorisé sur cet objet.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence, par exemple, `"activeInstances":{"ref":"myRunnableObjectId"}`
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	`cancellationReason` si l'objet a été annulé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple, `"cascadeFailedOn": {"ref":"myRunnableObjectId"}`
emrStepLog	Les journaux d'étapes ne sont disponibles que pour les tentatives d'activité sur Amazon EMR.	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	Message d'erreur si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@failureReason	Raison de l'échec de la ressource.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Des journaux de tâches Hadoop sont disponibles en cas de tentative d'activité sur Amazon EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime

Champs liés à l'exécution	Description	Type d'option
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence, par exemple, `"waitingOn": {"ref":"myRunnableObjectId"}`

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne

Champs système	Description	Type d'option
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	Emplacement d'un objet dans le cycle de vie. Les objets de composant entraînent des objets d'instance, qui exécutent des objets « tentative s ».	Chaîne

EmrCluster

Représente la configuration d'un cluster Amazon EMR. Cet objet est utilisé par EmrActivity et HadoopActivity pour lancer un cluster.

Table des matières

Schedulers

Les planificateurs fournissent un moyen de spécifier l'allocation des ressources et de définir les priorités de travail au sein d'un cluster Hadoop. Les administrateurs ou les utilisateurs peuvent choisir un planificateur pour différentes classes d'utilisateurs et d'applications. Un planificateur peut utiliser les files d'attente pour allouer des ressources aux utilisateurs et aux applications. Vous configurez ces files d'attente lorsque vous créez le cluster. Vous pouvez ensuite configurer la priorité de certains types de travail et d'utilisateur par rapport à d'autres. Vous bénéficiez ainsi d'une utilisation efficace des ressources du cluster, tout en permettant à plus d'un utilisateur de soumettre des tâches au cluster. Il existe trois types de planificateur disponibles :

- FairScheduler— Tente de planifier les ressources de manière uniforme sur une longue période.

- CapacityScheduler— Utilise des files d'attente pour permettre aux administrateurs de clusters d'affecter les utilisateurs à des files d'attente dont la priorité et l'allocation des ressources varient.
- Par défaut : utilisé par le cluster, ce qui peut être configuré par votre site.

Versions publiées par Amazon EMR

Une version Amazon EMR est un ensemble d'applications open-source issues de l'écosystème big data. Chaque version comprend différentes applications, composants et fonctionnalités Big Data que vous sélectionnez pour qu'Amazon EMR installe et configure lorsque vous créez un cluster. Vous spécifiez la version à l'aide de l'étiquette de version. Les étiquettes de version sont sous la forme `emr-x.x.x`. Par exemple, `emr-5.30.0`. Les clusters Amazon EMR sont basés sur l'étiquette de version `emr-4.0.0` et utilisent ultérieurement cette `releaseLabel` propriété pour spécifier l'étiquette de version d'un `EmrCluster` objet. Les versions antérieures utilisent la propriété `amiVersion`.

> **Important**
>
> Tous les clusters Amazon EMR créés à l'aide de la version 5.22.0 ou ultérieure utilisent Signature version 4 pour authentifier les demandes adressées à Amazon S3. Certaines versions antérieures utilisent Signature Version 2. La prise en charge de Signature Version 2 est interrompue. Pour de plus amples informations, veuillez consulter Mise à jour Amazon S3 — Période d'obsolescence SigV2 étendue et modifiée. Nous vous recommandons vivement d'utiliser une version d'Amazon EMR compatible avec Signature Version 4. Pour les versions antérieures, à commencer par EMR 4.7.x, la version la plus récente de la série a été mise à jour pour prendre en charge Signature Version 4. Lorsque vous utilisez une version EMR antérieure, nous vous recommandons d'utiliser la dernière version de la série. En outre, évitez les versions antérieures à EMR 4.7.0.

Considérations et restrictions

Utilisez la dernière version de Task Runner

Si vous utilisez un `EmrCluster` objet autogéré doté d'une étiquette de version, utilisez le dernier Task Runner. Pour plus d'informations sur Task Runner, consultez Utilisation de Task Runner. Vous pouvez configurer les valeurs des propriétés pour toutes les classifications de configuration Amazon EMR. Pour plus d'informations, consultez la section Configuration des applications dans le guide de

mise à jour d'Amazon EMRthe section called “EmrConfiguration”, le et les références aux the section called “Propriété” objets.

Support pour IMDSv2

Auparavant, seul AWS Data Pipeline IMDSv1 était pris en charge. Désormais, il est AWS Data Pipeline compatible avec IMDSv2 dans Amazon EMR 5.23.1, 5.27.1 et 5.32 ou version ultérieure, et Amazon EMR 6.2 ou version ultérieure. IMDSv2 utilise une méthode orientée session pour mieux gérer l'authentification lors de la récupération d'informations de métadonnées à partir d'instances. Vous devez configurer vos instances pour effectuer des appels IMDSv2 en créant des ressources gérées par l'utilisateur à l'aide de -2.0. TaskRunner

Amazon EMR 5.32 ou version ultérieure et Amazon EMR 6.x

Les séries Amazon EMR 5.32 ou versions ultérieures et 6.x utilisent la version 3.x de Hadoop, qui a apporté des modifications majeures à la façon dont le chemin de classe de Hadoop est évalué par rapport à la version 2.x de Hadoop. Les bibliothèques courantes telles que Joda-Time ont été supprimées du classpath.

Si EmrActivity ou HadoopActivity exécute un fichier Jar qui dépend d'une bibliothèque supprimée dans Hadoop 3.x, l'étape échoue avec l'erreur ou. `java.lang.NoClassDefFoundError` `java.lang.ClassNotFoundException` Cela peut se produire pour les fichiers Jar qui s'exécutent sans problème avec les versions 5.x d'Amazon EMR.

Pour résoudre le problème, vous devez copier les dépendances du fichier Jar dans le chemin de classe Hadoop d'un `EmrCluster` objet avant de démarrer le ou le. `EmrActivity` `HadoopActivity` Pour ce faire, nous fournissons un script bash. Le script bash est disponible à l'emplacement suivant, où se *`MyRegion`*trouve la AWS région dans laquelle votre `EmrCluster` objet s'exécute, par exemple`us-west-2`.

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

Le mode d'exécution du script varie selon qu'`EmrActivity`il `HadoopActivity` s'exécute sur une ressource gérée par AWS Data Pipeline ou sur une ressource autogérée.

Si vous utilisez une ressource gérée parAWS Data Pipeline, ajoutez un `bootstrapAction` à l'`EmrCluster`objet. `bootstrapAction`Spécifie le script et les fichiers Jar à copier en tant qu'arguments. Vous pouvez ajouter jusqu'à 255 `bootstrapAction` champs par `EmrCluster` objet,

et vous pouvez ajouter un `bootstrapAction` champ à un `EmrCluster` objet qui possède déjà des actions d'amorçage.

Pour spécifier ce script en tant qu'action d'amorçage, utilisez la syntaxe suivante, où se `JarFileRegion` trouve la région dans laquelle le fichier Jar est enregistré, et chaque *MyJarFilen* est le chemin absolu dans Amazon S3 d'un fichier Jar à copier dans le classpath Hadoop. Ne spécifiez pas les fichiers Jar qui se trouvent dans le chemin de classe Hadoop par défaut.

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-
hadoop-classpath.sh,JarFileRegion,MyJarFile1,MyJarFile2[, ...]
```

L'exemple suivant spécifie une action bootstrap qui copie deux fichiers Jar dans Amazon S3 : `my-jar-file.jar` et le`emr-dynamodb-tool-4.14.0-jar-with-dependencies.jar`. La région utilisée dans cet exemple est 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"]
}
```

Vous devez enregistrer et activer le pipeline pour que la modification apportée `bootstrapAction` au nouveau soit prise en compte.

Si vous utilisez une ressource autogérée, vous pouvez télécharger le script sur l'instance de cluster et l'exécuter depuis la ligne de commande à l'aide de SSH. Le script crée un répertoire nommé `/etc/hadoop/conf/shellprofile.d` et un fichier nommé `datapipeline-jars.sh` dans ce répertoire. Les fichiers jar fournis en tant qu'arguments de ligne de commande sont copiés dans un répertoire nommé créé par le script. `/home/hadoop/datapipeline_jars` Si votre cluster est configuré différemment, modifiez le script de manière appropriée après l'avoir téléchargé.

La syntaxe d'exécution du script sur la ligne de commande est légèrement différente de celle `bootstrapAction` utilisée dans l'exemple précédent. Utilisez des espaces plutôt que des virgules entre les arguments, comme indiqué dans l'exemple suivant.

```
./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
```

Autorisations Amazon EMR

Lorsque vous créez un rôle IAM personnalisé, considérez attentivement les autorisations minimales nécessaires pour que votre cluster effectue son travail. Assurez-vous d'accorder l'accès aux ressources requises, telles que les fichiers dans Amazon S3 ou les données dans Amazon RDS, Amazon Redshift ou DynamoDB. Si vous souhaitez définir `visibleToAllUsers` avec la valeur False, votre rôle doit avoir les autorisations appropriées pour le faire. Notez que `DataPipelineDefaultRole` ne dispose pas de ces autorisations. Vous devez soit fournir une union des `DataPipelineDefaultRole` rôles `DefaultDataPipelineResourceRole` et en tant que rôle `EmrCluster` objet, soit créer votre propre rôle à cette fin.

Syntaxe

Champs d'invocation de l'objet	Description	Type d'option
schedule	Cet objet est appelé dans le cadre de l'exécuti on d'un intervalle de planification. Spécifiez une référence de planification à un autre objet pour définir l'ordre d'exécution des dépendances de l'objet. Vous pouvez répondre à cette exigence en définissant explicitement une planifica tion sur l'objet, par exemple, en spécifian t `"schedule": {"ref": "DefaultS chedule"}` . Dans la plupart des cas, il est préférable de placer la planification de référence sur l'objet de pipeline par défaut de manière à ce que tous les objets héritent cette planification. Ou, si le pipeline dispose d'une	Objet de référence , par exemple, `"schedule ":{"ref": "mySchedu leId"}`

Champs d'invocation de l'objet	Description	Type d'option
	arborescence de planifications (planifications au sein de la planification maître), vous pouvez créer un objet parent ayant une référence de planification. Pour plus d'informations sur les exemples de configurations de planification facultatives, consultez https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html.	

Champs facultatifs	Description	Type d'option
actionOnResourceDéfaillance	Action effectuée après une défaillance de ressource pour cette ressource. Les valeurs valides sont « `retryall` », valeur qui retente toutes les tâches sur le cluster pendant la durée spécifiée, et « `retrynone` ».	Chaîne
actionOnTaskDéfaillance	Action effectuée après l'échec d'une tâche pour cette ressource. Les valeurs valides sont « continue », qui signifie de ne pas mettre fin au cluster, et « terminate ».	Chaîne
additionalMasterSecurityGroupIds	Identifiant des groupes de sécurité maître supplémentaires du cluster EMR, sous la forme sg-01XXXX6a. Pour plus d'informations, consultez la section Groupes de sécurité supplémentaires Amazon EMR dans le guide de gestion Amazon EMR.	Chaîne
additionalSlaveSecurityGroupIds	Identifiant des groupes de sécurité esclave supplémentaires du cluster EMR, sous la forme `sg-01XXXX6a` .	Chaîne

Champs facultatifs	Description	Type d'option
amiVersion	Version Amazon Machine Image (AMI) utilisée par Amazon EMR pour installer les nœuds du cluster. Pour de plus amples informations, veuillez consulter le Amazon EMR Management Guide.	Chaîne
applications	Applications à installer dans le cluster avec les arguments séparés par des virgules. Par défaut Hive et Pig sont installés. Ce paramètre s'applique uniquement à Amazon EMR version 4.0 et ultérieure.	Chaîne
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
availabilityZone	Zone de disponibilité dans laquelle exécuter le cluster.	Chaîne
bootstrapAction	Action à exécuter lorsque le cluster démarre. Vous pouvez spécifier des arguments séparés par des virgules. Pour spécifier plusieurs actions (jusqu'à 255), ajoutez plusieurs champs `bootstrapAction` . Le comportement par défaut consiste à lancer le cluster sans actions d'amorçage.	Chaîne

Champs facultatifs	Description	Type d'option
configuration	Configuration pour le cluster Amazon EMR. Ce paramètre s'applique uniquement à Amazon EMR version 4.0 et ultérieure.	Objet de référence , par exemple, `"configuration":{"ref":"myEmrConfigurationId"}`
coreInstanceBidPrix	Le prix spot maximum que vous êtes prêt à payer pour les instances Amazon EC2. Si le prix de l'offre est spécifié, Amazon EMR utilise les instances spot pour le groupe d'instances. Spécifié en USD.	Chaîne
coreInstanceCount	Nombre de nœuds principaux à utiliser pour le cluster.	Entier
coreInstanceType	Type d'instance Amazon EC2 à utiliser pour les nœuds principaux. Consultez Instances Amazon EC2 prises en charge pour les clusters Amazon EMR .	Chaîne
coreGroupConfiguration	Configuration du groupe d'instances principal du cluster Amazon EMR. Ce paramètre s'applique uniquement à Amazon EMR version 4.0 et ultérieure.	Objet de référence , par exemple, `“configuration”: {“ref”: “myEmrConfigurationId”}`

Champs facultatifs	Description	Type d'option
coreEbsConfiguration	Configuration des volumes Amazon EBS qui seront attachés à chacun des nœuds principaux du groupe principal du cluster Amazon EMR. Pour plus d'informations, consultez la section Types d'instances qui supportent l'optimisation EBS dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.	Objet de référence, par exemple, `“coreEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
customAmiId	S'applique uniquement aux versions 5.7.0 et ultérieures d'Amazon EMR. Spécifie l'ID AMI d'une AMI personnalisée à utiliser lorsqu'Amazon EMR approvisionne des instances Amazon EC2. Il peut également être utilisé à la place des actions bootstrap pour personnaliser les configurations des nœuds du cluster. Pour plus d'informations, consultez la rubrique suivante dans le guide de gestion Amazon EMR. Utilisation d'une AMI personnalisée	Chaîne

Champs facultatifs	Description	Type d'option
EbsBlockDeviceConfig	Configuration d'un périphérique de bloc Amazon EBS demandé associé au groupe d'instances. Inclut un nombre spécifié de volumes qui seront associés à chaque instance du groupe d'instances. Inclut `volumesPerInstance` et `volumeSpecification`, où : • `volumesPerInstance` représente le nombre de volumes EBS avec une configuration de volume spécifique qui seront associés à chaque instance du groupe d'instances. • `volumeSpecification` correspond aux spécifications du volume Amazon EBS, telles que le type de volume, les IOPS et la taille en gigaoctets (GiB), qui seront demandées pour le volume EBS attaché à une instance EC2 dans le cluster Amazon EMR.	Objet de référence, par exemple, `“EbsBlockDeviceConfig”: {“ref”: “myEbsBlockDeviceConfig”}`
emrManagedMasterSecurityGroupId	Identifiant du groupe de sécurité principal du cluster Amazon EMR, qui prend la forme de. `sg-01XXXX6a` Pour plus d'informations, consultez Configurer les groupes de sécurité dans le guide de gestion Amazon EMR.	Chaîne
emrManagedSlaveSecurityGroupId	L'identifiant du groupe de sécurité esclave du cluster Amazon EMR, qui suit le formulaire. `sg-01XXXX6a`	Chaîne
enableDebugging	Active le débogage sur le cluster Amazon EMR.	Chaîne
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération

Champs facultatifs	Description	Type d'option
hadoopSchedulerType	Type de planificateur du cluster. Les types valides sont : `PARALLEL_FAIR_SCHEDULING` , `PARALLEL_CAPACITY_SCHEDULING` et `DEFAULT_SCHEDULER` .	Énumération
httpProxy	Hôte proxy que les clients utilisent pour se connecter aux services AWS.	Objet de référence , par exemple, « HttpProxy » : {"ref » : » myHttpProxy Id "}
initTimeout	Délai d'attente pour le démarrage de la ressource.	Période
keyPair	La paire de clés Amazon EC2 à utiliser pour se connecter au nœud principal du cluster Amazon EMR.	Chaîne
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini sur`ondemand`.	Période
masterInstanceBidPrix	Le prix spot maximum que vous êtes prêt à payer pour les instances Amazon EC2. Valeur décimale comprise entre 0 et 20,00 (exclu). Spécifié en USD. La définition de cette valeur autorise les instances Spot pour le nœud maître du cluster Amazon EMR. Si le prix de l'offre est spécifié, Amazon EMR utilise les instances spot pour le groupe d'instances.	Chaîne
masterInstanceType	Type d'instance Amazon EC2 à utiliser pour le nœud maître. Consultez Instances Amazon EC2 prises en charge pour les clusters Amazon EMR .	Chaîne

Champs facultatifs	Description	Type d'option
masterGroupConfiguration	Configuration du groupe d'instances principal du cluster Amazon EMR. Ce paramètre s'applique uniquement à Amazon EMR version 4.0 et ultérieure.	Objet de référence, par exemple, `“configuration”: {“ref”: “myEmrConfigurationId”}`
masterEbsConfiguration	Configuration des volumes Amazon EBS qui seront attachés à chacun des nœuds principaux du groupe maître du cluster Amazon EMR. Pour plus d'informations, consultez la section Types d'instances qui supportent l'optimisation EBS dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.	Objet de référence, par exemple, `“masterEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
maxActiveInstances	Nombre maximal d'instances actives simultanées d'un composant. Les réexécutions ne sont pas comptabilisées dans le nombre d'instances actives.	Entier
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence, par exemple, `"onFail": {"ref":"myActionId"}`
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence, par exemple, `"onLateAction": {"ref":"myActionId"}`

Champs facultatifs	Description	Type d'option
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple, `"onSuccess": {"ref":"myActionId"}`
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple, `"parent": {"ref":"myBaseObjectId"}`
pipelineLogUri	L'URI Amazon S3 (tel que 's3 ://BucketName/Key/ ') pour le téléchargement des journaux pour le pipeline.	Chaîne
region	Code de la région dans laquelle le cluster Amazon EMR doit s'exécuter. Par défaut, le cluster s'exécute dans la même région que le pipeline. Vous pouvez exécuter le cluster dans la même région qu'un ensemble de données dépendantes.	Énumération
releaseLabel	Étiquette de publication pour le cluster EMR.	Chaîne
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
resourceRole	Rôle IAM AWS Data Pipeline utilisé pour créer le cluster Amazon EMR. Le rôle par défaut est `DataPipelineDefaultRole` .	Chaîne

Champs facultatifs	Description	Type d'option
retryDelay	Délai entre deux nouvelles tentatives.	Période
rôle	Le rôle IAM a été transmis à Amazon EMR pour créer des nœuds EC2.	Chaîne
runsOn	Ce champ n'est pas autorisé sur cet objet.	Objet de référence , par exemple, `"runsOn": {"ref":"myResourceId"}`
Configuration de sécurité	Identifiant de la configuration de sécurité EMR qui sera appliquée au cluster. Ce paramètre s'applique uniquement aux versions 4.8.0 et ultérieures d'Amazon EMR.	Chaîne
serviceAccessSecurityGroupId	Identifiant du groupe de sécurité d'accès aux services du cluster Amazon EMR.	String. Suit le format `sg-01XXXX6a` , par exemple, `sg-1234abcd` .

Champs facultatifs	Description	Type d'option
scheduleType	Le type de planification vous permet de spécifier si les objets de votre définition de pipeline doivent être planifiés au début ou à la fin de l'intervalle. Les valeurs sont : `cron`, `ondemand` et `timeseries` (cron, à la demande et séries chronologiques). La planification `timeseries` signifie que les instances sont programmées à la fin de chaque intervalle. La planification `cron` signifie que les instances sont programmées au début de chaque intervalle. Une planification `ondemand` vous permet d'exécuter un pipeline une fois par activation. Vous n'avez pas à cloner ou à recréer le pipeline pour l'exécuter à nouveau. Si vous utilisez une planification `ondemand`, elle doit être spécifiée dans l'objet par défaut et être le seul `scheduleType` spécifié pour les objets du pipeline. Pour utiliser des pipelines `ondemand`, vous devez appeler l'opération `ActivatePipeline` pour chaque exécution suivante.	Énumération
subnetId	Identifiant du sous-réseau dans lequel lancer le cluster Amazon EMR.	Chaîne
supportedProducts	Paramètre qui installe un logiciel tiers sur un cluster Amazon EMR, par exemple une distribut ion tierce de Hadoop.	Chaîne
taskInstanceBidPrix	Prix spot maximum que vous êtes disposé à payer pour les instances EC2. Valeur décimale comprise entre 0 et 20,00 (exclu). Spécifié en USD. Si le prix de l'offre est spécifié, Amazon EMR utilise les instances spot pour le groupe d'instances.	Chaîne

Champs facultatifs	Description	Type d'option
taskInstanceCount	Le nombre de nœuds de tâches à utiliser pour le cluster Amazon EMR.	Entier
taskInstanceType	Type d'instance Amazon EC2 à utiliser pour les nœuds de tâches.	Chaîne
taskGroupConfiguration	Configuration du groupe d'instances de tâches du cluster Amazon EMR. Ce paramètre s'applique uniquement à Amazon EMR version 4.0 et ultérieure.	Objet de référence, par exemple, `“configuration”: {“ref”: “myEmrConfigurationId”}`
taskEbsConfiguration	Configuration des volumes Amazon EBS qui seront attachés à chacun des nœuds de tâches du groupe de tâches du cluster Amazon EMR. Pour plus d'informations, consultez la section Types d'instances qui supportent l'optimisation EBS dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.	Objet de référence, par exemple, `“taskEbsConfiguration”: {“ref”: “myEbsConfiguration”}`
terminateAfter	Résiliez la ressource à l'issue de ce nombre d'heures.	Entier

Champs facultatifs	Description	Type d'option
VolumeSpecification	Les spécifications du volume Amazon EBS, telles que le type de volume, les IOPS et la taille en gigaoctets (GiB), qui seront demandées pour le volume Amazon EBS attaché à une instance Amazon EC2 dans le cluster Amazon EMR. Le nœud peut être un nœud principal, maître ou de tâche. `VolumeSpecification` inclut les éléments suivants : • `iops()` Integer. Le nombre d'opérations d'E/S par seconde (IOPS) prises en charge par le volume Amazon EBS, par exemple, 1 000. Pour plus d'informations, consultez la section Caractéristiques des E/S EBS dans le guide de l'utilisateur Amazon EC2 pour les instances Linux. • `sizeinGB()` . Nombre entier. La taille du volume Amazon EBS, en gibioctets (GiB), par exemple 500. Pour plus d'informations sur les combinaisons valides de types de volumes et de tailles de disque dur, consultez la section Types de volumes EBS dans le guide de l'utilisateur Amazon EC2 pour les instances Linux. • `volumetType` . Corde. Le type de volume Amazon EBS, par exemple gp2. Les types de volumes pris en charge incluent les types standard, gp2, io1, st1, sc1, etc. Pour plus d'informations, consultez la section Types de volumes EBS dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.	Objet de référence , par exemple, `“VolumeSpecification”: {“ref”: “myVolumeSpecification”}`

Champs facultatifs	Description	Type d'option
useOnDemandOnLastAttempt	Lors de la dernière tentative de demande d'une ressource, effectuez une demande d'instances à la demande, plutôt que d'instances Spot. Cela garantit que si toutes les tentatives précédentes ont échoué, la dernière tentative n'est pas interrompue.	Booléen
workerGroup	Champ non autorisé sur cet objet.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence, par exemple, « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence, par exemple, "cascadeFailedOn« : {" ref » : » myRunnableObject Id "}

Champs liés à l'exécution	Description	Type d'option
emrStepLog	Les journaux d'étapes sont disponibles uniquement pour les tentatives d'activité Amazon EMR.	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	Message d'erreur si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
@failureReason	Raison de l'échec de la ressource.	Chaîne
@finishedTime	Heure à laquelle l'objet a terminé son exécution .	DateTime
hadoopJobLog	Des journaux de tâches Hadoop sont disponibles en cas de tentative d'activité sur Amazon EMR.	Chaîne
@healthStatus	État de santé de l'objet qui reflète la réussite ou l'échec de la dernière instance qui a atteint un état résilié.	Chaîne
@healthStatusFromInstanceId	ID du dernier objet d'instance qui atteint un état résilié.	Chaîne
@ healthStatusUpdated Heure	Heure à laquelle l'état de santé a été mis à jour pour la dernière fois.	DateTime
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
@lastDeactivatedTime	Heure à laquelle l'objet a été désactivé pour la dernière fois.	DateTime
@ latestCompletedRun Heure	Heure de la dernière exécution pour laquelle l'exécution s'est terminée.	DateTime

Champs liés à l'exécution	Description	Type d'option
@latestRunTime	Heure de la dernière exécution pour laquelle l'exécution a été planifiée.	DateTime
@nextRunTime	Prochaine heure d'exécution planifiée.	DateTime
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple, « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	Emplacement d'un objet dans le cycle de vie. Les objets de composant entraînent des objets d'instance, qui exécutent des objets « tentative s ».	Chaîne

Exemples

Les exemples suivants sont des exemples de ce type d'objet.

Table des matières

Lancez un cluster Amazon EMR avec HadoopVersion

Example

L'exemple suivant lance un cluster Amazon EMR à l'aide de la version 1.0 de l'AMI et de 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"]
}
```

Lancez un cluster Amazon EMR avec le label de version emr-4.x ou supérieur

Example

L'exemple suivant lance un cluster Amazon EMR à l'aide du nouveau champ : `releaseLabel`

```
{
  "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"}
}
```

Installez des logiciels supplémentaires sur votre cluster Amazon EMR

Example

`EmrCluster`fournit le `supportedProducts` champ qui installe un logiciel tiers sur un cluster Amazon EMR. Par exemple, il vous permet d'installer une distribution personnalisée de Hadoop, telle que MapR. Il accepte une liste d'arguments séparés par des virgules pour que le logiciel tiers puisse lire et agir en conséquence. L'exemple suivant montre comment utiliser le champ `supportedProducts` d'`EmrCluster` pour créer un cluster MapR M3 personnalisé avec la suite Karmasphere Analytics installée et y exécuter un objet `EmrActivity`.

```
{
    "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"
}
```

Désactiver le chiffrement côté serveur sur les versions 3.x

Example

Une activité `EmrCluster` avec Hadoop version 2.x créé par AWS Data Pipeline permet un chiffrement côté serveur par défaut. Si vous souhaitez désactiver le chiffrement côté serveur, vous devez spécifier une action de démarrage dans la définition de l'objet cluster.

L'exemple suivant crée une activité `EmrCluster` avec le chiffrement côté serveur désactivé :

```
{
   "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"]
}
```

Désactiver le chiffrement côté serveur sur les versions 4.x

Example

Vous devez désactiver le chiffrement côté serveur à l'aide d'un objet `EmrConfiguration`.

L'exemple suivant crée une activité `EmrCluster` avec le chiffrement côté serveur désactivé :

```
   {
```

```
      "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"
    }
```

Configurer des listes ACL Hadoop KMS et créer des zones de chiffrement dans HDFS

Example

Les objets suivants créent les listes de contrôle d'accès (ACL) pour Hadoop KMS et créent des zones de chiffrement et les clés de chiffrement correspondantes dans 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"
    }
```

Spécifier les rôles IAM personnalisés

Example

Par défaut, il AWS Data Pipeline est transmis `DataPipelineDefaultRole` en tant que rôle de service Amazon EMR et `DataPipelineDefaultResourceRole` en tant que profil d'instance Amazon EC2 pour créer des ressources en votre nom. Cependant, vous pouvez créer un rôle de service Amazon EMR personnalisé et un profil d'instance personnalisé et les utiliser à la place. AWS Data Pipelinedoit disposer des autorisations suffisantes pour créer des clusters à l'aide du rôle personnalisé, et vous devez les ajouter AWS Data Pipeline en tant qu'entité de confiance.

L'exemple d'objet suivant spécifie des rôles personnalisés pour le cluster Amazon EMR :

```
{
   "id":"MyEmrCluster",
   "type":"EmrCluster",
   "hadoopVersion":"2.x",
   "keyPair":"my-key-pair",
   "masterInstanceType":"m3.xlarge",
   "coreInstanceType":"m3.large",
   "coreInstanceCount":"10",
   "taskInstanceType":"m3.large",
   "taskInstanceCount":"10",
   "role":"emrServiceRole",
   "resourceRole":"emrInstanceProfile"
}
```

Utiliser EmrCluster une ressource dans le kit AWS SDK for Java

Example

L'exemple suivant montre comment utiliser un `EmrCluster` et `EmrActivity` pour créer un cluster Amazon EMR 4.x pour exécuter une étape Spark à l'aide du SDK Java :

```
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);

     }

}
```

Configuration d'un cluster Amazon EMR dans un sous-réseau privé

Example

Cet exemple comprend une configuration qui lance le cluster dans un sous-réseau privé dans un VPC. Pour plus d'informations, consultez la section Lancer des clusters Amazon EMR dans un VPC dans le guide de gestion Amazon EMR. Cette configuration est facultative. Vous pouvez l'utiliser dans n'importe quel pipeline utilisant un objet `EmrCluster`.

Pour lancer un cluster Amazon EMR dans un sous-réseau privé, spécifiez, `SubnetId` `emrManagedMasterSecurityGroupIdemrManagedSlaveSecurityGroupId`, et `serviceAccessSecurityGroupId` dans votre configuration. `EmrCluster`

```
{
  "objects": [
    {
      "output": {
        "ref": "S3BackupLocation"
      },
      "input": {
        "ref": "DDBSourceTable"
      },
      "maximumRetries": "2",
      "name": "TableBackupActivity",
      "step": "s3://dynamodb-emr-#{myDDBRegion}/emr-ddb-storage-handler/2.1.0/emr-
ddb-2.1.0.jar,org.apache.hadoop.dynamodb.tools.DynamoDbExport,#{output.directoryPath},#{input.t
      "id": "TableBackupActivity",
      "runsOn": {
        "ref": "EmrClusterForBackup"
      },
      "type": "EmrActivity",
      "resizeClusterBeforeRunning": "false"
    },
    {
      "readThroughputPercent": "#{myDDBReadThroughputRatio}",
      "name": "DDBSourceTable",
      "id": "DDBSourceTable",
      "type": "DynamoDBDataNode",
      "tableName": "#{myDDBTableName}"
    },
    {
      "directoryPath": "#{myOutputS3Loc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-
mm-ss')}",
      "name": "S3BackupLocation",
      "id": "S3BackupLocation",
      "type": "S3DataNode"
    },
    {
      "name": "EmrClusterForBackup",
      "coreInstanceCount": "1",
      "taskInstanceCount": "1",
      "taskInstanceType": "m4.xlarge",
```

```
      "coreInstanceType": "m4.xlarge",
      "releaseLabel": "emr-4.7.0",
      "masterInstanceType": "m4.xlarge",
      "id": "EmrClusterForBackup",
      "subnetId": "#{mySubnetId}",
      "emrManagedMasterSecurityGroupId": "#{myMasterSecurityGroup}",
      "emrManagedSlaveSecurityGroupId": "#{mySlaveSecurityGroup}",
      "serviceAccessSecurityGroupId": "#{myServiceAccessSecurityGroup}",
      "region": "#{myDDBRegion}",
      "type": "EmrCluster",
      "keyPair": "user-key-pair"
    },
    {
      "failureAndRerunMode": "CASCADE",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "#{myPipelineLogUri}",
      "scheduleType": "ONDEMAND",
      "name": "Default",
      "id": "Default"
    }
  ],
  "parameters": [
    {
      "description": "Output S3 folder",
      "id": "myOutputS3Loc",
      "type": "AWS::S3::ObjectKey"
    },
    {
      "description": "Source DynamoDB table name",
      "id": "myDDBTableName",
      "type": "String"
    },
    {
      "default": "0.25",
      "watermark": "Enter value between 0.1-1.0",
      "description": "DynamoDB read throughput ratio",
      "id": "myDDBReadThroughputRatio",
      "type": "Double"
    },
    {
      "default": "us-east-1",
      "watermark": "us-east-1",
      "description": "Region of the DynamoDB table",
```

```
      "id": "myDDBRegion",
      "type": "String"
    }
  ],
  "values": {
     "myDDBRegion": "us-east-1",
      "myDDBTableName": "ddb_table",
      "myDDBReadThroughputRatio": "0.25",
      "myOutputS3Loc": "s3://s3_path",
      "mySubnetId": "subnet_id",
      "myServiceAccessSecurityGroup":  "service access security group",
      "mySlaveSecurityGroup": "slave security group",
      "myMasterSecurityGroup": "master security group",
      "myPipelineLogUri": "s3://s3_path"
  }
}
```

Attachement des volumes EBS aux nœuds de cluster

Example

Vous pouvez attacher des volumes EBS à n'importe quel type de nœud du cluster EMR dans de votre pipeline. Pour attacher des volumes EBS à des nœuds, utilisez `coreEbsConfiguration`, `masterEbsConfiguration`et `TaskEbsConfiguration` dans votre configuration `EmrCluster`.

Cet exemple de cluster Amazon EMR utilise des volumes Amazon EBS pour ses nœuds principaux, de tâches et principaux. Pour plus d'informations, consultez les volumes Amazon EBS dans Amazon EMR dans le guide de gestion Amazon EMR.

Ces configurations sont facultatives. Vous pouvez les utiliser dans n'importe quel pipeline utilisant un objet `EmrCluster`.

Dans le pipeline, cliquez sur la configuration d'objet `EmrCluster`, choisissez Master EBS Configuration (Configuration EBS maître), Core EBS Configuration (Configuration EBS principal) ou Task EBS Configuration (Configuration EBS de tâches) et saisissez les détails de configuration similaire à l'exemple suivant.

```
{
  "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"
  }
}
```

consultez aussi

- EmrActivity

HttpProxy

HttpProxy vous permet de configurer votre propre proxy et de permettre à Task Runner d'accéder au AWS Data Pipeline service via celui-ci. Vous n'avez pas besoin de configurer un Task Runner en cours d'exécution avec ces informations.

Exemple d' HttpProxy entrée TaskRunner

La définition de pipeline suivante présente un objet `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": []
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
hostname	Hôte du proxy que les clients utilisent pour se connecter aux services AWS.	Chaîne
port	Port de l'hôte proxy que les clients utilisent pour se connecter aux services AWS.	Chaîne

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple

Champs facultatifs	Description	Type d'option
		« parent » : {"ref » : » myBaseObject Id "}
*password	Mot de passe du proxy.	Chaîne
s3 NoProxy	Désactive le proxy HTTP lors de la connexion à Amazon S3.	Booléen
username	Nom d'utilisateur du proxy.	Chaîne
windowsDomain	Nom de domaine Windows pour le proxy NTLM.	Chaîne
windowsWorkgroup	Nom du groupe de travail Windows pour le proxy NTLM.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

Conditions préalables

Les objets suivants sont les objets de condition préalable AWS Data Pipeline :

Objets

- DynamoDB DataExists
- DynamoDB TableExists
- Existe
- S3 KeyExists
- S3 PrefixNotEmpty
- ShellCommandPrecondition

DynamoDB DataExists

Une condition préalable pour vérifier que les données existent dans une table DynamoDB.

Syntaxe

Champs obligatoires	Description	Type d'option
rôle	Spécifie le rôle à utiliser pour exécuter la condition préalable.	Chaîne
tableName	Table DynamoDB à vérifier.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période

Champs facultatifs	Description	Type d'option
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
preconditionTimeout	Période depuis le démarrage après laquelle la condition préalable est marquée comme ayant échoué si elle n'est toujours pas satisfaite	Période

Champs facultatifs	Description	Type d'option
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence, par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence, par exemple "cascadeFailedOn« : {" ref » : » myRunnableObject Id "}
currentRetryCount	Nombre de fois où la condition préalable a été essayée dans la tentative.	Chaîne

Champs liés à l'exécution	Description	Type d'option
emrStepLog	Journaux d'étapes EMR disponibles uniquement sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
lastRetryTime	Dernière fois où la condition préalable a été essayée au sein de la tentative.	Chaîne
nœud	Nœud pour lequel la condition préalable est en cours d'exécution	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

DynamoDB TableExists

Une condition préalable pour vérifier l'existence de la table DynamoDB.

Syntaxe

Champs obligatoires	Description	Type d'option
rôle	Spécifie le rôle à utiliser pour exécuter la condition préalable.	Chaîne
tableName	Table DynamoDB à vérifier.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
preconditionTimeout	Période depuis le démarrage après laquelle la condition préalable est marquée comme ayant échoué si elle n'est toujours pas satisfaite	Période
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
currentRetryCount	Nombre de fois où la condition préalable a été essayée dans la tentative.	Chaîne
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
lastRetryTime	Dernière fois où la condition préalable a été essayée au sein de la tentative.	Chaîne
nœud	Nœud pour lequel la condition préalable est en cours d'exécution	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime

Champs liés à l'exécution	Description	Type d'option
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

Existe

Vérifie si un objet de nœud de données existe.

> **Note**
>
> Nous vous recommandons d'utiliser à la place les conditions préalables gérées par le système. Pour plus d'informations, consultez Conditions préalables.

Exemple

Voici un exemple de ce type d'objet. L'objet `InputData` référence cet objet, `Ready`, ainsi qu'un autre objet que vous pourriez définir dans le même fichier de définition du pipeline. `CopyPeriod` est un objet `Schedule`.

```
{
  "id" : "InputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://example-bucket/InputData/#{@scheduledStartTime.format('YYYY-MM-dd-
hh:mm')}.csv",
  "precondition" : { "ref" : "Ready" }
},
{
  "id" : "Ready",
  "type" : "Exists"
}
```

Syntaxe

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période

Champs facultatifs	Description	Type d'option
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
preconditionTimeout	Période depuis le démarrage après laquelle la condition préalable est marquée comme ayant échoué si elle n'est toujours pas satisfaite	Période

Champs facultatifs	Description	Type d'option
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence, par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence, par exemple "cascadeFailedOn« : {" ref » : » myRunnableObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquement sur les tentatives d'activité EMR	Chaîne

Champs liés à l'exécution	Description	Type d'option
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
nœud	Nœud pour lequel la condition préalable est en cours d'exécution.	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- ShellCommandPrecondition

S3 KeyExists

Vérifie si une clé existe dans un nœud de données Amazon S3.

Exemple

Voici un exemple de ce type d'objet. La condition préalable se déclenche lorsqu'il existe une clé `s3://mybucket/mykey`, référencée par le paramètre `s3Key`.

```
{
"id" : "InputReady",
"type" : "S3KeyExists",
"role" : "test-role",
"s3Key" : "s3://mybucket/mykey"
}
```

Vous pouvez également utiliser `S3KeyExists` en tant que condition préalable sur le second pipeline qui attend la fin de l'exécution du premier pipeline. Pour ce faire :

1. Écrivez un fichier sur Amazon S3 à la fin du premier pipeline.
2. Créez une condition préalable `S3KeyExists` sur le second pipeline.

Syntaxe

Champs obligatoires	Description	Type d'option
rôle	Spécifie le rôle à utiliser pour exécuter la condition préalable.	Chaîne
s3Key	La clé Amazon S3.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai avant de tenter à nouveau de compléter la tâche à distance. Si une valeur est définie, toute activité à distance qui n'est pas exécutée pendant la période définie après le lancement fait l'objet d'une nouvelle tentative.	Période
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maximumRetries	Nombre maximum de tentatives initiées en cas d'échec.	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
preconditionTimeout	Période depuis le démarrage après laquelle la condition préalable est marquée comme ayant échoué si elle n'est toujours pas satisfaite.	Période
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à `reportProgress` . Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et font l'objet d'une nouvelle tentative.	Période
retryDelay	Délai entre deux tentative successives.	Période

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
currentRetryCount	Nombre de fois où la condition préalable a été essayée dans la tentative.	Chaîne
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne

Champs liés à l'exécution	Description	Type d'option
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
lastRetryTime	Dernière fois où la condition préalable a été essayée au sein de la tentative.	Chaîne
nœud	Nœud pour lequel la condition préalable est en cours d'exécution	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne

Champs système	Description	Type d'option
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

consultez aussi

- ShellCommandPrecondition

S3 PrefixNotEmpty

Une condition préalable pour vérifier que les objets Amazon S3 avec le préfixe donné (représenté sous forme d'URI) sont présents.

Exemple

Voici un exemple de ce type d'objet à l'aide de champs obligatoires, de champs facultatifs et de champs d'expression.

```
{
  "id" : "InputReady",
  "type" : "S3PrefixNotEmpty",
  "role" : "test-role",
  "s3Prefix" : "#{node.filePath}"
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
rôle	Spécifie le rôle à utiliser pour exécuter la condition préalable.	Chaîne

Champs obligatoires	Description	Type d'option
s3Prefix	Le préfixe Amazon S3 pour vérifier l'existence d'objets.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini surondemand.	Période
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
preconditionTimeout	Période depuis le démarrage après laquelle la condition préalable est marquée comme ayant échoué si elle n'est toujours pas satisfaite	Période
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellement planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnableObject Id "}

Champs liés à l'exécution	Description	Type d'option
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
currentRetryCount	Nombre de fois où la condition préalable a été essayée dans la tentative.	Chaîne
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne
lastRetryTime	Dernière fois où la condition préalable a été essayée au sein de la tentative.	Chaîne

Champs liés à l'exécution	Description	Type d'option
nœud	Nœud pour lequel la condition préalable est en cours d'exécution.	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

consultez aussi

- ShellCommandPrecondition

ShellCommandPrecondition

Commande shell Unix/Linux qui peut être exécutée en tant que condition préalable.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "id" : "VerifyDataReadiness",
  "type" : "ShellCommandPrecondition",
  "command" : "perl check-data-ready.pl"
}
```

Syntaxe

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
command	Commande à exécuter. Cette valeur et les paramètres associés doivent fonctionner dans l'environnement à partir duquel vous lancez l'exécuteur de tâches.	Chaîne
scriptUri	Chemin d'accès par URI Amazon S3 d'un fichier à télécharger et à exécuter en tant que commande shell. Un seul champ scriptUri ou command doit être présent. Étant donné que le champ scriptUri ne peut pas utiliser de paramètres, utilisez plutôt command.	Chaîne

Champs facultatifs	Description	Type d'option
attemptStatus	État de l'activité à distance le plus récemment rapporté.	Chaîne
attemptTimeout	Délai d'achèvement de la tâche à distance. Si une valeur est définie, une activité à distance qui n'est pas exécutée dans le cadre de la période de départ définie peut être retentée.	Période
failureAndRerunMode	Décrit le comportement du nœud de consommateurs lorsque les dépendances échouent ou sont à nouveau exécutées.	Énumération
lateAfterTimeout	Temps écoulé après le début du pipeline pendant lequel l'objet doit être terminé. Il est déclenché uniquement lorsque le type de planification n'est pas défini sur`ondemand`.	Période
maximumRetries	Nombre maximal de nouvelles tentatives en cas d'échec	Entier
onFail	Action à exécuter en cas d'échec de l'objet actuel.	Objet de référence , par exemple « onFail » : {"ref » : » myActionId «}
onLateAction	Actions à déclencher si un objet n'a pas encore été planifié ou n'est toujours pas terminé.	Objet de référence , par exemple "onLateAction« : {" ref » : » myActionId «}
onSuccess	Action à exécuter en cas de réussite de l'objet actuel.	Objet de référence , par exemple « onSuccess » : {"ref » : » myActionId «}

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
preconditionTimeout	Période depuis le démarrage après laquelle la condition préalable est marquée comme ayant échoué si elle n'est toujours pas satisfaite	Période
reportProgressTimeout	Délai pour les appels successifs de travail à distance adressés à reportProgress. Si une valeur est définie, les activités à distance qui ne font pas état d'avancement pour la période spécifiée doivent être considérées comme bloquées et, par conséquent, retentées.	Période
retryDelay	Délai entre deux nouvelles tentatives.	Période
scriptArgument	Argument à transmettre au script shell.	Chaîne
stderr	Le chemin Amazon S3 qui reçoit les messages d'erreur système redirigés depuis la commande. Si vous utilisez ce `runsOn` champ, il doit s'agir d'un chemin Amazon S3 en raison de la nature transitoire de la ressource exécutant votre activité. Toutefois, si vous spécifiez le champ `workerGroup` , un chemin de fichier local est autorisé.	Chaîne
stdout	Le chemin Amazon S3 qui reçoit la sortie redirigée de la commande. Si vous utilisez ce `runsOn` champ, il doit s'agir d'un chemin Amazon S3 en raison de la nature transitoire de la ressource exécutant votre activité. Toutefois, si vous spécifiez le champ `workerGroup` , un chemin de fichier local est autorisé.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@activeInstances	Liste des objets d'instances actives actuellem ent planifiés.	Objet de référence , par exemple « ActiveInstances » : {"ref » : » myRunnabl eObject Id "}
@actualEndTime	Heure à laquelle l'exécution de l'objet s'est terminée.	DateTime
@actualStartTime	Heure à laquelle l'exécution de l'objet a démarré.	DateTime
cancellationReason	Motif de l'annulation si l'objet a été annulé.	Chaîne
@cascadeFailedOn	Description de la chaîne de dépendances sur laquelle l'objet a échoué.	Objet de référence , par exemple "cascadeFailedOn« : {" ref » : » myRunnabl eObject Id "}
emrStepLog	Journaux d'étapes EMR disponibles uniquemen t sur les tentatives d'activité EMR	Chaîne
errorId	ID de l'erreur si l'objet a échoué.	Chaîne
errorMessage	errorMessage si l'objet a échoué.	Chaîne
errorStackTrace	Suivi de la pile d'erreurs si l'objet a échoué.	Chaîne
hadoopJobLog	Journaux de travail Hadoop disponibles sur les tentatives pour les activités EMR.	Chaîne
hostname	Nom d'hôte du client qui a sélectionné la tentative de tâche.	Chaîne

Champs liés à l'exécution	Description	Type d'option
nœud	Nœud pour lequel la condition préalable est en cours d'exécution	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
reportProgressTime	Heure la plus récente pour laquelle l'activité distante a signalé une progression.	DateTime
@scheduledEndTime	Heure de fin planifiée pour l'objet.	DateTime
@scheduledStartTime	Heure de début planifiée pour l'objet.	DateTime
@État	État de l'objet.	Chaîne
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne
@waitingOn	Description de la liste des dépendances sur laquelle l'objet est en attente.	Objet de référence , par exemple « WaitingOn » : {"ref » : » myRunnabl eObject Id "}

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

consultez aussi

- ShellCommandActivity
- Existe

Bases de données

Les objets suivants représentent les objets de base de données AWS Data Pipeline :

Objets

- JdbcDatabase
- RdsDatabase
- RedshiftDatabase

JdbcDatabase

Définit une base de données JDBC.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "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"
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
connectionChaîne	Chaîne de connexion JDBC permettant d'accéder à la base de données.	Chaîne

Champs obligatoires	Description	Type d'option
jdbcDriverClass	Classe de pilote à charger avant d'établir la connexion JDBC.	Chaîne
*password	Mot de passe à fournir.	Chaîne
username	Nom d'utilisateur à fournir lors de la connexion à la base de données.	Chaîne

Champs facultatifs	Description	Type d'option
databaseName	Nom de la base de données logique à laquelle s'attacher.	Chaîne
jdbcDriverJarUri	Emplacement dans Amazon S3 du fichier JAR du pilote JDBC utilisé pour se connecter à la base de données. AWS Data Pipeline doit avoir l'autorisation de lire le fichier JAR.	Chaîne
jdbcProperties	Paires sous la forme A = B qui seront définies comme propriétés sur les connexions JDBC de la base de données.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

RdsDatabase

Définit une base de données Amazon RDS.

> **Note**
>
> RdsDatabase ne prend pas en charge Aurora. the section called “JdbcDatabase”Utilisez-le plutôt pour Aurora.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "id" : "MyRdsDatabase",
  "type" : "RdsDatabase",
  "region" : "us-east-1",
  "username" : "user_name",
  "*password" : "my_password",
  "rdsInstanceId" : "my_db_instance_identifier"
}
```

Pour le moteur Oracle, le champ `jdbcDriverJarUri` est obligatoire et vous pouvez spécifier le pilote suivant : `http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html`. Pour le moteur SQL Server, le champ `jdbcDriverJarUri` est obligatoire et vous pouvez spécifier le pilote suivant : `https://`

`www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774`. Pour les moteurs MySQL et PostgreSQL, le champ `jdbcDriverJarUri` est facultatif.

Syntaxe

Champs obligatoires	Description	Type d'option
*password	Mot de passe à fournir.	Chaîne
rdsInstanceId	`DBInstanceIdentifier` Propriété de l'instance de base de données.	Chaîne
username	Nom d'utilisateur à fournir lors de la connexion à la base de données.	Chaîne

Champs facultatifs	Description	Type d'option
databaseName	Nom de la base de données logique à laquelle s'attacher.	Chaîne
jdbcDriverJarUri	Emplacement dans Amazon S3 du fichier JAR du pilote JDBC utilisé pour se connecter à la base de données. AWS Data Pipeline doit avoir l'autorisation de lire le fichier JAR. Pour les moteurs MySQL et PostgreSQL, le pilote par défaut est utilisé si ce champ n'est pas spécifié, mais vous pouvez remplacer la valeur par défaut à l'aide de ce champ. Pour les moteurs Oracle et SQL Server, ce champ est obligatoi re.	Chaîne
jdbcProperties	Paires sous la forme A = B qui seront définies comme propriétés sur les connexions JDBC de la base de données.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple,

Champs facultatifs	Description	Type d'option
		« parent » : {"ref » : » myBaseObject Id "}
region	Code de la région où se trouve la base de données. Par exemple, us-east-1.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

RedshiftDatabase

Définit une base de données Amazon Redshift. `RedshiftDatabase`représente les propriétés de la base de données utilisée par votre pipeline.

Exemple

Voici un exemple de ce type d'objet.

```
{
```

```
  "id" : "MyRedshiftDatabase",
  "type" : "RedshiftDatabase",
  "clusterId" : "myRedshiftClusterId",
  "username" : "user_name",
  "*password" : "my_password",
  "databaseName" : "database_name"
}
```

Par défaut, l'objet utilise le pilote Postgres, qui nécessite le champ `clusterId`. Pour utiliser le pilote Amazon Redshift, spécifiez plutôt la chaîne de connexion à la base de données Amazon Redshift depuis la console Amazon Redshift (qui commence par « jdbc:redshift : ») dans le champ. `connectionString`

Syntaxe

Champs obligatoires	Description	Type d'option
*password	Mot de passe à fournir.	Chaîne
username	Nom d'utilisateur à fournir lors de la connexion à la base de données.	Chaîne

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
clusterId	L'identifiant fourni par l'utilisateur lors de la création du cluster Amazon Redshift. Par exemple, si le point de terminaison de votre cluster Amazon Redshift est mydb.example.us-east-1.redshift.amazonaws.com, l'identifiant correct est. `mydb` Dans la console Amazon Redshift, vous pouvez obtenir cette valeur à partir des champs Cluster Identifier ou Cluster Name.	Chaîne

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
connectionChaîne	Point de terminaison JDBC permettant de se connecter à une instance Amazon Redshift détenue par un compte différent du pipeline. Vous ne pouvez pas spécifier à la fois `connectionString` et `clusterId` .	Chaîne

Champs facultatifs	Description	Type d'option
databaseName	Nom de la base de données logique à laquelle s'attacher.	Chaîne
jdbcProperties	Paires sous la forme A = B qui sont définies comme propriétés sur les connexions JDBC de la base de données.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple, « parent » : {"ref » : » myBaseObject Id "}
region	Code de la région où se trouve la base de données. Par exemple, us-east-1.	Énumération

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

Formats de données

Les objets suivants représentent les objets de format de données AWS Data Pipeline :

Objets

- Format de données CSV
- Format de données personnalisé
- DynamoDB DataFormat
- DynamoDB ExportDataFormat
- RegEx Format des données
- Format de données TSV

Format de données CSV

Format de données séparées par des virgules dans lequel le séparateur de colonnes est une virgule et le séparateur d'enregistrements un caractère de nouvelle ligne.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "id" : "MyOutputDataType",
  "type" : "CSV",
  "column" : [
```

```
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Syntaxe

Champs facultatifs	Description	Type d'option
column	Nom de colonne avec le type de données spécifié par chaque champ pour les données décrites par ce nœud de données. Exemple : nom d'hôte STRING. Pour plusieurs valeurs, utilisez les noms de colonnes et les types de données séparés par un espace.	Chaîne
escapeChar	Caractère (\, par exemple) qui indique à l'analyseur d'ignorer le caractère suivant.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne

Champs système	Description	Type d'option
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

Format de données personnalisé

Format de données personnalisé défini par la combinaison d'un séparateur de colonnes, d'un séparateur d'enregistrements et du caractère d'échappement.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "id" : "MyOutputDataType",
  "type" : "Custom",
  "columnSeparator" : ",",
  "recordSeparator" : "\n",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
columnSeparator	Caractère qui indique la fin d'une colonne dans un fichier de données.	Chaîne

Champs facultatifs	Description	Type d'option
column	Nom de colonne avec le type de données spécifié par chaque champ pour les données décrites par ce nœud de données. Exemple : nom d'hôte STRING. Pour plusieurs valeurs, utilisez les noms de colonnes et les types de données séparés par un espace.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}
recordSeparator	Caractère qui indique la fin d'une ligne dans un fichier de données, par exemple \n. Seuls les caractères uniques sont pris en charge.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

DynamoDB DataFormat

Applique un schéma à une table DynamoDB pour la rendre accessible par une requête Hive. `DynamoDBDataFormat`est utilisé avec un `HiveActivity` objet et une `DynamoDBDataNode` entrée et une sortie. `DynamoDBDataFormat`nécessite que vous spécifiiez toutes les colonnes de votre requête Hive. Pour plus de flexibilité dans la spécification de certaines colonnes dans une requête Hive ou pour le support Amazon S3, consultezDynamoDB ExportDataFormat.

> **Note**
>
> Les types booléens DynamoDB ne sont pas mappés aux types booléens Hive. Cependant, il est possible de mapper les valeurs entières DynamoDB de 0 ou 1 avec les types booléens Hive.

Exemple

L'exemple suivant montre comment utiliser `DynamoDBDataFormat` pour attribuer un schéma à une entrée `DynamoDBDataNode`, qui permet à un objet `HiveActivity` d'accéder aux données par colonnes nommées et de copier les données vers une sortie `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"
    }
  ]
}
```

Syntaxe

Champs facultatifs	Description	Type d'option
column	Nom de colonne avec le type de données spécifié par chaque champ pour les données décrites par ce nœud de données. Par exemple, `hostname STRING`. Pour plusieurs valeurs, utilisez des noms de colonnes et des types de données séparés par un espace.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , tel que « parent » : {"ref » : » myBaseObj ect Id "}

Champs liés à l'exécution	Description	Type d'option
@Version	Version de pipeline utilisée pour créer l'objet.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

DynamoDB ExportDataFormat

Applique un schéma à une table DynamoDB pour la rendre accessible par une requête Hive. Utilisez `DynamoDBExportDataFormat` avec un objet `HiveCopyActivity`, et une entrée et une sortie `DynamoDBDataNode` ou `S3DataNode`. `DynamoDBExportDataFormat` offre les avantages suivants :

- Fournit le support de DynamoDB et d'Amazon S3
- Permet de filtrer des données sur certaines colonnes dans votre requête Hive.
- Exporte tous les attributs depuis DynamoDB même si vous avez un schéma fragmenté

> **Note**
>
> Les types booléens DynamoDB ne sont pas mappés aux types booléens Hive. Cependant, il est possible de mapper les valeurs entières DynamoDB de 0 ou 1 avec les types booléens Hive.

Exemple

L'exemple suivant montre comment utiliser `HiveCopyActivity` et `DynamoDBExportDataFormat` pour copier les données d'un `DynamoDBDataNode` dans un autre, tout en filtrant les données en fonction de l'horodatage.

```
{
  "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"
    }
  ]
```

```
}
```

Syntaxe

Champs facultatifs	Description	Type d'option
column	Nom de colonne avec le type de données spécifié par chaque champ pour les données décrites par ce nœud de données. Exemple : nom d'hôte CHAINE	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

RegEx Format des données

Format de données personnalisé défini par une expression régulière.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "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"
  ]
}
```

Syntaxe

Champs facultatifs	Description	Type d'option
column	Nom de colonne avec le type de données spécifié par chaque champ pour les données décrites par ce nœud de données. Exemple : nom d'hôte STRING. Pour plusieurs valeurs, utilisez les noms de colonnes et les types de données séparés par un espace.	Chaîne
inputRegEx	Expression régulière pour analyser un fichier d'entrée S3. inputRegEx permet de récupérer	Chaîne

Champs facultatifs	Description	Type d'option
	des colonnes à partir de données relativement peu structurées d'un fichier.	
outputFormat	Les champs de colonne extraits par inputRegEx, mais référencés sous la forme %1$s %2$s à l'aide de la syntaxe du formateur Java.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

Format de données TSV

Format de données séparées par des virgules dans lequel le séparateur de colonnes est le caractère de tabulation et le séparateur d'enregistrements un caractère de nouvelle ligne.

Exemple

Voici un exemple de ce type d'objet.

```
{
  "id" : "MyOutputDataType",
  "type" : "TSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

Syntaxe

Champs facultatifs	Description	Type d'option
column	Nom de colonne et type des données décrites par ce nœud de données. Par exemple, "`Name STRING`" désigne une colonne nommée `Name` avec des champs de type de données `STRING`. Séparez les paires nom de colonne-type de données avec des virgules (comme indiqué dans l'exemple).	Chaîne
columnSeparator	Caractère de séparation des champs d'une colonne des champs de la colonne suivante. La valeur par défaut est '\t'.	Chaîne
escapeChar	Caractère (\, par exemple) qui indique à l'analyseur d'ignorer le caractère suivant.	Chaîne
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple,

Champs facultatifs	Description	Type d'option
		« parent » : {"ref » : » myBaseObject Id "}
recordSeparator	Caractère de séparation des enregistrements. La valeur par défaut est '\n'.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance », qui exécutent les objets « tentative ».	Chaîne

Actions

Les objets suivants sont les objets d'action AWS Data Pipeline :

Objets

- SnsAlarm
- Terminer

SnsAlarm

Envoie un message de notification Amazon SNS lorsqu'une activité échoue ou se termine correctement.

Exemple

Voici un exemple de ce type d'objet. Les valeurs de `node.input` et `node.output` proviennent du nœud de données ou de l'activité qui fait référence à cet objet dans son champ `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}."
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
message	Corps du texte de la notification Amazon SNS.	Chaîne
rôle	Rôle IAM à utiliser pour créer l'alarme Amazon SNS.	Chaîne
subject	Ligne d'objet du message de notification Amazon SNS.	Chaîne
topicArn	ARN de rubrique Amazon SNS de destination pour le message.	Chaîne

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple

Champs facultatifs	Description	Type d'option
		« parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
nœud	Nœud pour lequel cette action est en cours d'exécution.	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative ».	Chaîne

Terminer

Action pour déclencher l'annulation d'une activité inachevée ou en attente, d'une ressource ou d'un nœud de données. AWS Data Pipeline tente de placer l'activité, la ressource ou le nœud de données dans l'état CANCELLED s'ils ne commencent pas par la valeur `lateAfterTimeout`.

Vous ne pouvez pas mettre fin à des actions qui comprennent des ressources `onSuccess`, `OnFail` ou `onLateAction`.

Exemple

Voici un exemple de ce type d'objet. Dans cet exemple, le champ `onLateAction` de `MyActivity` contient une référence à l'action `DefaultAction1`. Lorsque vous fournissez une action pour `onLateAction`, vous devez également fournir une valeur `lateAfterTimeout` pour indiquer la période écoulée depuis le début planifié du pipeline qui indique que l'activité est en retard.

```
{
  "name" : "MyActivity",
  "id" : "DefaultActivity1",
  "schedule" : {
    "ref" : "MySchedule"
  },
  "runsOn" : {
    "ref" : "MyEmrCluster"
  },
  "lateAfterTimeout" : "1 Hours",
  "type" : "EmrActivity",
  "onLateAction" : {
    "ref" : "DefaultAction1"
  },
  "step" : [
    "s3://myBucket/myPath/myStep.jar,firstArg,secondArg",
    "s3://myBucket/myPath/myOtherStep.jar,anotherArg"
  ]
},
{
  "name" : "TerminateTasks",
  "id" : "DefaultAction1",
  "type" : "Terminate"
}
```

Syntaxe

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple

Champs facultatifs	Description	Type d'option
		« parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
nœud	Nœud pour lequel cette action est en cours d'exécution.	Objet de référence , par exemple « node » : {"ref » : » myRunnableObject Id "}
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance », qui exécutent les objets « tentative ».	Chaîne

Planificateur

Définit les informations temporelles d'un événement planifié, par exemple, le moment où une activité s'exécute.

> **Note**
>
> Lorsque l'heure de début d'un calendrier se trouve dans le passé, AWS Data Pipeline remplit votre pipeline et commence à planifier les exécutions immédiatement à compter de l'heure de début spécifiée. Pour les tests/le développement, utilisez un intervalle relativement court. Dans le cas contraire, AWS Data Pipeline tente de mettre en file d'attente et de planifier toutes les exécutions de votre pipeline pendant cet intervalle. AWS Data Pipeline tente d'éviter les renvois accidentels en bloquant l'activation du pipeline si le composant de pipeline `scheduledStartTime` est antérieur à il y a 1 jour.

Exemples

Voici un exemple de ce type d'objet. Il définit une planification toutes les heures à partir de 00:00:00 heure le 01/09/2012 et jusqu'à 00:00:00 heure le 01/10/2012. La première période se termine à 01:00:00 le 01/09/2012.

```
{
  "id" : "Hourly",
  "type" : "Schedule",
  "period" : "1 hours",
  "startDateTime" : "2012-09-01T00:00:00",
  "endDateTime" : "2012-10-01T00:00:00"
}
```

Le pipeline suivant démarre à `FIRST_ACTIVATION_DATE_TIME` et s'exécute toutes les heures jusqu'à 22:00:00 heures le 25/04/2014.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "endDateTime": "2014-04-25T22:00:00"
   }
```

Le pipeline suivante démarre à `FIRST_ACTIVATION_DATE_TIME`, s'exécute toutes les heures et prend fin après trois occurrences.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

Le pipeline suivant démarre à 22:00:00 le 25/04/2014, s'exécute toutes les heures et prend fin après trois occurrences.

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startDateTime": "2014-04-25T22:00:00",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

A la demande à l'aide de l'objet Default

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
}
```

A la demande à l'aide de l'objet explicite Schedule

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
},
{
  "name": "DefaultSchedule",
  "type": "Schedule",
```

```
  "id": "DefaultSchedule",
  "period": "ONDEMAND_PERIOD",
  "startAt": "ONDEMAND_ACTIVATION_TIME"
},
```

Les exemples suivants montrent comment un objet Schedule peut être hérité de l'objet Default, être explicitement défini pour cet objet ou être fourni par une référence Parent :

Objet Schedule hérité de l'objet 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!'"
    }
```

```
  ]
}
```

Objet Schedule explicite sur l'objet

```
{
  "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!'"
    }
  ]
}
```

Objet Schedule de la référence Parent

```
{
  "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!'"
    }
  ]
}
```

Syntaxe

Champs obligatoires	Description	Type d'option
point	Fréquence d'exécution du pipeline. Le format est « N [minutes|heures|jours|semaines|mois] », où N est un nombre suivi d'un des spécificateurs de temps. Par exemple, la valeur « 15 minutes » exécute le pipeline toutes les 15 minutes. La période minimale est de 15 minutes et la durée maximale de 3 ans.	Période

Groupe obligatoire (l'un des groupes suivants est obligatoire)	Description	Type d'option
startAt	Date et heure de début des exécutions planifiées du pipeline. La valeur valide est FIRST_ACTIVATION_DATE_TIME, qui est obsolète et remplacée par la création d'un pipeline à la demande.	Énumération
startDateTime	Date et heure de début des exécutions planifiées. Vous devez utiliser l'un ou l'autre startDateTime ou StartAt, mais pas les deux.	DateTime

Champs facultatifs	Description	Type d'option
endDateTime	Date et heure de fin des exécutions planifiées. La date et l'heure doivent être postérieures à la valeur de startDateTime ou StartAt. Le	DateTime

Champs facultatifs	Description	Type d'option
	comportement par défaut consiste à planifier les exécutions jusqu'à l'arrêt du pipeline.	
occurrences	Nombre d'exécutions du pipeline après son activation. Vous ne pouvez pas utiliser d'occurrences avec endDateTime.	Entier
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@firstActivationTime	Heure de création de l'objet.	DateTime
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

Utilitaires

Les objets d'utilitaire suivants configurent les autres objets du pipeline :

Rubriques

ShellScriptConfig

À utiliser avec une activité pour exécuter un script shell pour preActivityTask Config et postActivityTask Config. Cet objet est disponible pour HadoopActivityHiveActivity, HiveCopyActivity, et PigActivity. Vous pouvez spécifier un URI S3 et une liste d'arguments pour le script.

Exemple

A ShellScriptConfig avec des arguments :

```
{
   "id" : "ShellScriptConfig_1”,
   "name" : “prescript”,
   "type" : "ShellScriptConfig",
   "scriptUri": “s3://my-bucket/shell-cleanup.sh”,
   "scriptArgument" : ["arg1","arg2"]
 }
```

Syntaxe

Cet objet inclut les champs suivants.

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple, « parent » : {"ref » : » myBaseObject Id "}

Champs facultatifs	Description	Type d'option
scriptArgument	Liste d'arguments à utiliser avec le script shell.	Chaîne
scriptUri	URI du script dans Amazon S3 qui doit être téléchargé et exécuté.	Chaîne

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance », qui exécutent les objets « tentative ».	Chaîne

EmrConfiguration

L' EmrConfiguration objet est la configuration utilisée pour les clusters EMR avec les versions 4.0.0 ou supérieures. Les configurations (sous forme de liste) sont un paramètre de l'appel RunJobFlow d'API. L'API de configuration pour Amazon EMR utilise une classification et des propriétés. AWS Data Pipelineutilise EmrConfiguration avec les objets Property correspondants pour configurer une EmrCluster application telle que Hadoop, Hive, Spark ou Pig sur des clusters EMR lancés lors d'une exécution de pipeline. Comme la configuration ne peut être modifiée que pour les nouveaux clusters, vous ne pouvez pas fournir d' EmrConfiguration objet pour les ressources existantes. Pour plus d'informations, consultez https://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/.

Exemple

L'objet de configuration suivant définit les propriétés `io.file.buffer.size` et `fs.s3.block.size` dans `core-site.xml` :

```
[
   {
      "classification":"core-site",
      "properties":
      {
         "io.file.buffer.size": "4096",
         "fs.s3.block.size": "67108864"
      }
   }
]
```

La définition d'objet de pipeline correspondante utilise un EmrConfiguration objet et une liste d'objets Property dans le `property` champ :

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      },
      {
        "ref": "fs-s3-block-size"
      }
      ]
```

```
    },
    {
      "name": "io-file-buffer-size",
      "id": "io-file-buffer-size",
      "type": "Property",
      "key": "io.file.buffer.size",
      "value": "4096"
    },
    {
      "name": "fs-s3-block-size",
      "id": "fs-s3-block-size",
      "type": "Property",
      "key": "fs.s3.block.size",
      "value": "67108864"
    }
  ]
}
```

L'exemple suivant illustre une configuration imbriquée utilisée pour définir l'environnement Hadoop avec la classification `hadoop-env` :

```
[
  {
    "classification": "hadoop-env",
    "properties": {},
    "configurations": [
      {
        "classification": "export",
        "properties": {
          "YARN_PROXYSERVER_HEAPSIZE": "2396"
        }
      }
    ]
  }
]
```

L'objet de définition de pipeline correspondant qui utilise cette configuration se trouve ci-après :

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.0.0",
```

```
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "hadoop-env"
      }
    },
    {
      "name": "hadoop-env",
      "id": "hadoop-env",
      "type": "EmrConfiguration",
      "classification": "hadoop-env",
      "configuration": {
        "ref": "export"
      }
    },
    {
      "name": "export",
      "id": "export",
      "type": "EmrConfiguration",
      "classification": "export",
      "property": {
        "ref": "yarn-proxyserver-heapsize"
      }
    },
    {
      "name": "yarn-proxyserver-heapsize",
      "id": "yarn-proxyserver-heapsize",
      "type": "Property",
      "key": "YARN_PROXYSERVER_HEAPSIZE",
      "value": "2396"
    },
  ]
}
```

L'exemple suivant modifie une propriété spécifique à Hive pour un cluster EMR :

```
{
    "objects": [
        {
            "name": "hivesite",
            "id": "hivesite",
            "type": "EmrConfiguration",
```

```
            "classification": "hive-site",
            "property": [
                {
                    "ref": "hive-client-timeout"
                }
            ]
        },
        {
            "name": "hive-client-timeout",
            "id": "hive-client-timeout",
            "type": "Property",
            "key": "hive.metastore.client.socket.timeout",
            "value": "2400s"
        }
    ]
}
```

Syntaxe

Cet objet inclut les champs suivants.

Champs obligatoires	Description	Type d'option
classification	Classification de la configuration.	Chaîne

Champs facultatifs	Description	Type d'option
configuration	Sous-configuration de la configuration.	Objet de référence , par exemple « configuration » : {"ref » : » myEmrConf iguration Id "}
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple « parent » : {"ref » : » myBaseObject Id "}

Champs facultatifs	Description	Type d'option
property	Propriété de configuration	Objet de référence , par exemple « property » : {"ref » : » myPropert yId «}

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	Id du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance » qui exécutent les objets « tentative »	Chaîne

consultez aussi

- EmrCluster
- Propriété
- Amazon EMR Guide de version

Propriété

Propriété clé-valeur unique à utiliser avec un EmrConfiguration objet.

Exemple

La définition de pipeline suivante montre un EmrConfiguration objet et les objets Property correspondants pour lancer un EmrCluster :

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      },
      {
        "ref": "fs-s3-block-size"
      }
      ]
    },
    {
      "name": "io-file-buffer-size",
      "id": "io-file-buffer-size",
      "type": "Property",
      "key": "io.file.buffer.size",
      "value": "4096"
    },
    {
      "name": "fs-s3-block-size",
```

```
      "id": "fs-s3-block-size",
      "type": "Property",
      "key": "fs.s3.block.size",
      "value": "67108864"
    }
  ]
}
```

Syntaxe

Cet objet inclut les champs suivants.

Champs obligatoires	Description	Type d'option
key	key	Chaîne
value	value	Chaîne

Champs facultatifs	Description	Type d'option
parent	Parent de l'objet actuel à partir duquel les emplacements sont hérités.	Objet de référence , par exemple, « parent » : {"ref » : » myBaseObject Id "}

Champs liés à l'exécution	Description	Type d'option
@Version	Version du pipeline avec laquelle l'objet a été créé.	Chaîne

Champs système	Description	Type d'option
@error	Erreur décrivant l'objet mal formé.	Chaîne
@pipelineId	ID du pipeline auquel l'objet appartient.	Chaîne
@sphere	La sphère d'un objet désigne sa place dans le cycle de vie : les objets « composant » entraînent les objets « instance », qui exécutent les objets « tentative ».	Chaîne

consultez aussi

- EmrCluster
- EmrConfiguration
- Amazon EMR Guide de version

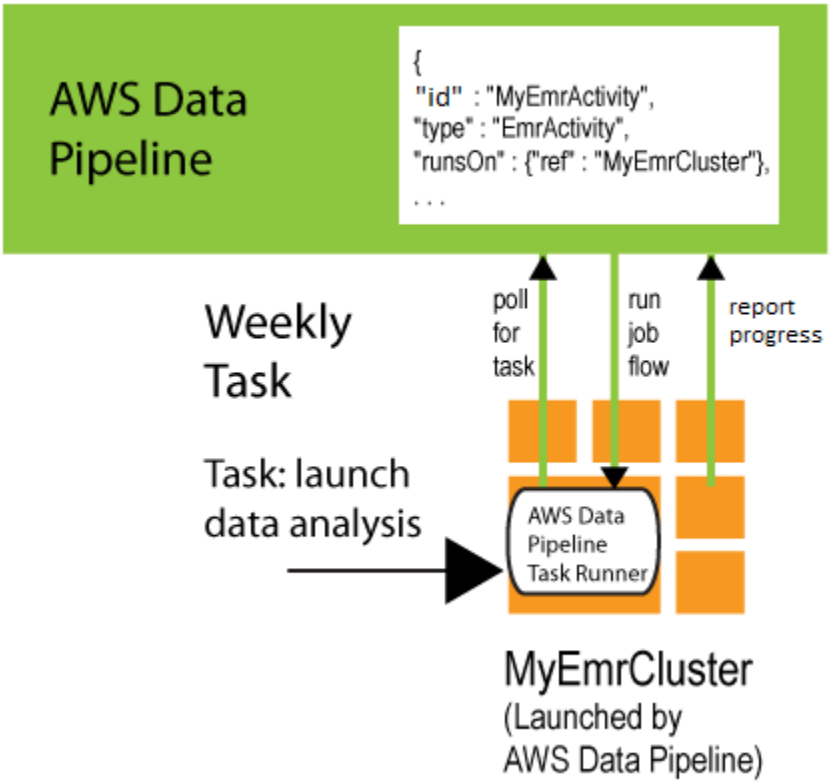
Utilisation de Task Runner

Task Runner est une application d'agentAWS Data Pipeline de tâches qui interroge les tâches planifiées et les exécute sur des instances Amazon EC2, des clusters Amazon EMR ou d'autres ressources informatiques, tout en signalant leur état. Selon votre application, vous pouvez choisir d'effectuer les actions suivantes :

- PermetAWS Data Pipeline d'installer et de gérer une ou plusieurs applications Task Runner pour vous. Lorsqu'un pipeline est activé, la valeur par défaut`Ec2Instance` ou`EmrCluster` l'objet référencé par un champ d'activité RunsOn est automatiquement créé. AWS Data Pipelinese charge d'installer Task Runner sur une instance EC2 ou sur le nœud maître d'un cluster EMR. Dans ce modèle, AWS Data Pipeline peut effectuer la majeure partie de la gestion de l'instances ou du cluster à votre place.
- Exécuter tout ou partie d'un pipeline sur des ressources que vous gérez. Les ressources potentielles incluent une instance Amazon EC2 de longue durée, un cluster Amazon EMR ou un serveur physique. Vous pouvez installer un exécuteur de tâches (qui peut être soit un exécuteur de tâches, soit un agent de tâches personnalisé créé par vos soins) à peu près n'importe où, à condition qu'il puisse communiquer avec le serviceAWS Data Pipeline Web. Dans ce modèle, vous contrôlez presque totalement les ressources utilisées et la manière dont elles sont gérées, et vous devez installer et configurer manuellement Task Runner. Pour ce faire, utilisez les procédures de cette section, décrites dans Exécution de tâches sur des ressources existantes à l'aide de Task Runner.

Exécuteur de tâches sur les ressourcesAWS Data Pipeline gérées

Lorsqu'une ressource est lancée et gérée parAWS Data Pipeline, le service Web installe automatiquement Task Runner sur cette ressource pour traiter les tâches du pipeline. Vous spécifiez une ressource informatique (instance Amazon EC2 ou cluster Amazon EMR) pour le`runs0n` champ d'un objet d'activité. Lorsque cette ressource estAWS Data Pipeline lancée, il installe Task Runner sur cette ressource et le configure pour traiter tous les objets d'activité dont`runs0n` le champ est défini sur cette ressource. LorsqueAWS Data Pipeline la ressource est interrompue, les journaux de Task Runner sont publiés sur un emplacement Amazon S3 avant sa fermeture.



Supposons par exemple, que vous utilisez l'activité `EmrActivity` dans un pipeline et spécifiez une ressource `EmrCluster` dans le champ `runsOn`. LorsAWS Data Pipeline du traitement de cette activité, il lance un cluster Amazon EMR et installe Task Runner sur le nœud maître. Cet exécuteur de tâches traite ensuite les tâches relatives aux activités dont`runsOn` le champ est défini sur cet`EmrCluster` objet. L'extrait suivant d'une définition de pipeline montre la relation entre les deux objets.

```
{
  "id" : "MyEmrActivity",
  "name" : "Work to perform on my data",
  "type" : "EmrActivity",
  "runsOn" : {"ref" : "MyEmrCluster"},
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : "s3://myBucket/myPath/myStep.jar,firstArg,secondArg",
  "step" : "s3://myBucket/myPath/myOtherStep.jar,anotherArg",
  "postStepCommand" : "scp localFiles remoteFiles",
  "input" : {"ref" : "MyS3Input"},
  "output" : {"ref" : "MyS3Output"}
},
{
```

```
  "id" : "MyEmrCluster",
  "name" : "EMR cluster to perform the work",
  "type" : "EmrCluster",
  "hadoopVersion" : "0.20",
  "keypair" : "myKeyPair",
  "masterInstanceType" : "m1.xlarge",
  "coreInstanceType" : "m1.small",
  "coreInstanceCount" : "10",
  "taskInstanceType" : "m1.small",
  "taskInstanceCount": "10",
  "bootstrapAction" : "s3://elasticmapreduce/libs/ba/configure-hadoop,arg1,arg2,arg3",
  "bootstrapAction" : "s3://elasticmapreduce/libs/ba/configure-other-stuff,arg1,arg2"
}
```

Pour plus d'informations et des exemples d'exécution de cette activité, consultez EmrActivity.

Si vous avez plusieursAWS Data Pipeline ressources gérées dans un pipeline, Task Runner est installé sur chacune d'elles et toutes interrogentAWS Data Pipeline les tâches à traiter.

Exécution de tâches sur des ressources existantes à l'aide de Task Runner

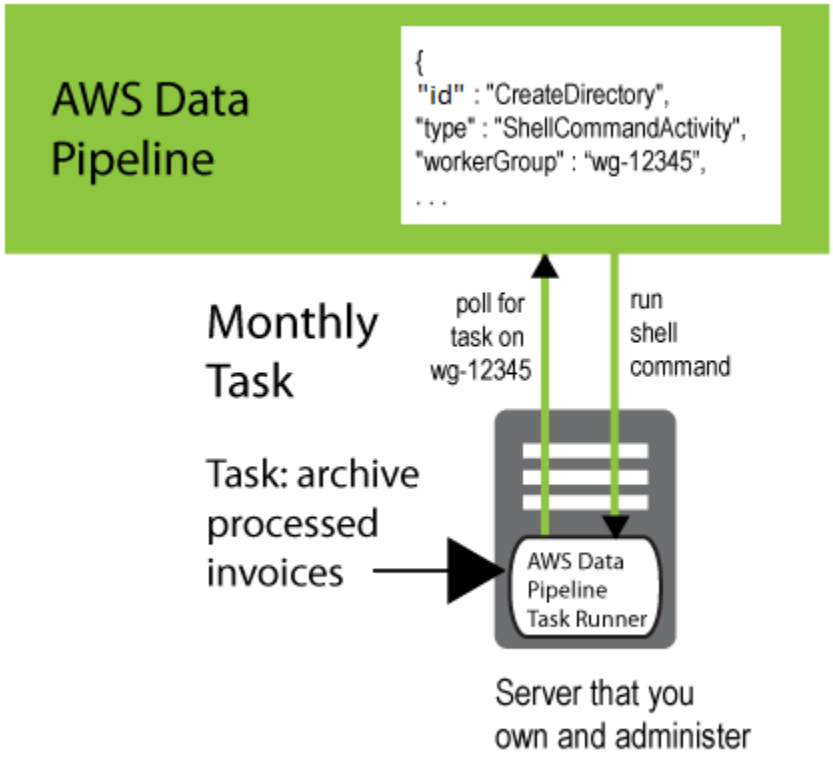
Vous pouvez installer Task Runner sur des ressources informatiques que vous gérez, telles qu'une instance Amazon EC2, un serveur physique ou une station de travail. Task Runner peut être installé n'importe où, sur n'importe quel matériel ou système d'exploitation compatible, à condition qu'il puisse communiquer avec le serviceAWS Data Pipeline Web.

Cette approche peut être utile, par exemple, quand vous voulez utiliser AWS Data Pipeline pour traiter les données qui sont stockées à l'intérieur du pare-feu de votre organisation. En installant Task Runner sur un serveur du réseau local, vous pouvez accéder à la base de données locale en toute sécurité, puis interrogerAWS Data Pipeline la prochaine tâche à exécuter. Lorsque le traitement estAWS Data Pipeline terminé ou que le pipeline est supprimé, l'instance Task Runner continue de s'exécuter sur votre ressource de calcul jusqu'à ce que vous l'arrêtiez manuellement. Les journaux de Task Runner persistent une fois l'exécution du pipeline terminée.

Pour utiliser Task Runner sur une ressource que vous gérez, vous devez d'abord télécharger Task Runner, puis l'installer sur votre ressource informatique en suivant les procédures décrites dans cette section.

> **Note**
>
> Vous ne pouvez installer Task Runner que sous Linux, UNIX ou macOS. Task Runner n'est pas pris en charge sur le système d'exploitation Windows.
> Pour utiliser Task Runner 2.0, la version Java minimale requise est 1.7.

Pour connecter un Task Runner que vous avez installé aux activités du pipeline qu'il doit traiter, ajoutez un`workerGroup` champ à l'objet et configurez Task Runner pour qu'il interroge la valeur de ce groupe de travail. Pour ce faire, transmettez la chaîne du groupe de travail en tant que paramètre (par exemple`--workerGroup=wg-12345`) lorsque vous exécutez le fichier JAR Task Runner.




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

Installation Task Runner

Cette section explique comment installer et configurer Task Runner. L'installation est un processus manuel simple.

Pour installer Task Runner

1. Task Runner nécessite les versions 1.6 ou 1.8 de Java. Pour déterminer si Java est installé et la version qui est en cours d'exécution, utilisez la commande suivante :

   ```
   java -version
   ```

   Si Java 1.6 ou 1.8 n'est pas installé sur votre ordinateur, téléchargez l'une de ces versions à l'adresse http://www.oracle.com/technetwork/java/index.html. Téléchargez et installez Java, puis passez à l'étape suivante.

2. Téléchargez`TaskRunner-1.0.jar` depuis https://s3.amazonaws.com/datapipeline-us-east-1/us-east-1/software/latest/TaskRunner/TaskRunner-1.0.jar, puis copiez-le dans un dossier de la ressource informatique cible. Pour les clusters Amazon EMR exécutant`EmrActivity` des tâches, installez Task Runner sur le nœud principal du cluster.

3. Lorsqu'ils utilisent Task Runner pour se connecter au serviceAWS Data Pipeline Web afin de traiter vos commandes, les utilisateurs ont besoin d'un accès programmatique à un rôle autorisé à créer ou à gérer des pipelines de données. Pour plus d'informations, veuillez consulter Accorder un accès par programmation.

4. Task Runner se connecte au serviceAWS Data Pipeline Web via HTTPS. Si vous utilisez une ressource AWS, assurez-vous que le protocole HTTPS est activé dans la table de routage et la liste de contrôle d'accès de sous-réseau appropriées. Si vous utilisez un pare-feu ou un proxy, assurez-vous que le port 443 est ouvert.

(Facultatif) Octroi d'un accès à Amazon RDS à Task Runner

Amazon RDS vous permet de contrôler l'accès à vos instances DB à l'aide des groupes de sécurité de base de données (groupes de sécurité DB). Un Security Group DB agit comme un pare-feu contrôlant l'accès du réseau à votre instance de base de données. Par défaut, l'accès réseau est désactivé pour vos instances de base de données. Vous devez modifier vos groupes de sécurité de base de données pour permettre à Task Runner d'accéder à vos instances Amazon RDS. Task Runner accède à Amazon RDS depuis l'instance sur laquelle il s'exécute. Les comptes et les groupes

de sécurité que vous ajoutez à votre instance Amazon RDS dépendent donc de l'endroit où vous installez Task Runner.

Pour accorder l'accès à Task Runner dans EC2-Classic

1. Ouvrez la console Amazon RDS.
2. Dans le volet de navigation, choisissez Instances , puis sélectionnez votre instance de base de données.
3. Sous Sécurité et réseau, sélectionnez le groupe de sécurité, qui ouvre la page Groupes de sécurité dans laquelle ce groupe de sécurité de base de données est sélectionné. Sélectionnez l'icône de détails du groupe de sécurité de base de données.
4. Sous Informations concernant le groupe de sécurité, créez une règle avec les valeurs appropriées pour Type de connexion et Détails. Ces champs dépendent de l'emplacement d'exécution de Task Runner, comme décrit ici :

   - `Ec2Resource`
     - Type de connexion : `EC2 Security Group`

       Détails : *`my-security-group-name`*(nom du groupe de sécurité que vous avez créé pour l'instance EC2)
   - `EmrResource`
     - Type de connexion : `EC2 Security Group`

       Détails: `ElasticMapReduce-master`
     - Type de connexion : `EC2 Security Group`

       Détails: `ElasticMapReduce-slave`
   - Votre environnement local (sur site)
     - Type de connexion : `CIDR/IP`

       Détails : *`my-ip-address`*(adresse IP de votre ordinateur ou plage d'adresses IP de votre réseau, si votre ordinateur est protégé par un pare-feu)
5. Cliquez sur Ajouter.

Pour accorder l'accès à Task Runner dans EC2-VPC

1. Ouvrez la console Amazon RDS.

2. Dans le panneau de navigation, sélectionnez Instances.
3. Sélectionnez l'icône de détails de l'instance de base de données. Sous Sécurité et réseau, ouvrez le lien vers le groupe de sécurité, qui vous permet d'accéder à la console Amazon EC2. Si vous utilisez l'ancienne console pour les groupes de sécurité, passez à la nouvelle console en sélectionnant l'icône qui s'affiche en haut de la page de la console.
4. Sous l'onglet Inbound, choisissez Edit, Add Rule. Spécifiez le port de base de données que vous avez utilisé lors du lancement de l'instance de base de données. La source dépend de l'emplacement d'exécution de Task Runner, comme décrit ici :

   - `Ec2Resource`
     - *`my-security-group-id`*(l'ID du groupe de sécurité que vous avez créé pour l'instance EC2)
   - `EmrResource`
     - *`master-security-group-id`*(l'ID du groupe`ElasticMapReduce-master` de sécurité)
     - *`slave-security-group-id`*(l'ID du groupe`ElasticMapReduce-slave` de sécurité)
   - Votre environnement local (sur site)
     - *`ip-address`* (adresse IP de votre ordinateur ou plage d'adresses IP de votre réseau si votre ordinateur se trouve derrière un pare-feu)
5. Cliquez sur Sauvegarder

Démarrage de Task Run

Dans une nouvelle fenêtre d'invite de commande définie sur le répertoire dans lequel vous avez installé Task Runner, démarrez Task Runner à l'aide de la commande suivante.

```
java -jar TaskRunner-1.0.jar --config ~/credentials.json --workerGroup=myWorkerGroup --
region=MyRegion --logUri=s3://mybucket/foldername
```

L'option `--config` pointe vers votre fichier d'informations d'identification.

L'option `--workerGroup` indique le nom de votre groupe de travail, qui doit être identique à la valeur indiquée dans votre pipeline pour les tâches à traiter.

L'option `--region` indique la région du service où les tâches à exécuter doivent être récupérées.

`--logUri`Cette option est utilisée pour transférer vos journaux compressés vers un emplacement dans Amazon S3.

Lorsque Task Runner est actif, il affiche le chemin vers lequel les fichiers journaux sont écrits dans la fenêtre du terminal. Voici un exemple.

```
Logging to /Computer_Name/.../output/logs
```

Task Runner doit être exécuté détaché de votre shell de connexion. Si vous utilisez une application de terminal pour vous connecter à votre ordinateur, vous devrez peut-être utiliser un utilitaire comme nohup ou screen pour empêcher l'application Task Runner de se fermer lorsque vous vous déconnecterez. Pour plus d'informations sur les options de ligne de commande, consultez Options de configuration de Task Runner.

Vérification de la journalisation de Task Runner

Le moyen le plus simple de vérifier que Task Runner fonctionne est de vérifier s'il écrit des fichiers journaux. Task Runner écrit des fichiers journaux horaires dans le répertoire,`output/logs`, situé sous le répertoire dans lequel Task Runner est installé. Le nom du fichier est `Task Runner.log.YYYY-MM-DD-HH`, où HH va de 00 à 23, au format UDT. Pour économiser de l'espace de stockage, tous les fichiers journaux de plus de huit heures sont compressés avec GZip.

Sujets et conditions préalables à Task Runner

Task Runner utilise un pool de threads pour chacune des tâches, activités et conditions préalables. La valeur par défaut du paramètre `--tasks` est 2, ce qui signifie que deux threads sont alloués à partir du pool de tâches et que chaque thread interroge le service AWS Data Pipeline pour rechercher les nouvelles tâches. Ainsi, `--tasks` est un attribut d'ajustement des performances qui peut être utilisé pour optimiser le débit du pipeline.

La logique de nouvelle tentative du pipeline pour les conditions préalables se produit dans Task Runner. Deux threads de condition préalable sont alloués pour interroger AWS Data Pipeline et rechercher les objets de condition préalable. Task Runner respecte les champs RetryDelay et PreconditionTimeout de l'objet précondition que vous définissez sur les préconditions.

Dans de nombreux cas, la diminution du délai d'interrogation des conditions préalables et du nombre de nouvelles tentatives contribue à améliorer les performances de votre application. De même, les applications ayant des conditions préalables de longue durée peuvent nécessiter une augmentation des valeurs de délai et de nouvelles tentatives. Pour plus d'informations sur les objets de condition préalable, consultez Conditions préalables.

Options de configuration de Task Runner

Voici les options de configuration disponibles depuis la ligne de commande lorsque vous lancez Task Runner.

Paramètre de ligne de commande	Description
`--help`	Aide de ligne de commande. Exemple : `Java -jar TaskRunner-1.0.jar --help`
`--config`	Chemin et nom de votre fichier `credentials.json` .
`--accessId`	Votre identifiant de clé d'AWSaccès que Task Runner peut utiliser pour effectuer des demandes. Les`--secretKey` options`--accessID` et fournissent une alternative à l'utilisation d'un fichier credentials.json. Si un fichier `credentials.json` est également fourni, les options `--accessID` et `--secretKey` sont prioritaires.
`--secretKey`	Votre cléAWS secrète que Task Runner peut utiliser pour effectuer des demandes. Pour plus d'informations, veuillez consulter `--accessID` .
`--endpoint`	Un point de terminaison est une URL qui est le point d'entrée d'un service Web. Point de terminaison du service AWS Data Pipeline dans la région dans laquelle vous effectuez les demandes. Facultatif. En général, il suffit de spécifier une région, et vous n'avez pas besoin de définir le point de terminaison. Pour obtenir la liste des régions et points de terminaison AWS Data Pipeline, consultez Régions et

Paramètre de ligne de commande	Description
	points de terminaison AWS Data Pipeline dans le document Références générales AWS.
`--workerGroup`	Nom du groupe de travail pour lequel Task Runner récupère le travail. Obligatoire. Lorsque Task Runner interroge le service Web, il utilise les informations d'identification que vous avez fournies et la valeur de`workerGroup` pour sélectionner les tâches (le cas échéant) à récupérer. Vous pouvez utiliser n'importe quel nom qui vous semble significatif ; la seule exigence est que la chaîne correspond à celle du Task Runner et de ses activités de pipeline correspondantes. Le nom du groupe de travail est lié à une région. Même s'il existe des noms de groupes de travail identiques dans d'autres régions, Task Runner obtient toujours les tâches de la région spécifiée dans`--region`.
`--taskrunnerId`	ID de l'exécuteur de tâches à utiliser pour les rapports d'avancement. Facultatif.
`--output`	Le répertoire Task Runner pour les fichiers de sortie des journaux. Facultatif. Les fichiers journaux sont stockés dans un répertoire local jusqu'à ce qu'ils soient transférés vers Amazon S3. Cette option remplace le répertoire par défaut.

Paramètre de ligne de commande	Description
`--region`	Région à utiliser. Facultatif, mais il est recommandé de toujours définir la région. Si vous ne spécifiez pas la région, Task Runner extrait les tâches de la région de service par défaut`us-east-1` . Les autres régions prises en charge sont : `eu-west-1` , `ap-northeast-1` , `ap-southeast-2` , `us-west-2` .
`--logUri`	Le chemin de destination Amazon S3 sur lequel Task Runner doit sauvegarder les fichiers journaux toutes les heures. Lorsque Task Runner se termine, les journaux actifs du répertoire local sont envoyés vers le dossier de destination Amazon S3.
`--proxyHost`	Hôte du proxy utilisé par les clients Task Runner pour se connecter aux services AWS.
`--proxyPort`	Port de l'hôte proxy utilisé par les clients Task Runner pour se connecter aux services AWS.
`--proxyUsername`	Nom d'utilisateur pour le proxy.
`--proxyPassword`	Mot de passe pour le proxy.
`--proxyDomain`	Nom de domaine Windows pour le proxy NTLM.
`--proxyWorkstation`	Nom de poste de travail Windows pour NTLM Proxy.

Utilisation de Task Runner avec un proxy

Si vous utilisez un hôte proxy, vous pouvez spécifier sa configuration lors de l'appel de Task Runner ou définir la variable d'environnement HTTPS_PROXY. La variable d'environnement utilisée avec Task Runner accepte la même configuration que celle utilisée pour l'interface de ligne de commande AWS.

Task Runner et AMI personnalisées

Lorsque vous spécifiez un`Ec2Resource` objet pour votre pipeline, vousAWS Data Pipeline créez une instance EC2 à l'aide d'une AMI qui installe et configure Task Runner pour vous. Un type d'instance compatible avec PV est requis dans ce cas. Vous pouvez également créer une AMI personnalisée avec Task Runner, puis spécifier l'ID de cette AMI à l'aide du`imageId` champ de l'`Ec2Resource`objet. Pour plus d'informations, veuillez consulter Ec2Resource.

Une AMI personnalisée doit répondre aux exigences suivantesAWS Data Pipeline pour pouvoir être utilisée correctement pour Task Runner :

- Créez l'AMI dans la région où les instances vont s'exécuter. Pour plus d'informations, consultez Création de votre propre AMI dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
- Assurez-vous que le type de virtualisation de l'AMI est pris en charge par le type d'instance que vous prévoyez d'utiliser. Par exemple, les types d'instance I2 et G2 exigent une AMI HVM et les types d'instance T1, C1, M1 et M2 exigent une AMI PV. Pour plus d'informations, consultez Types de virtualisation AMI Linux dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
- Installez les logiciels suivants :
  - Linux
  - Bash
  - wget
  - unzip
  - Java 1.6 ou 1.8
  - cloud-init
- Créez et configurez un utilisateur nommé`ec2-user`.

Résolution des problèmes

Lorsque vous avez un problème avec AWS Data Pipeline, le symptôme le plus courant est qu'un pipeline ne s'exécute pas. Vous pouvez utiliser les données fournies par la console et l'interface de ligne de commande pour identifier le problème et trouver une solution.

Table des matières

Localisation des erreurs dans les pipelines

La console AWS Data Pipeline est un outil pratique qui vous permet de surveiller visuellement l'état de vos pipelines et de localiser aisément les erreurs liées à des exécutions de pipeline incomplètes ou ayant échoué.

Pour localiser les erreurs liées à des exécutions incomplètes ou ayant échoué à l'aide de la console

1. Sur la page List Pipelines, si la colonne Status de l'une de vos instances de pipeline affiche un autre état que FINISHED, cela signifie soit que votre pipeline attend qu'une condition préalable soit remplie, soit qu'il a échoué et que vous devez le dépanner.
2. Sur la page List Pipelines (Lister les pipelines), localisez le pipeline d'instance et sélectionnez le triangle à gauche de celui-ci pour développer les détails.
3. Au bas de cet écran, choisissez View execution details (Afficher les détails d'exécution). L'écran Instance summary (Récapitulatif de l'instance) affiche alors les détails de l'instance sélectionnée.
4. Dans le volet Instance summary, cliquez sur le triangle en regard d'une instance pour afficher les détails de l'instance sélectionnée, puis choisissez Détails, Plus... Si l'état de l'instance sélectionnée est FAILED, la zone des détails contient des entrées pour le message d'erreur, `errorStackTrace` et d'autres informations. Vous pouvez enregistrer ces informations dans un fichier. Sélectionnez OK.
5. Dans le volet Instance summary (Récapitulatif de l'instance), choisissez Attempts (Tentatives) pour afficher les détails de chaque ligne de tentative.

6. Pour effectuer une action sur votre instance incomplète ou ayant échoué, cochez la case en regard de l'instance. Cela permet d'activer les actions. Puis, sélectionnez une action (`Rerun|Cancel|Mark Finished`).

Identification du cluster Amazon EMR qui dessert votre pipeline

Si un `EMRCluster` ou `EMRActivity` échoue et que les informations d'erreur fournies par la AWS Data Pipeline console ne sont pas claires, vous pouvez identifier le cluster Amazon EMR qui dessert votre pipeline à l'aide de la console Amazon EMR. Cela vous permet de localiser les journaux fournis par Amazon EMR afin d'obtenir plus de détails sur les erreurs qui se produisent.

Pour obtenir des informations plus détaillées sur les erreurs Amazon EMR

1. Dans la console AWS Data Pipeline, sélectionnez le triangle en regard de l'instance de pipeline pour développer les détails de l'instance.
2. Choisissez View execution details (Afficher les détails d'exécution), puis sélectionnez le triangle en regard du composant.
3. Dans la colonne Details (Détails), choisissez More... (Plus). L'écran d'information s'ouvre en répertoriant les détails du composant. Localisez et copiez la valeur instanceParent sur l'écran, telle que : `@EmrActivityId_xiFDD_2017-09-30T21:40:13`
4. Accédez à la console Amazon EMR, recherchez un cluster dont le nom contient la valeur InstanceParent correspondante, puis choisissez Debug.

   > **Note**
   >
   > Pour que le bouton Debug fonctionne, la définition de votre pipeline doit avoir défini l'EmrActivity`enableDebugging`option `true` et l'`EmrLogUri`option sur un chemin valide.

5. Maintenant que vous savez quel cluster Amazon EMR contient l'erreur à l'origine de la défaillance de votre pipeline, suivez les conseils de résolution des problèmes du Guide du développeur Amazon EMR.

Interprétation des détails sur l'état du pipeline

Les divers niveaux d'état affichés dans la console AWS Data Pipeline et l'interface de ligne de commande indiquent la condition d'un pipeline et de ses composants. L'état du pipeline n'est qu'une

présentation d'un pipeline ; pour afficher plus d'informations, consultez l'état de chaque composant du pipeline. Pour ce faire, cliquez sur un pipeline dans la console ou récupérez les détails des composants du pipeline à l'aide de l'interface de ligne de commande.

Codes de statut

`ACTIVATING`

Le composant ou la ressource est en cours de démarrage, par exemple une instance EC2.

`CANCELED`

Le composant a été annulé par un utilisateur ou AWS Data Pipeline avant son exécution. Cela peut se produire automatiquement lorsqu'une défaillance survient dans un composant ou une ressource différente dont dépend ce composant.

`CASCADE_FAILED`

Le composant ou la ressource a été annulé à la suite d'une défaillance en cascade liée à l'une de ses dépendances, mais le composant n'était probablement pas à l'origine de la panne.

`DEACTIVATING`

Le pipeline est en cours de désactivation.

`FAILED`

Le composant ou la ressource a rencontré une erreur et a cessé de fonctionner. Lorsqu'un composant ou une ressource tombe en panne, cela peut entraîner des annulations et des défaillances se répercuter sur d'autres composants qui en dépendent.

`FINISHED`

La composante a terminé le travail qui lui avait été assigné.

`INACTIVE`

Le gazoduc a été désactivé.

`PAUSED`

Le composant a été suspendu et ne fonctionne pas actuellement.

`PENDING`

Le pipeline est prêt à être activé pour la première fois.

`RUNNING`

La ressource est en cours d'exécution et prête à recevoir du travail.

`SCHEDULED`

L'exécution de la ressource est planifiée.

`SHUTTING_DOWN`

La ressource s'arrête après avoir terminé son travail avec succès.

`SKIPPED`

Le composant a ignoré des intervalles d'exécution après l'activation du pipeline en utilisant un horodatage postérieur au calendrier actuel.

`TIMEDOUT`

La ressource a dépassé le `terminateAfter` seuil et a été bloquéeAWS Data Pipeline. Une fois que la ressource a atteint ce statut, AWS Data Pipeline ignore les `retryTimeout` valeurs `actionOnResourceFailureretryDelay`, et de cette ressource. Ce statut s'applique uniquement aux ressources.

`VALIDATING`

La définition du pipeline est en cours de validation parAWS Data Pipeline.

`WAITING_FOR_RUNNER`

Le composant attend que son client de travail récupère un élément de travail. La relation entre le composant et le travailleur-client est contrôlée par les `workerGroup` champs `runsOn` ou définis par ce composant.

`WAITING_ON_DEPENDENCIES`

Le composant vérifie que ses conditions préalables par défaut et configurées par l'utilisateur sont remplies avant d'effectuer son travail.

Localisation des journaux des erreurs

Cette section explique comment trouver les divers journaux écrits par AWS Data Pipeline, que vous pouvez utiliser pour déterminer la source de certains échecs et erreurs.

Journaux de pipeline

Nous vous recommandons de configurer les pipelines pour créer des fichiers journaux dans un emplacement persistant, comme dans l'exemple suivant où vous utilisez le `pipelineLogUri` champ de l'`Default`objet d'un pipeline pour que tous les composants du pipeline utilisent un emplacement de journal Amazon S3 par défaut (vous pouvez remplacer cela en configurant un emplacement de journal dans un composant de pipeline spécifique).

> **Note**
>
> Task Runner stocke ses journaux dans un emplacement différent par défaut, qui peut ne pas être disponible lorsque le pipeline se termine et que l'instance qui exécute Task Runner se termine. Pour plus d'informations, veuillez consulter Vérification de la journalisation de Task Runner.

Pour configurer l'emplacement de journal à l'aide de l'interface de ligne de commande AWS Data Pipeline dans un fichier JSON de pipeline, commencez votre fichier de pipeline par le texte suivant :

```
{ "objects": [
{
  "id":"Default",
  "pipelineLogUri":"s3://mys3bucket/error_logs"
},
...
```

Après avoir configuré un répertoire de journaux de pipeline, Task Runner crée une copie des journaux de votre répertoire, avec le même format et les mêmes noms de fichier que ceux décrits dans la section précédente sur les journaux de Task Runner.

Journaux d'étapes Hadoop Job et Amazon EMR

Pour toute activité basée sur Hadoop telle que HadoopActivityHiveActivity, ou PigActivity vous pouvez consulter les journaux des tâches Hadoop à l'emplacement renvoyé dans le slot d'exécution,. hadoopJobLog EmrActivitypossède ses propres fonctionnalités de journalisation et ces journaux sont stockés à l'emplacement choisi par Amazon EMR et renvoyés par le slot d'exécution. emrStepLog Pour plus d'informations, consultez Afficher les fichiers journaux dans le manuel Amazon EMR Developer Guide.

Résolution des problèmes courants

Cette rubrique présente divers symptômes de problèmes AWS Data Pipeline et les étapes recommandées pour les résoudre.

Table des matières

Pipeline bloqué à l'état Pending (en suspens)

Un pipeline qui semble bloqué à l'état PENDING (en suspens) indique qu'un pipeline n'a pas encore été activé ou que l'activation a échoué en raison d'une erreur dans la définition du pipeline. Vérifiez que vous n'avez reçu aucune erreur lorsque vous avez envoyé votre pipeline à l'aide de l'interface de ligne de commande AWS Data Pipeline ou lorsque vous avez tenté d'enregistrer ou d'activer votre pipeline à l'aide de la console AWS Data Pipeline. Vérifiez également que la définition de votre pipeline est valide.

Pour afficher votre définition de pipeline à l'écran à l'aide de l'interface de ligne de commande :

```
aws datapipeline --get-pipeline-definition --pipeline-id df-EXAMPLE_PIPELINE_ID
```

Assurez-vous que la définition de pipeline est complète, vérifiez les accolades fermantes, les virgules requises, et assurez-vous qu'il ne manque aucune référence et qu'il n'y a pas d'autres erreurs de syntaxe. Il est préférable d'utiliser un éditeur de texte qui permet de valider visuellement la syntaxe des fichiers JSON.

Composant de pipeline bloqué à l'état Waiting for Runner

Si votre pipeline a l'état SCHEDULED et qu'une ou plusieurs tâches semblent bloquées à l'état WAITING_FOR_RUNNER, vérifiez que vous avez affecté une valeur valide aux champs runsOn et workerGroup de ces tâches. Si ces deux valeurs sont vides ou manquantes, la tâche ne peut pas commencer car il n'y a aucune association entre la tâche et un programme d'exécution devant effectuer les tâches. Dans ce cas, vous avez défini le travail mais vous n'avez pas défini l'ordinateur qui doit effectuer ce travail. Le cas échéant, vérifiez que la valeur WorkerGroup attribuée au composant du pipeline correspond exactement au même nom et à la même majuscule que la valeur WorkerGroup que vous avez configurée pour Task Runner.

> **Note**
>
> Si vous fournissez une valeur runsOn et que workerGroup existe, workerGroup est ignoré.

Une autre cause potentielle de ce problème est que le point de terminaison et la clé d'accès fournis à Task Runner ne sont pas les mêmes que ceux de la AWS Data Pipeline console ou de l'ordinateur sur lequel les outils AWS Data Pipeline CLI sont installés. Vous avez peut-être créé de nouveaux pipelines sans erreur visible, mais Task Runner interroge le mauvais emplacement en raison de la différence entre les informations d'identification, ou interroge le bon emplacement avec des autorisations insuffisantes pour identifier et exécuter le travail spécifié par la définition du pipeline.

Composant de pipeline bloqué à l'état WAITING_ON_DEPENDENCIES

Si votre pipeline est à l'état `SCHEDULED` et qu'une ou plusieurs tâches semblent bloquées à l'état `WAITING_ON_DEPENDENCIES`, vérifiez que les conditions préalables initiales de votre pipeline ont été remplies. Si les conditions préalables du premier objet de la chaîne logique ne sont pas remplies, aucun des objets qui dépendent de ce premier objet ne peut quitter l'état `WAITING_ON_DEPENDENCIES`.

Par exemple, étudiez l'extrait suivant d'une définition de pipeline. Dans ce cas, l'`InputData`objet possède une précondition « Prêt » qui indique que les données doivent exister avant que l'InputDataobjet ne soit terminé. Si les données n'existent pas, l'InputDataobjet reste dans l'`WAITING_ON_DEPENDENCIES`état en attendant que les données spécifiées par le champ de chemin soient disponibles. Tous les objets qui en InputData dépendent restent dans un `WAITING_ON_DEPENDENCIES` état en attendant InputData qu'ils atteignent `FINISHED` cet état.

```
{
    "id": "InputData",
    "type": "S3DataNode",
    "filePath": "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
    "schedule":{"ref":"MySchedule"},
    "precondition": "Ready"
},
{
    "id": "Ready",
    "type": "Exists"
...
```

Par ailleurs, vérifiez que vos objets disposent des autorisations nécessaires pour accéder aux données. Dans l'exemple précédent, si les informations du champ d'identification n'étaient pas autorisées à accéder aux données spécifiées dans le champ de chemin, l'InputDataobjet resterait bloqué dans un `WAITING_ON_DEPENDENCIES` état car il ne pouvait pas accéder aux données spécifiées par le champ de chemin, même si ces données existent.

Il est également possible qu'aucune adresse IP publique ne soit associée à une ressource communiquant avec Amazon S3. Par exemple, une ressource `Ec2Resource` d'un sous-réseau public doit avoir une adresse IP publique associée.

Enfin, dans certaines conditions, les instances de ressource peuvent atteindre l'état `WAITING_ON_DEPENDENCIES` beaucoup plus tôt que le début planifié de leurs activités associées, ce qui peut donner l'impression que la ressource ou l'activité échoue.

L'exécution ne démarre pas au moment planifié

Vérifiez que vous avez sélectionné le type de planification correct qui détermine si votre tâche démarre au début de l'intervalle de planification (planification de type cron) ou à la fin de l'intervalle de planification (planification de type séries chronologiques).

Vérifiez également que vous avez correctement spécifié les dates dans vos objets de calendrier et que les endDateTime valeurs startDateTime et sont au format UTC, comme dans l'exemple suivant :

```
{
    "id": "MySchedule",
    "startDateTime": "2012-11-12T19:30:00",
    "endDateTime":"2012-11-12T20:30:00",
    "period": "1 Hour",
    "type": "Schedule"
},
```

Les composants du pipeline s'exécutent dans le mauvais ordre

Vous pouvez remarquer que les heures de début et de fin de vos composants de pipeline s'exécutent dans le mauvais ordre ou dans une séquence différente de celle que vous attendez. Il est important de comprendre que les composants de pipeline peuvent commencer à s'exécuter simultanément si leurs conditions préalables sont remplies au démarrage. En d'autres termes, les composants de pipeline ne s'exécutent pas de manière séquentielle par défaut ; si vous avez besoin d'un ordre d'exécution spécifique, vous devez contrôler l'ordre d'exécution à l'aide de conditions préalables et de champs `dependsOn`.

Vérifiez que vous utilisez le champ `dependsOn` renseigné avec une référence aux composants de pipeline avec conditions préalables adéquats et que tous les pointeurs nécessaires entre les composants sont présents pour atteindre l'ordre souhaité.

Le cluster EMR échoue en renvoyant l'erreur suivante : Le jeton de sécurité inclus dans la demande n'est pas valide

Vérifiez vos rôles IAM, vos politiques et vos relations de confiance comme décrit dansRôles IAM pour AWS Data Pipeline.

Autorisations insuffisantes pour accéder aux ressources

Les autorisations que vous définissez sur les rôles IAM déterminent si vous AWS Data Pipeline pouvez accéder à vos clusters EMR et à vos instances EC2 pour exécuter vos pipelines. De plus, IAM propose le concept de relations de confiance qui va plus loin pour permettre la création de ressources en votre nom. Par exemple, lorsque vous créez un pipeline qui utilise une instance EC2 pour exécuter une commande afin de déplacer des données, AWS Data Pipeline peut mettre automatiquement cette instance EC2 en service. Si vous rencontrez des problèmes, en particulier

ceux impliquant des ressources auxquelles vous pouvez accéder manuellement mais que vous AWS Data Pipeline ne pouvez pas, vérifiez vos rôles IAM, vos politiques et vos relations de confiance comme décrit dansRôles IAM pour AWS Data Pipeline.

Code d'état : 400 Code d'erreur : PipelineNotFoundException

Cette erreur signifie que vos rôles IAM par défaut ne disposent peut-être pas des autorisations requises AWS Data Pipeline pour fonctionner correctement. Pour plus d'informations, veuillez consulter Rôles IAM pour AWS Data Pipeline.

La création d'un pipeline provoque une erreur de jeton de sécurité

Vous recevez le message d'erreur suivant lorsque vous essayez de créer un pipeline :

Échec de la création de pipeline avec 'pipeline_name'. Erreur : UnrecognizedClientException - Le jeton de sécurité inclus dans la demande n'est pas valide.

Impossible de voir les détails du pipeline dans la console

Le filtre de pipeline de la console AWS Data Pipeline s'applique à la date de début planifiée d'un pipeline, sans tenir compte du moment où le pipeline a été envoyé. Il est possible d'envoyer un nouveau pipeline en utilisant une date de début planifiée qui appartient au passé, que le filtre de date par défaut ne peut pas afficher. Pour voir les détails du pipeline, modifiez votre filtre de date afin de garantir que la date de début planifiée du pipeline soit prise en compte dans le filtre de plage de dates.

Erreur du programme d'exécution à distance - Code d'état : 404, service AWS : Amazon S3

Cette erreur signifie que Task Runner n'a pas pu accéder à vos fichiers dans Amazon S3. Vérifiez que :

- Vous avez correctement défini les informations d'identification
- Le compartiment Amazon S3 auquel vous essayez d'accéder existe
- Vous êtes autorisé à accéder au compartiment Amazon S3

Accès refusé - Vous n'êtes pas autorisé à exécuter la fonction datapipeline :

Dans les journaux de Task Runner, vous pouvez voir une erreur similaire à la suivante :

- ERREUR - Code d'état : 403
- Service AWS : DataPipeline
- Code d'erreur AWS : AccessDenied
- Message d'erreur AWS : L'utilisateur : arn:aws:sts : :XXXXXXXXXX:Federated-User/I-XXXXXXXX n'est pas autorisé à exécuter : datapipeline :. PollForTask

> **Note**
>
> Dans ce message d'erreur, PollForTask peuvent être remplacés par les noms d'autres AWS Data Pipeline autorisations.

Ce message d'erreur indique que le rôle IAM que vous avez spécifié nécessite des autorisations supplémentaires pour interagir avec AWS Data Pipeline celui-ci. Assurez-vous que votre politique de rôle IAM contient les lignes suivantes, où elles PollForTask sont remplacées par le nom de l'autorisation que vous souhaitez ajouter (utilisez* pour accorder toutes les autorisations). Pour plus d'informations sur la façon de créer un nouveau rôle IAM et de lui appliquer une politique, consultez la section Gestion des politiques IAM dans le guide Using IAM.

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

Les anciennes AMI Amazon EMR peuvent créer de fausses données pour les fichiers CSV volumineux

Sur les AMI Amazon EMR antérieures à la version 3.9 (3.8 et antérieures), elle AWS Data Pipeline utilise une fonctionnalité personnalisée InputFormat pour lire et écrire des fichiers CSV à utiliser avec des MapReduce tâches. Ceci est utilisé lorsque le service transfère des tables vers et depuis Amazon S3. Ce problème a InputFormat été découvert : la lecture d'enregistrements à partir de fichiers CSV volumineux peut entraîner la production de tableaux qui ne sont pas correctement copiés. Ce problème a été résolu dans les versions ultérieures d'Amazon EMR. Veuillez utiliser Amazon EMR AMI 3.9 ou une version Amazon EMR 4.0.0 ou ultérieure.

Augmentation des limites pour AWS Data Pipeline

De temps en temps, il est possible que vous dépassiez des limites spécifiques du système AWS Data Pipeline. Par exemple, la limite de pipelines par défaut est 20 pipelines contenant chacun 50 objets. Si vous découvrez que vous avez besoin d'un nombre de pipelines supérieur à la limite, envisagez de fusionner plusieurs pipelines pour créer moins de pipelines contenant chacun davantage d'objets. Pour plus d'informations sur les limites d'AWS Data Pipeline, consultez Limites AWS Data Pipeline. Toutefois, si vous n'êtes pas en mesure de contourner les limites à l'aide de la technique de fusion de pipelines, demandez une augmentation de votre capacité à l'aide de ce formulaire : Augmentation de limite Data Pipeline.

Limites AWS Data Pipeline

Afin de s'assurer qu'il y ait de la capacité pour tous les utilisateurs, AWS Data Pipeline impose des limites sur les ressources que vous pouvez allouer et sur la vitesse à laquelle vous pouvez les allouer.

Table des matières

Limites de compte

Les limites suivantes s'appliquent à un seul compte AWS. Si vous avez besoin d'une capacité supplémentaire, vous pouvez utiliser le formulaire de demande du centre de support Amazon Web Services pour augmenter votre capacité.

Attribut	Limite	Ajustable
Nombre de pipelines	100	Oui
Nombre d'objets par pipeline	100	Oui
Nombre d'instances actives par objet	5	Oui
Nombre de champs par objet	50	Non
Nombre d'octets UTF8 par nom de champ ou identifiant	256	Non
Nombre d'octets UTF8 par champ	10 240	Non

Attribut	Limite	Ajustable
Nombre d'octets UTF8 par objet	15 360 (y compris les noms de champs)	Non
Taux de création d'une instance à partir d'un objet	1 toutes les 5 minutes	Non
Nouvelles tentatives d'une activité de pipeline	5 par tâche	Non
Délai minimal entre deux nouvelles tentatives	2 minutes	Non
Intervalle de planification minimal	15 minutes	Non
Nombre maximal de regroupements dans un seul objet	32	Non
Nombre maximal d'instances EC2 par objet Ec2Resource	1	Non

Limites de l'appel du service web

AWS Data Pipeline limite la vitesse à laquelle vous pouvez appeler l'API du service web. Ces limites s'appliquent également auxAWS Data Pipeline agents qui appellent l'API du service Web en votre nom, tels que la console, la CLI et Task Runner.

Les limites suivantes s'appliquent à un seul compte AWS. Cela signifie que l'utilisation totale sur le compte, y compris par les utilisateurs , ne peut pas dépasser ces limites.

Le débit en rafale vous permet d'économiser les appels de service web pendant les périodes d'inactivité et de tous les utiliser en un court laps de temps. Par exemple, CreatePipeline a un taux normal d'un appel toutes les cinq secondes. Si vous n'avez pas appelé le service pendant 30 secondes, vous économisez six appels. Vous pouvez alors appeler le service web six fois en une seconde. Comme ce chiffre est inférieur à la limite du débit en rafale et maintient la moyenne de vos appels à la limite de fréquence standard, vos appels ne sont pas limités.

Si vous dépassez la limite de fréquence et la limite de débit en rafale, votre service web risque d'échouer et de renvoyer une exception de limitation. L'implémentation par défaut d'un worker, Task Runner, relance automatiquement les appels d'API qui échouent avec une exception de limitation. Task Runner dispose d'une fonction de sauvegarde, de sorte que les tentatives ultérieures d'appel de l'API se produisent à des intervalles de plus en plus longs. Si vous écrivez un travail, nous vous recommandons d'implémenter une logique similaire de nouvelle tentative.

Ces limites sont appliquées sur un compte AWS individuel.

API	Limite de fréquence régulière	Limite de débit en rafale
ActivatePipeline	1 appel par seconde	100 appels
CreatePipeline	1 appel par seconde	100 appels
DeletePipeline	1 appel par seconde	100 appels
DescribeObjects	2 appels par seconde	100 appels
DescribePipelines	1 appel par seconde	100 appels
GetPipelineDefinition	1 appel par seconde	100 appels
PollForTask	2 appels par seconde	100 appels
ListPipelines	1 appel par seconde	100 appels
PutPipelineDefinition	1 appel par seconde	100 appels
QueryObjects	2 appels par seconde	100 appels
ReportTaskProgress	10 appels par seconde	100 appels
SetTaskStatus	10 appels par seconde	100 appels

API	Limite de fréquence régulière	Limite de débit en rafale
SetStatus	1 appel par seconde	100 appels
ReportTaskRunnerHeartbeat	1 appel par seconde	100 appels
ValidatePipelineDefinition	1 appel par seconde	100 appels

Considérations sur le dimensionnement

AWS Data Pipeline s'adapte pour prendre en charge un très grand nombre de tâches simultanées et vous pouvez le configurer de manière à créer automatiquement les ressources nécessaires pour gérer les charges de travail très importantes. Ces ressources créées automatiquement sont sous votre contrôle et prises en compte dans le calcul des limites des ressources de votre compte AWS. Par exemple, si vous configurezAWS Data Pipeline pour créer automatiquement un cluster Amazon EMR à 20 nœuds pour traiter les données et que la limite d'instances EC2 de votreAWS compte est fixée à 20, vous risquez d'épuiser par inadvertance les ressources de remplacement disponibles. Par conséquent, tenez compte de ces restrictions de ressources dans votre conception ou augmentez les limites de votre compte en conséquence.

Si vous avez besoin d'une capacité supplémentaire, vous pouvez utiliser le formulaire de demande du centre de support Amazon Web Services pour augmenter votre capacité.

Ressources AWS Data Pipeline

Les ressources suivantes peuvent vous aider à utiliser AWS Data Pipeline.

- AWS Data PipelineInformations sur le produit : page web principale pour des informations surAWS Data Pipeline.
- AWS Data PipelineFAQ technique : couvre les 20 principales questions que les développeurs se posent à propos de ce produit.
- Notes de mise à jour — Fournissez un aperçu de haut niveau de la version actuelle. Elles indiquent, en particulier, les nouvelles fonctions et dernières corrections apportées, ainsi que les problèmes connus.
- Forums de discussion AWS Data Pipeline — Un forum communautaire qui permet aux développeurs d'échanger à propos de questions techniques liées à Amazon Web Services.

- Formations et ateliers — Liens vers des formations spécialisées et basées sur les rôles, en plus des ateliers d'autoformation pour améliorer vosAWS compétences et acquérir une expérience pratique.
- AWSCentre pour développeurs : découvrez des tutoriels, téléchargez des outils et découvrez les événements pour lesAWS développeurs.
- AWSOutils de développement — Liens vers des outils de développement, kits SDK, boîtes à outils IDE et outils de ligne de commande pour développer et gérer desAWS applications.
- Centre de ressources de mise en route (français non garanti) : découvrez comment configurer votreCompte AWS, rejoindre laAWS communauté et lancer votre première application.
- Tutoriels pratiques : suivez des step-by-step tutoriels pour lancer votre première application surAWS.
- AWSLivres blancs — Liens vers une liste complète desAWS livres blancs techniques, couvrant des sujets tels que l'architecture, la sécurité et l'économie, créés parAWS des architectes de solutions ou d'autres experts techniques.
- AWS SupportCentre – Hub pour la création et la gestion de vos cas AWS Support. Inclut également des liens vers d'autres ressources utiles, telles que des forums, des FAQ techniques, l'état de santé d'un service et AWS Trusted Advisor.
- AWS Support— Principale page web d'informations à propos d'AWS Support one-on-one, un canal d'assistance technique rapide pour vous aider à développer et à exécuter des applications dans le cloud.

- Contactez-nous : point de contact central pour toute question relative à la facturation AWS, à votre compte, aux événements, à des abus ou à d'autres problèmes.
- AWSConditions d'utilisation du site : informations détaillées sur nos droits d'auteur et notre marque, sur votre compte, votre licence et votre accès au site, et sur d'autres sujets.

Historique du document

Cette documentation est associée à la version 2012-10-29 de. AWS Data Pipeline

Modification	Description	Date de parution
Documentation ajoutée pour exécuter certaines procédures à l'aide de l'AWSinterface de ligne de commande. Procédures liées à AWS Data Pipeline la console supprimées.	Pour plus d'informations, consultez Clonage de votre pipeline, Affichage des journaux de pipelines et Création d'un pipeline à partir de modèles de pipeline de données à l'aide de l'interface de ligne de commande.	26 mai 2023
Ajout de contenu et d'exemples supplémentaires pour la migration AWS Data Pipeline vers d'autres services alternatifs.	Mise à jour de la rubrique relative à la migration AWS Data Pipeline vers AWS Step Functions ou Amazon MWAA avec plus d'informations sur chaque alternati ve, les mappages conceptuels entre les services et des exemples. AWS Glue Pour plus d'informations, veuillez consulter Migration des charges de travail depuis AWS Data Pipeline.	31 mars 2023
Ajout d'informations sur le AWS Data Pipeline support d'IMDSv2.	AWS Data Pipelineprend en charge les ressources IMDSv2 pour Amazon EMR et Amazon EC2. Pour plus d'informations, consultez Protection des données dans AWS Data Pipeline, EmrCluster et Ec2Resource.	16 décembre 2022
Ajout d'une rubrique concernant la migration AWS Data Pipeline vers d'autres services alternatifs.	Il existe désormais d'autres AWS services qui offrent aux clients une meilleure expérience d'intégration des données. Vous pouvez migrer des cas d'AWS Data Pipelineutilisation classiques vers AWS Step Functions ou Amazon MWAA. AWS Glue Pour plus d'informa	16 décembre 2022

Modification	Description	Date de parution
	tions, veuillez consulter Migration des charges de travail depuis AWS Data Pipeline.	
Mise à jour des listes des instances Amazon EC2 et Amazon EMR prises en charge. Mise à jour de la liste des ID des AMI HVM (Hardware Virtual Machine) utilisées pour les instances.	Mise à jour des listes des instances Amazon EC2 et Amazon EMR prises en charge. Pour plus d'informations, veuillez consulter Types d'instances prises en charge pour les activités de pipeline. Mise à jour de la liste des ID des AMI HVM (Hardware Virtual Machine) utilisées pour les instances. Pour plus d'informations, consultez Syntaxe et recherchez `imageId`.	9 novembre 2018

Modification	Description	Date de parution
Ajout d'une configuration pour attacher des volumes Amazon EBS à des nœuds de cluster et pour lancer un cluster Amazon EMR dans un sous-réseau privé.	Ajout d'options de configuration à un objet `EMRcluster` . Vous pouvez utiliser ces options dans les pipelines qui utilisent des clusters Amazon EMR. Utilisez les `TaskEbsConfiguration` champs `coreEbsConfiguration` `masterEbsConfiguration` , et pour configurer l'attachement des volumes Amazon EBS aux nœuds principaux, principaux et de tâches du cluster Amazon EMR. Pour plus d'informations, veuillez consulter Attachement des volumes EBS aux nœuds de cluster. Utilisez les `ServiceAccessSecurityGroupId` champs `emrManagedMasterSecurityGroupId` `emrManagedSlaveSecurityGroupId` , et pour configurer un cluster Amazon EMR dans un sous-réseau privé. Pour plus d'informations, veuillez consulter Configuration d'un cluster Amazon EMR dans un sous-réseau privé. Pour plus d'informations sur la syntaxe `EMRcluster` , consultez EmrCluster.	19 avril 2018
Ajout de la liste des instances Amazon EC2 et Amazon EMR prises en charge.	Ajout de la liste des instances qui sont créées par défaut par AWS Data Pipeline, si vous ne spécifiez pas de type d'instance dans la définition de pipeline. Ajout d'une liste des instances Amazon EC2 et Amazon EMR prises en charge. Pour plus d'informations, veuillez consulter Types d'instances prises en charge pour les activités de pipeline.	22 mars 2018
Ajout de la prise en charge des pipelines à la demande.	• Ajout de la prise en charge des pipelines à la demande, ce qui permet d'exécuter à nouveau un pipeline en le réactivant.	22 février 2016

Modification	Description	Date de parution
Prise en charge additionnelle des bases de données RDS	• Ajout de `rdsInstanceId` , `region` et `jdbcDriverJarUri` à RdsDatabase. • `database` mis à jour dans SqlActivity pour également prendre en charge `RdsDatabase` .	17 août 2015
Prise en charge de JDBC supplémentaire	• `database` mis à jour dans SqlActivity pour également prendre en charge `JdbcDatabase` . • Ajout de `jdbcDriverJarUri` à JdbcDatabase. • Ajout de `initTimeout` à Ec2Resource et EmrCluster. • Ajout de `runAsUser` à Ec2Resource	7 juillet 2015
HadoopActivity, zone de disponibilité et support ponctuel	• Ajout de la prise en charge de l'envoi de travaux parallèles aux clusters Hadoop. Pour plus d'informations, veuillez consulter HadoopActivity. • Ajout de la possibilité de demander des instances Spot avec Ec2Resource et EmrCluster. • Ajout de la possibilité de lancer des ressources `EmrCluster` dans une zone de disponibilité spécifiée.	1 juin 2015
Désactivation des pipelines	Ajout de la prise en charge de la désactivation des pipelines actifs. Pour plus d'informations, veuillez consulter Désactivation de votre pipeline.	7 avril 2015

Modification	Description	Date de parution
Mise à jour des modèles et de la console	Ajout de nouveaux modèles. Le chapitre Getting Started a été mis à jour pour utiliser le ShellComm andActivity modèle Getting Started with. Pour plus d'informations, veuillez consulter Création d'un pipeline à partir de modèles de pipeline de données à l'aide de l'interface de ligne de commande.	25 novembre 2014
Prise en charge de VPC	Ajout de la prise en charge du lancement des ressources dans un cloud privé virtuel (VPC).	12 mars 2014
Prise en charge de la région	Ajout de la prise en charge de plusieurs régions de service. Outre `us-east-1` , AWS Data Pipeline est pris en charge dans `eu-west-1` , `ap-northeast-1` , `ap-southeast-2` et `us-west-2` .	20 février 2014
Prise en charge d'Amazon Redshift	La prise en charge d'Amazon Redshift a été ajoutée dansAWS Data Pipeline, notamment un nouveau modèle de console (Copier vers Redshift) et un didacticiel pour présenter le modèle. Pour plus d'informations, consultez Copier des données vers Amazon Redshift à l'aide de AWS Data Pipeline, RedshiftDataNode, RedshiftDatabase et RedshiftCopyActivity.	6 novembre 2013
PigActivity	AjoutéPigActivity, qui fournit un support natif pour Pig. Pour plus d'informations, veuillez consulter PigActivity.	15 octobre 2013
Nouveaux modèle de console, activité et format de données	Ajout du nouveau modèle de console CrossRegion DynamoDB Copy, y compris le nouveau HiveCopyActivity et DynamoDB. ExportDataFormat	21 août 2013

Modification	Description	Date de parution
Mise en cascade des échecs et des réexécutions	Ajout d'informations sur le comportement AWS Data Pipeline de mise en cascade des échecs et des réexécutions. Pour plus d'informations, veuillez consulter Mise en cascade des échecs et des réexécutions.	8 août 2013
Vidéo de résolution des problèmes	Ajout de la vidéo sur le dépannage de base des problèmes AWS Data Pipeline. Pour plus d'informations, veuillez consulter Résolution des problèmes.	17 juillet 2013
Modification des pipelines actifs	Ajout d'informations supplémentaires sur la modification des pipelines actifs et la réexécution de composants de pipeline. Pour plus d'informations, veuillez consulter Modification de votre pipeline.	17 juillet 2013
Utilisation des ressources dans différentes régions	Ajout d'informations sur l'utilisation des ressources dans différentes régions. Pour plus d'informations, veuillez consulter Utilisation d'un pipeline avec des ressources dans plusieurs régions.	17 juin 2013
État WAITING_ON_DEPENDENCIES	État CHECKING_PRECONDITIONS modifié en WAITING_ON_DEPENDENCIES et ajout du champ lié à l'exécution @waitingOn pour les objets de pipeline.	20 mai 2013
DynamoDB DataFormat	Modèle DynamoDB DataFormat ajouté.	23 avril 2013
Vidéo sur le traitement des journaux web et prise en charge des instances Spot	Présentation de la vidéo « Traitez les journaux Web avec AWS Data Pipeline, Amazon EMR et Hive » et de la prise en charge des instances Spot Amazon EC2.	21 février 2013
	Version initiale du Guide du développeur AWS Data Pipeline.	20 décembre 2012

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.